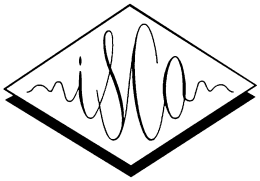
SIGDIAL 2022

23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue



Proceedings of the Conference

07-09 September 2022
Heriot-Watt University, Edinburgh, UK

In cooperation with:
Association for Computational Linguistics (ACL)
International Speech Communication Association (ISCA)
Association for the Advancement of Artificial Intelligence (AAAI)

We thank our sponsors:

- LivePerson
- Apple
- Alana
- Toshiba
- Furhat Robotics

In cooperation with

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-955917-66-7

Preface

We are glad to pen the first few words for the proceedings of SIGDIAL 2022, the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue. The SIGDIAL conference is a premier publication venue for research in discourse and dialogue. This year the conference is organized as a hybrid event with both in-person and remote participation on September 7-9, 2022, at Heriot-Watt University, Edinburgh, Scotland, and is hosted by the Interaction Lab and the National Robotarium.

The SIGDIAL 2022 program features 3 keynote talks, 6 sessions of in-person paper presentations, including the special session on Natural Language in Human-Robot Interaction (NLiHRI), 2 in-person mixed demo and poster sessions, and 5 remote presentation sessions. The 2022 Young Researchers' Roundtable on Spoken Dialog Systems (YRRSDS 2022) is also being held as a satellite event, just before SIGDIAL, on September 5-6.

SIGDIAL received 140 submissions this year, comprising 79 long papers, 49 short papers, and 12 demo descriptions. We had 14 Senior Program Committee (SPC) members who were each responsible for 9-11 papers, leading the discussion process and also contributing with meta-reviews. Each submission was assigned to an SPC member and received at least three reviews. Decisions carefully considered the original reviews, meta-reviews, and discussions among reviewers facilitated by the SPCs. We are immensely grateful to the members of the Program Committee and Senior Program Committee for their efforts in providing excellent, thoughtful reviews of the large number of submissions. Their contributions have been essential to selecting the accepted papers and providing a high-quality technical program for the conference. We have aimed to develop a broad, varied program spanning the many positively-rated papers identified by the review process. We therefore accepted 64 papers in total: 37 long papers (47%), 19 short papers (39%), and 8 demo descriptions, for an overall acceptance rate of 45.7%. The topics to be presented demonstrate the current breadth of research in discourse and dialogue.

In organizing this hybrid in-person/ remote conference, we have tried to maintain as much of the spirit of a fully in-person conference as possible, allowing opportunities for questions and discussion. Recordings for all remote papers and demos will be made available, and will be played to the audience in the conference auditorium, with an opportunity for authors to answer questions live online. We have also set up slack channels for online discussions. Long remote papers will each be presented as a seven-minute pre-recorded talk followed by three minutes of live Q&A, and short/demo remote papers will be presented as a four-minute pre-recorded talk followed by three minutes of live Q&A. A conference of this scale requires the energy, guidance, and contributions of many parties, and we would like to take this opportunity to thank and acknowledge them all.

We thank our three keynote speakers, Yun-Nung (Vivian) Chen (National Taiwan University), Angeliki Lazaridou (DeepMind), and Giuseppe Carenini (University of British Columbia), for their inspiring talks on "Robustness, Scalability, and Practicality of Conversational AI", "On opportunities and challenges on communicating using Large Language Models", and "Unlimited discourse structures in the era of distant supervision, pre-trained language models and autoencoders". We also thank the organizers of the special session: "Natural Language in Human-Robot Interaction (NLiHRI)". We are grateful for their coordination with the main conference.

SIGDIAL 2022 is made possible by the dedication and hard work of our community, and we are indebted to many. The conference would not have been possible without the advice and support of the SIGDIAL board, particularly Gabriel Skantze and Milica Gasic. The hybrid nature of the conference inevitably increases the workload for the organizers, and so special thanks go to Daniel Hernández Garcia for his tireless effort in managing the website with timely updates, and to the team handling various online aspects of participation: Angus Addlesee, Arash Ashrafzadeh, Bhathiya Hemanthage, Selina Meyer, and Nikolas Vitsakis. Many thanks also go to Tanvi Dinkar, Amit Parekh, and Weronika Sieinska for their

support with local arrangements.

We would also like to thank the sponsorship chair David Vandyke, who has been our SIGDIAL ambassador to industry year after year. He continues to bring to the conference an impressive panel of conference sponsors. We thank David for his dedicated effort. We gratefully acknowledge the support of our sponsors: LivePerson (Platinum), Apple (Gold), Alana (Gold), Toshiba Research Europe (Silver), and Furhat Robotics (Bronze). In addition, we thank Malihe Alikhani, the publication chair, and Ondřej Dušek, the mentoring chair for their dedicated service.

Finally, it is our great pleasure to welcome you physically and remotely to the conference. We hope that you will have an enjoyable and productive experience, and leave with fond memories of SIGDIAL 2022. With our best wishes for a successful conference.

Fàilte gu Alba !

Oliver Lemon, General Chair

Junyi Jessy Li, Dilek Hakkani-Tur, Program Co-Chairs

General Chair:

Oliver Lemon, Heriot-Watt University, Edinburgh, UK

Local Chairs:

Arash Ashrafzadeh, Heriot-Watt University, Edinburgh, UK
Daniel Hernández Garcia, Heriot-Watt University, Edinburgh, UK

Program Chairs:

Dilek Hakkani-Tur, Amazon, USA
Junyi Jessy Li, University of Texas at Austin, USA

Publication Chair:

Malihe Alikhani, University of Pittsburgh, USA

Sponsorship Chair:

David Vandyke, Apple, UK

Mentoring Chair:

Ondřej Dušek, Charles University, Czech Republic

SIGdial Officers:

President: Gabriel Skantze, KTH Royal Institute of Technology, Sweden
Vice President: Milica Gasic, University of Düsseldorf, Germany
Secretary: Alexandros Papangelis, Amazon, USA
Treasurer: Ethan Selfridge, LivePerson, USA
President Emeritus: Jason Williams, Apple, USA

Senior Program Committee:

Nancy Chen, Institute for Infocomm Research, A*STAR, Singapore
Ryuichiro Higashinaka, Nagoya University/NTT, Japan
Yufang Hou, IBM Research, Ireland
Yang Liu, Amazon, United States
Vincent Ng, University of Texas at Dallas, United States
Massimo Poesio, Queen Mary University of London, United Kingdom
David Schlangen, University of Potsdam, Germany
Matthew Stone, Rutgers University, United States
Deyi Xiong, Tianjin University, China
Ingrid Zukerman, Monash University, Australia
Chloé Braud, IRIT - CNRS - ANITI, France
Staffan Larsson, University of Gothenburg, Sweden
Verena Rieser, Heriot-Watt University, United Kingdom
Svetlana Stoyanchev, Toshiba Europe, United Kingdom

Program Committee:

Gavin Abercrombie, Heriot Watt University, United Kingdom
Tazin Afrin, University of Pittsburgh, United States
Sanchit Agarwal, Amazon, United States
Sean Andrist, Microsoft Research, United States
Masahiro Araki, Kyoto Institute of Technology, Japan
Ron Artstein, USC Institute for Creative Technologies, United States
Katherine Atwell, University of Pittsburgh, United States
Timo Baumann, Ostbayerische Technische Hochschule Regensburg, Germany
Jose Miguel Benedi, Universitàt Politècnica de València, Spain
Luciana Benotti, Universidad Nacional de Cordoba, Argentina
Parminder Bhatia, Amazon, United States
Nate Blaylock, Canary Speech, United States
Johan Boye, KTH, Sweden
Kristy Boyer, University of Florida, United States
Senthil Chandramohan, Microsoft, United States
Lin Chen, Engineering Manager, Meta Platform Inc., United States
Derek Chen, ASAPP, United States
Jinho D. Choi, Emory University, United States
Paul Crook, Facebook, United States
Heriberto Cuayahuitl, University of Lincoln, United Kingdom
Nina Dethlefs, University of Hull, United Kingdom
David DeVault, Anticipant Speech, Inc., United States
Arash Eshghi, Heriot-Watt University, United Kingdom
Maxine Eskenazi, Carnegie Mellon University, United States
Mauro Falcone, Fondazione Ugo Bordoni, Italy
Manaal Faruqui, Google, United States
Elisa Ferracane, Abridge AI, Inc., United States
Milica Gasic, Heinrich Heine University Duesseldorf, Germany
Kallirroi Georgila, University of Southern California Institute for Creative Technologies, United States
Alborz Geramifard, Facebook AI, United States
Felix Gervits, US Army Research Laboratory, United States
Tirthankar Ghosal, Institute of Formal and Applied Linguistics, Charles University, Czech Republic
Venkata Subrahmanyan, Govindarajan University of Texas at Austin, United States
David Gros, University of California - Davis, United States
Prakhar Gupta, Carnegie Mellon University, United States
joakim gustafson, KTH, Sweden
Ivan Habernal, Technische Universität Darmstadt, Germany
Helen Hastie, Heriot-Watt University, United Kingdom
Devamanyu Hazarika, Amazon, United States
Larry Heck, Georgia Institute of Technology, United States
Behnam Hedayatnia, Amazon, United States
Julian Hough, Queen Mary University of London, United Kingdom
David M. Howcroft, Edinburgh Napier University, United Kingdom
Ruihong Huang, Texas A&M University, United States
Koji Inoue, Kyoto University, Japan
Di Jin, Amazon, United States
Chris Kedzie, Microsoft Semantic Machines, United States
Simon Keizer, Toshiba Europe Ltd, United Kingdom

Casey Kennington, Boise State University, United States
Chandra Khatri, Chief Scientist Officer and Head of AI, Got It AI, United States
Seokhwan Kim, Amazon Alexa AI, United States
Wei-Jen Ko, University of Texas at Austin, United States
Kazunori Komatani, Osaka University, Japan
Murathan Kurfalı, Stockholm University, Sweden
Kornel Laskowski, Carnegie Mellon University, United States
Pierre Lison, Norwegian Computing Centre, Norway
Bing Liu, Facebook, United States
Yang Janet Liu, Georgetown University, United States
Eduardo Lleida, Solano University of Zaragoza, Spain
Stephanie M. Lukin, U.S. Army Research Laboratory, United States
Andrea Madotto, Meta, United States
Ramesh Manuvinakurike, Intel labs, United States
Michael McTear, Ulster University, United Kingdom
Shikib Mehri, Carnegie Mellon University, United States
Mohsen Mesgar, UKP Lab, Technische Universität Darmstadt, Germany
Teruhisa Misu, Honda Research Institute USA, United States
Anhad Mohananey, Google, United States
Seungwhan Moon, Facebook Reality Labs, United States
Raymond Mooney, University of Texas at Austin, United States
Elena Musi, University of Liverpool, United Kingdom
Satoshi Nakamura, Nara Institute of Science and Technology, Japan
Mikio Nakano, C4A Research Institute, Inc., Japan
Mahdi Namazifar, Amazon Alexa AI, United States
Anna Nedoluzhko, Charles University in Prague, Czech Republic
Douglas O'Shaughnessy, INRS-EMT (Univ. of Quebec), Canada
Aishwarya Padmakumar, Amazon, United States
Alexis Palmer, University of Colorado Boulder, United States
Sheena Panthaplackel, The University of Texas at Austin, United States
Alexandros Papangelis, Amazon Alexa AI, United States
Rebecca Passonneau, The Pennsylvania State University, United States
Baolin Peng, Microsoft Research, United States
Nanyun Peng, University of California, Los Angeles, United States
Paul Piwek, The Open University, United Kingdom
Stephen Pulman, Apple Inc., United Kingdom
Matthew Purver, Queen Mary University of London, United Kingdom
Kun Qian, Columbia University, United States
Liang Qiu, Amazon Alexa AI, United States
Vikram Ramanarayanan, University of California, San Francisco, United States
Hannah Rashkin, Google Research, United States
Antoine Raux, Apple, United States
Ehud Reiter, University of Aberdeen, United Kingdom
Giuseppe Riccardi, University of Trento, Italy
Carolyn Rosé, Carnegie Mellon University, United States
Saurav Sahay, Intel Labs, United States
Sakriani Sakti, Japan Advanced Institute of Science and Technology, Japan
Chinnadhurai Sankar, Meta AI, United States
Ruhi Sarikaya, Amazon, United States
Abigail See, Stanford University, United States

Ethan Selfridge, LivePerson, United States
Weiyan Shi, Columbia University, United States
Gabriel Skantze, KTH Speech Music and Hearing, Sweden
Hiroaki Sugiyama, NTT Communication Science Labs., Japan
Alessandro Suglia, Heriot-Watt University, United Kingdom
António Teixeira, DETI/IEETA, University of Aveiro, Portugal
Takenobu Tokunaga, Tokyo Institute of Technology, Japan
Bo-Hsiang Tseng, Apple, United States
Gokhan Tur, Amazon Alexa AI, United States
Stefan Ultes, Mercedes-Benz AG, Germany
David Vandyke, Apple, United Kingdom
Ivan Vulić, University of Cambridge, United Kingdom
Hsin-Min Wang, Academia Sinica, Taiwan
Yi-Chia Wang, Facebook AI, United States
Nigel Ward, University of Texas at El Paso, United States
Jason D Williams, Apple, United States
Qingyang Wu, Columbia University, United States
Jiacheng Xu, Salesforce, United States
Koichiro Yoshino, RIKEN Robotics, Nara Institute of Science and Technology, Japan
Kai Yu, Shanghai Jiao Tong University, China
Maryam Zare, Pennsylvania State University, United States
Tiancheng Zhao, Binjiang Institute of Zhejiang University, China
Mingyang Zhou, Ph.D. Student at University of California, Davis, United States
Hendrik Buschmeier, Bielefeld University, Germany
Roger Moore, University of Sheffield, United Kingdom
Asad Sayeed, University of Gothenburg, Sweden
Kristina Striegnitz, Union College, United States
Jennifer Williams, University of Southampton, United Kingdom
Sina Zarrieß, University of Bielefeld, Germany

Invited Speakers:

Yun-Nung (Vivian) Chen, National Taiwan University, Taiwan
Angeliki Lazaridou, DeepMind, UK
Giuseppe Carenini, University of British Columbia, Canada

Table of Contents

Conference Program

Wednesday September 7, 2022

08:45–09:00 Opening Remarks

09:00–10:00 *Keynote 1: Robustness, Scalability, and Practicality of Conversational AI*
Yun-Nung (Vivian) Chen

10:00–10:30 Break

10:30–12:10 Oral Session 1: "E2E dialogue systems"

Post-processing Networks: Method for Optimizing Pipeline Task-oriented Dialogue Systems using Reinforcement Learning
Atsumoto Ohashi and Ryuichiro Higashinaka

Reducing Model Churn: Stable Re-training of Conversational Agents
Christopher Hidey, Fei Liu and Rahul Goel

Knowledge-Grounded Conversational Data Augmentation with Generative Conversational Networks
Yen Ting Lin, Alexandros Papangelis, Seokhwan Kim and Dilek Hakkani-Tur

Guiding the Release of Safer E2E Conversational AI through Value Sensitive Design
A. Stevie Bergman, Gavin Abercrombie, Shannon Spruit, Dirk Hovy, Emily Dinan, Y-Lan Boureau and Verena Rieser

Wednesday September 7, 2022 (continued)

12:10–13:00 Lunch

13:00–14:00 Poster + Demo Session 1

Controllable User Dialogue Act Augmentation for Dialogue State Tracking
Chun-Mao Lai, Ming-Hao Hsu, Chao-Wei Huang and Yun-Nung Chen

Developing an argument annotation scheme based on a semantic classification of arguments
Lea Kawaletz, Heidrun Dorgeloh, Stefan Conrad and Zeljko Bekcic

Multi-Task Learning for Depression Detection in Dialogs
Chuyuan Li, Chloé Braud and Maxime Amblard

To laugh or not to laugh? The use of laughter to mark discourse structure
Bogdan Ludusan and Barbara Schuppler

QualityAdapt: an Automatic Dialogue Quality Estimation Framework
John Mendonca, Alon Lavie and Isabel Trancoso

Graph Neural Network Policies and Imitation Learning for Multi-Domain Task-Oriented Dialogues
Thibault Cordier, Tanguy Urvoy, Fabrice Lefèvre and Lina M. Rojas Barahona

The DialPort tools
Jessica Huynh, Shikib Mehri, Cathy Jiao and Maxine Eskenazi

Simultaneous Job Interview System Using Multiple Semi-autonomous Agents
Haruki Kawai, Yusuke Muraki, Kenta Yamamoto, Divesh Lala, Koji Inoue and Tatsuya Kawahara

Wednesday September 7, 2022 (continued)

14:00–15:00 Oral Session 2: NLiHRI Special Session

Dialog Acts for Task Driven Embodied Agents
Spandana Gella, Aishwarya Padmakumar, Patrick Lange and Dilek Hakkani-Tur

Symbol and Communicative Grounding through Object Permanence with a Mobile Robot
Josue Torres-Foncesca, Catherine Henry and Casey Kennington

Towards Personality-Aware Chatbots
Daniel Fernau, Stefan Hillmann, Nils Feldhus, Tim Polzehl and Sebastian Möller

15:00–15:25 NLiHRI Special Session Panel

15:25–15:45 Break

15:45–16:00 Sponsor Session: LivePerson, Alana, Apple, Toshiba, Furhat Robotics

16:00–17:00 Remote Session 1

Towards Socially Intelligent Agents with Mental State Transition and Human Value
Liang Qiu, Yizhou Zhao, Yuan Liang, Pan Lu, Weiyan Shi, Zhou Yu and Song-Chun Zhu

Automatic Verbal Depiction of a Brick Assembly for a Robot Instructing Humans
rami younes, Gérard Bailly, Frederic Elisei and Damien Pellier

Are Interaction Patterns Helpful for Task-Agnostic Dementia Detection? An Empirical Exploration
Shahla Farzana and Natalie Parde

EDU-AP: Elementary Discourse Unit based Argument Parser
Sougata Saha, Souvik Das and Rohini Srihari

Using Transition Duration to Improve Turn-taking in Conversational Agents
Charles Threlkeld, Muhammad Umair and JP de Ruiter

Wednesday September 7, 2022 (continued)

DG2: Data Augmentation Through Document Grounded Dialogue Generation
Qingyang Wu, Song Feng, Derek Chen, Sachindra Joshi, Luis Lastras and Zhou Yu

17:00–18:00 Remote Session 2

When can I Speak? Predicting initiation points for spoken dialogue agents
Siyan Li, Ashwin Paranjape and Christopher Manning

Using Interaction Style Dimensions to Characterize Spoken Dialog Corpora
Nigel Ward

Multi-Domain Dialogue State Tracking with Top-K Slot Self Attention
Longfei Yang, Jiyi Li, Sheng Li and Takahiro Shinozaki

Building a Knowledge-Based Dialogue System with Text Infilling
Qiang Xue, Tetsuya Takiguchi and Yasuo Ariki

D&D: When to Say What and How: Adapting the Elaborateness and Indirectness of Spoken Dialogue Systems
Juliana Miehle, Wolfgang Minker, Stefan Ultes

D&D: Cognitive and social delays in the initiation of conversational repair
Julia Beret Mertens and J. P. de Ruiter

D&D: Referential Communication Between Friends and Strangers in the Wild
Kris Liu, Trevor D'Arcey, Marilyn Walker, Jean Fox Tree

Wednesday September 7, 2022 (continued)

18:00 **Drinks Reception**

Thursday September 8, 2022

09:00–10:00 *Keynote 2: On opportunities and challenges on communicating using Large Language Models*
Angeliki Lazaridou

10:00–10:30 **Break**

10:30–12:10 **Oral Session 3: "Generation"**

Generating Meaningful Topic Descriptions with Sentence Embeddings and LDA
Javier Miguel Sastre Martinez, Sean Gorman, Aisling Nugent and Anandita Pal

How Well Do You Know Your Audience? Toward Socially-aware Question Generation
Ian Stewart and Rada Mihalcea

GenTUS: Simulating User Behaviour and Language in Task-oriented Dialogues with Generative Transformers
Hsien-chin Lin, Christian Geishauser, Shutong Feng, Nurul Lubis, Carel van Niekerk, Michael Heck and Milica Gasic

AARGH! End-to-end Retrieval-Generation for Task-Oriented Dialog
Tomáš Nekvinda and Ondřej Dušek

Thursday September 8, 2022 (continued)

12:10–13:00 Lunch

13:00–14:40 Oral Session 4: "Deep dives into dialogue systems"

A Systematic Evaluation of Response Selection for Open Domain Dialogue
Behnam Hedayatnia, Di Jin, Yang Liu and Dilek Hakkani-Tur

Inferring Ranked Dialog Flows from Human-to-Human Conversations
Javier Miguel Sastre Martinez and Aisling Nugent

Structured Dialogue Discourse Parsing
Ta-Chung Chi and alexander rudnicky

"Do you follow me?": A Survey of Recent Approaches in Dialogue State Tracking
Léo Jacqmin, Lina M. Rojas Barahona and Benoit Favre

14:40–15:00 Break

15:00–16:00 Remote Session 3

MultiWOZ 2.4: A Multi-Domain Task-Oriented Dialogue Dataset with Essential Annotation Corrections to Improve State Tracking Evaluation
Fanghua Ye, Jarana Manotumruksa and Emine Yilmaz

The Duration of a Turn Cannot be Used to Predict When It Ends
Charles Threlkeld and JP de Ruiter

Getting Better Dialogue Context for Knowledge Identification by Leveraging Document-level Topic Shift
Nhat Tran and Diane Litman

Neural Generation Meets Real People: Building a Social, Informative Open-Domain Dialogue Agent
Ethan A. Chi, Ashwin Paranjape, Abigail See, Caleb Chiam, Trenton Chang, Kathleen Kenealy, Swee Kiat Lim, Amelia Hardy, Chetanya Rastogi, Haojun Li, Alexander Iyabor, Yutong He, Hari Sowrirajan, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, Jillian Tang, Avanika Narayan, Giovanni Campagna and Christopher Manning

DeepCon: An End-to-End Multilingual Toolkit for Automatic Minuting of Multi-Party Dialogues
Aakash Bhatnagar, Nidhir Bhavsar and Muskaan Singh

ICM : Intent and Conversational Mining from Conversation Logs
Sayantan Mitra, Roshni Ramnani, Sumit Ranjan and Shubhashis Sengupta

16:00–17:00 Remote Session 4

Entity-based De-noising Modeling for Controllable Dialogue Summarization
Zhengyuan Liu and Nancy Chen

iEval: Interactive Evaluation Framework for Open-Domain Empathetic Chatbots
Ekaterina Svikhnushina, Anastasiia Filippova and Pearl Pu

Unsupervised Domain Adaptation on Question-Answering System with Conversation Data
Amalia Adiba, Takeshi Homma and Yasuhiro Sogawa

UniDU: Towards A Unified Generative Dialogue Understanding Framework
Zhi Chen, Lu Chen, Bei Chen, Libo Qin, Yuncong Liu, Su Zhu, Jian-Guang LOU and Kai Yu

Advancing Semi-Supervised Task Oriented Dialog Systems by JSA Learning of Discrete Latent Variable Models
Yucheng Cai, Hong Liu, Zhijian Ou, Yi Huang and Junlan Feng

Thursday September 8, 2022 (continued)

17:00–18:00 Remote Session 5

Redwood: Using Collision Detection to Grow a Large-Scale Intent Classification Dataset
Stefan Larson and Kevin Leach

Dialogue Evaluation with Offline Reinforcement Learning
Nurul Lubis, Christian Geishauser, Hsien-chin Lin, Carel van Niekerk, Michael Heck, Shutong Feng and Milica Gasic

Disruptive Talk Detection in Multi-Party Dialogue within Collaborative Learning Environments with a Regularized User-Aware Network
Kyungjin Park, Hyunwoo Sohn, Wookhee Min, Bradford Mott, Krista Glazewski, Cindy E. Hmelo-Silver and James Lester

Generating Discourse Connectives with Pre-trained Language Models: Conditioning on Discourse Relations Helps Reconstruct the PDTB
Symon Stevens-Guille, Aleksandre Maskharashvili, Xintong Li and Michael White

Toward Self-Learning End-to-End Task-oriented Dialog Systems
Xiaoying ZHANG, Baolin Peng, Jianfeng Gao and Helen Meng

19:30 Banquet

Friday September 9, 2022

09:00–10:00 *Keynote 3: Unlimited discourse structures in the era of distant supervision, pre-trained language models and autoencoders*
Guiseppe Carenini

Friday September 9, 2022 (continued)

10:00–10:15 Break

10:15–11:30 Oral Session 5: "Dynamics and Methods I"

Combining Structured and Unstructured Knowledge in an Interactive Search Dialogue System
Svetlana Stoyanchev, Suraj Pandey, Simon Keizer, Norbert Braunschweiler and Rama Sanand Doddipatla

How Much Does Prosody Help Turn-taking? Investigations using Voice Activity Projection Models
Erik Ekstedt and Gabriel Skantze

What makes you change your mind? An empirical investigation in online group decision-making conversations
Georgi Karadzhov, Tom Stafford and Andreas Vlachos

11:30–11:40 Break

11:40–12:55 Oral Session 6: "Dynamics and Methods II"

Dialogue Term Extraction using Transfer Learning and Topological Data Analysis
Renato Vukovic, Michael Heck, Benjamin Ruppik, Carel van Niekerk, Marcus Zibrowius and Milica Gasic

Evaluating N-best Calibration of Natural Language Understanding for Dialogue Systems
Ranim Khojah, Alexander Berman and Staffan Larsson

LAD: Language Models as Data for Zero-Shot Dialog
Shikib Mehri, Yasemin Altun and Maxine Eskenazi

Friday September 9, 2022 (continued)

12:55–13:45 Lunch

13:45–14:45 Poster + Demo Session 2:

Improving Bot Response Contradiction Detection via Utterance Rewriting
Di Jin, Sijia Liu, Yang Liu and Dilek Hakkani-Tur

Comparison of Lexical Alignment with a Teachable Robot in Human-Robot and Human-Human-Robot Interactions
Yuya Asano, Diane Litman, Mingzhi Yu, Nikki Lobczowski, Timothy Nokes-Malach, Adriana Kovashka and Erin Walker

TREND: Trigger-Enhanced Relation-Extraction Network for Dialogues
Po-Wei Lin, Shang-Yu Su and Yun-Nung Chen

User Satisfaction Modeling with Domain Adaptation in Task-oriented Dialogue Systems
Yan Pan, Mingyang Ma, Bernhard Pflugfelder and Georg Groh

N-best Response-based Analysis of Contradiction-awareness in Neural Response Generation Models
Shiki Sato, Reina Akama, Hiroki Ouchi, Ryoko Tokuhisa, Jun Suzuki and Kentaro Inui

A Visually-Aware Conversational Robot Receptionist
Nancie Gunson, Daniel Hernandez Garcia, Weronika Sieińska, Angus Addlesee, Christian Dondrup, Oliver Lemon, Jose L. Part and Yanchao Yu

Demonstrating EMMA: Embodied MultiModal Agent for Language-guided Action Execution in 3D Simulated Environments
Alessandro Suglia, Bhathiya Hemanthage, Malvina Nikandrou, George Pantazopoulos, Amit Parekh, Arash Eshghi, Claudio Greco, Ioannis Konstas, Oliver Lemon and Verena Rieser

GRILLBot: A multi-modal conversational agent for complex real-world tasks
Carlos Gemmell, Federico Rossetto, Iain Mackie, Paul Owoicho, Sophie Fischer and Jeff Dalton

A System For Robot Concept Learning Through Situated Dialogue
Benjamin Kane, Felix Gervits, Matthias Scheutz and Matthew Marge

Friday September 9, 2022 (continued)

14:45 SIGDIAL BUSINESS MEETING

15:15 CLOSING and BEST PAPER AWARDS

Keynote Abstracts

Keynote 1 - Robustness, Scalability, and Practicality of Conversational AI

Yun-Nung (Vivian) Chen
National Taiwan University

Abstract

Even conversational systems have attracted a lot of attention recently, there are many remaining challenges to be resolved. This talk presents three different dimensions for improvement: 1) Robustness — how to deal with speech recognition errors for better language understanding performance, 2) Scalability — how to better utilize the limited data, and 3) Practicality — how to naturally perform recommendation in a conversational manner. All directions enhance the usefulness of conversational systems, showing the potential of guiding future research areas

Biography

Yun-Nung (Vivian) Chen is currently an associate professor in the Department of Computer Science Information Engineering at National Taiwan University. She earned her Ph.D. degree from Carnegie Mellon University, where her research interests focus on spoken dialogue systems and natural language processing. She was recognized as the Taiwan Outstanding Young Women in Science and received Google Faculty Research Awards, Amazon AWS Machine Learning Research Awards, MOST Young Scholar Fellowship, and FAOS Young Scholar Innovation Award. Her team was selected to participate in the first Alexa Prize TaskBot Challenge in 2021. Prior to joining National Taiwan University, she worked in the Deep Learning Technology Center at Microsoft Research Redmond.

Keynote 2 - On opportunities and challenges on communicating using Large Language Models

Angeliki Lazaridou
DeepMind

Abstract

From science fiction to Turing's seminal work on AI, language and communication have been among the central components of intelligent agents. Towards that dream, the new-generation of large language models (LLMs) have recently given rise to a new set of impressive capabilities, from generating human-like text to engaging in simple, few-turn conversations. So, how close do LLMs bring us to being able to interact with such intelligent agents during our lifetime? In this talk, I will review key recent developments on LLMs by the community and I will discuss these in the context of advancing communication research. At the same time, I will also highlight challenges of current models in producing goal-driven, safe and factual dialogues. Capitalizing on their strengths and addressing their weaknesses might allow us to unlock LLMs full potential in responsibly interacting with us, humans, about different aspects of our lives.

Biography

Angeliki Lazaridou is a Staff Research Scientist at DeepMind. She received a PhD in Brain and Cognitive Sciences from the University of Trento. Her PhD initially focused on developing neural network models and techniques for teaching agents language in grounded environments. However, one day in late 2015, while walking towards the lab she realized that interaction and communication should play a key role in this learning . This was the beginning of her work in deep learning and multi-agent communication. In the following years, she looked at this fascinating problem from many different angles: how to make this learning more realistic or how to extend findings from cooperative to self-agents and even how to make this communication resemble more natural language. Currently, she spends most of her time thinking and working on how to best make language models be in sync with the complex and ever-evolving world.

Keynote 3 - Unlimited discourse structures in the era of distant supervision, pre-trained language models and autoencoders

Giuseppe Carenini
University of British Columbia

Abstract

Historically, discourse processing relies on human annotated corpora that are very small and lack diversity, often leading to overfitting, poor performance in domain transfer, and minimal success of modern deep-learning solutions. So, wouldn't it be great if we could generate an unlimited amount of discourse structures for both monologues and dialogues, across genres, without involving human annotation? In this talk, I will present some preliminary results on possible strategies to achieve this goal: by either leveraging natural text annotations (like sentiment and summaries), by extracting discourse information from pre-trained and fine-tuned language models, or by inducing discourse trees from task-agnostic autoencoding learning objectives. Besides the many remaining challenges and open issues, I will discuss the potential of these novel approaches not only to boost the performance of discourse parsers (NLU) and text planners (NLG), but also lead to more explanatory and useful data-driven theories of discourse.

Biography

Giuseppe Carenini is a Professor in Computer Science and Director of the Master in Data Science at UBC (Vancouver, Canada). His work on natural language processing and information visualization to support decision making has been published in over 140 peer-reviewed papers (including best paper at UMAP-14 and ACM-TiiS-14). Dr. Carenini was the area chair for many conferences including recently for ACL'21 in "Natural language Generation", as well as Senior Area Chair for NAACL'21 in "Discourse and Pragmatics". Dr. Carenini was also the Program Co-Chair for IUI 2015 and for SigDial 2016. In 2011, he published a co-authored book on "Methods for Mining and Summarizing Text Conversations". In his work, Dr. Carenini has also extensively collaborated with industrial partners, including Microsoft and IBM. He was awarded a Google Research Award in 2007 and a Yahoo Faculty Research Award in 2016.

Post-processing Networks: Method for Optimizing Pipeline Task-oriented Dialogue Systems using Reinforcement Learning

Atsumoto Ohashi **Ryuichiro Higashinaka**
Graduate School of Informatics, Nagoya University
ohashi.atsumoto.c0@s.mail.nagoya-u.ac.jp
higashinaka@i.nagoya-u.ac.jp

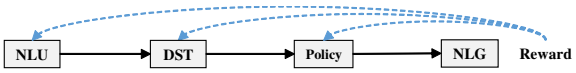
Abstract

Many studies have proposed methods for optimizing the dialogue performance of an entire pipeline task-oriented dialogue system by jointly training modules in the system using reinforcement learning. However, these methods are limited in that they can only be applied to modules implemented using trainable neural-based methods. To solve this problem, we propose a method for optimizing a pipeline system composed of modules implemented with arbitrary methods for dialogue performance. With our method, neural-based components called post-processing networks (PPNs) are installed inside such a system to post-process the output of each module. All PPNs are updated to improve the overall dialogue performance of the system by using reinforcement learning, not necessitating each module to be differentiable. Through dialogue simulation and human evaluation on the MultiWOZ dataset, we show that our method can improve the dialogue performance of pipeline systems consisting of various modules[1].

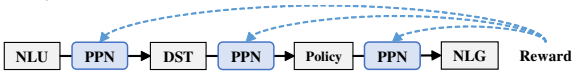
1 Introduction

Task-oriented dialogue systems can be classified into two categories: pipeline systems, in which multiple modules take on a sequential structure, and neural-based end-to-end systems (Chen et al., 2017; Gao et al., 2018; Zhang et al., 2020b).

A typical pipeline system consists of four modules (Zhang et al., 2020b): natural language understanding (NLU), dialogue state tracking (DST), Policy, and natural language generation (NLG). Each module can be implemented individually using various methods (e.g., rule-based and neural-based) (Ultes et al., 2017; Zhu et al., 2020). In a pipeline system, the inputs and outputs of each module are explicit, making it easy for humans to interpret.



(a) Diagram of conventional method. Modules are fine-tuned using RL.



(b) Diagram of proposed method. Each PPN that post-processes output of each module is optimized using RL.

Figure 1: Comparison of conventional and proposed methods

However, since each module is processed sequentially, errors in the preceding module can easily propagate to the following ones, and the performance of the entire system cannot be optimized (Tseng et al., 2021). This results in low dialogue performance of the entire system (Takanobu et al., 2020).

In contrast, neural-based methods can optimize entire neural-based end-to-end systems, which allows for less error propagation than pipeline systems and high dialogue performance (Dinan et al., 2019; Gunasekara et al., 2020). The drawback of these methods is the large amount of annotation data required to train systems (Zhao and Eskenazi, 2016). Compared with pipeline systems, neural-based end-to-end systems are also less interpretable and more difficult to adjust or add functions.

To marry the benefits of both pipeline and end-to-end systems, methods (Liu et al., 2018; Mehri et al., 2019; Lee et al., 2021; Lin et al., 2021) have been proposed for optimizing an entire pipeline system in an end-to-end fashion by using reinforcement learning (RL) (Figure 1(a)). These methods are powerful because they jointly train and fine-tune neural-based implementations of the modules, such as NLU, Policy, and NLG, by using RL. However, these methods may not always be applicable because there may be situations in which modules can only be implemented with rules or the modules' internals cannot be accessed, such as with a Web

[1]Our code is publicly available at https://github.com/nu-dialogue/post-processing-networks

Proceedings of the SIGdial 2022 Conference, pages 1–13
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

API.

With this background, we propose a method for optimizing an entire pipeline system composed of modules implemented in arbitrary methods. We specifically focus on modules that output fixed sets of classes (i.e., NLU, DST, and Policy) and install neural-based components (post-processing networks; PPNs) in the system to post-process the outputs of these modules, as shown in Figure 1(b). Each PPN modifies the output of each module by adding or removing information as necessary to facilitate connections to subsequent modules, resulting in a better flow of the entire pipeline. To enable the appropriate post-processing for the entire system, each PPN uses the states of all modules in the system when executing post-processing. The post-processing of each PPN is optimized using RL so that the system can improve its dialogue performance, e.g., task success. A major advantage of our method is that each module does not need to be trainable since PPNs are trained instead.

To evaluate the effectiveness of our method, we applied PPNs to pipeline systems consisting of modules implemented with various methods (e.g., rule-based and neural-based) on the basis of the MultiWOZ dataset (Budzianowski et al., 2018) and conducted experiments by using dialogue simulation and human participants. The contributions of this study are as follows.

- We propose a method of improving the dialogue performance of a pipeline task-oriented dialogue system by post-processing outputs of modules. Focusing on NLU, DST, and Policy, our method can be applied to various pipeline systems because PPNs do not depend on the implementation method of each module or a combination of modules.

- Dialogue simulation experiments have shown that our method can improve the dialogue performance of pipeline systems consisting of various combinations of modules. Additional analysis and human evaluation experiments also verified the effectiveness of the proposed method.

2 Related Work

Our study is related to optimizing an entire dialogue system with a modular architecture. Wen et al. (2017) proposed a method for implementing all the functions of NLU, DST, Policy, and NLG modules by using neural networks, enabling the entire system to be trained. Lei et al. (2018) incorporated both a decoder for generating belief states (i.e., DST module) and a response-generation decoder (i.e., NLG module) into a sequence-to-sequence model (Sutskever et al., 2014). Zhang et al. (2020a) also proposed a method for jointly optimizing a system that includes three decoders that respectively execute the functions of DST, Policy, and NLG. Liang et al. (2020) extended the method of Lei et al. (2018) by jointly optimizing four decoders that generate user dialogue acts (DAs), belief states, system DAs, and system responses. However, these systems are trained in a supervised manner and require large amounts of data (Liu et al., 2017).

Our study is related to improving the dialogue performance of a pipeline system by using RL. Zhao and Eskenazi (2016) and Li et al. (2017) implemented DST and Policy in a neural model and used the Deep Q Network (Mnih et al., 2013) algorithm to optimize the system to achieve robustness against errors that occur in interactions. Liu et al. (2018) proposed a Policy-learning optimization method for real users by combining supervised learning, imitation learning, and RL. Mehri et al. (2019) proposed a method for training a response-generation model by using RL while using the hidden states of the learned NLU, Policy, and NLG. Methods have been proposed (Lee et al., 2021; Lin et al., 2021) for building a pipeline system with individually trained modules and fine-tuning specific modules by using RL, which significantly improved the performance of the overall system. These methods are powerful because they can fine-tune a system directly through RL. However, they can only be applied to systems consisting of specific differentiable modules implemented using neural-based methods, not to systems consisting of non-differentiable modules. Our method is independent of the module-implementation method, trainability of each module in pipeline systems, and combination of modules.

3 Proposed Method

We developed our method to improve the dialogue performance of an entire pipeline system by optimizing the output of each module through post-processing. Post-processing means modifying the output by adding or removing information from the actual output of the module. With our method,

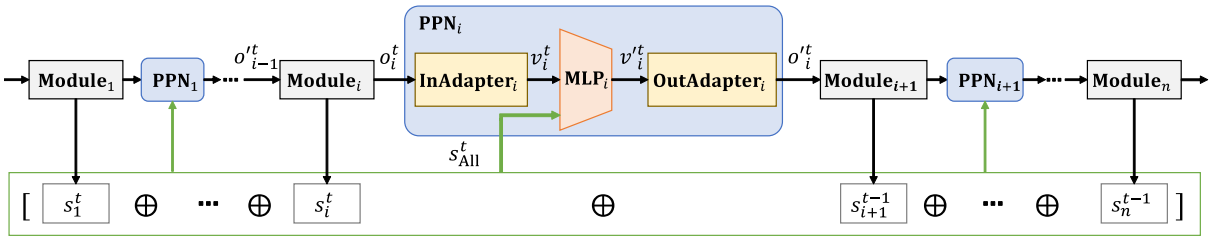


Figure 2: Architecture of our proposed method. Output of each module is post-processed by subsequent PPN. Each PPN has InAdapter to convert output label o of module into multi-binary vector v, MLP to post-process multi-binary vector into v' on basis of v and state s_{All} of all modules, and OutAdapter to restore v' to output label o'.

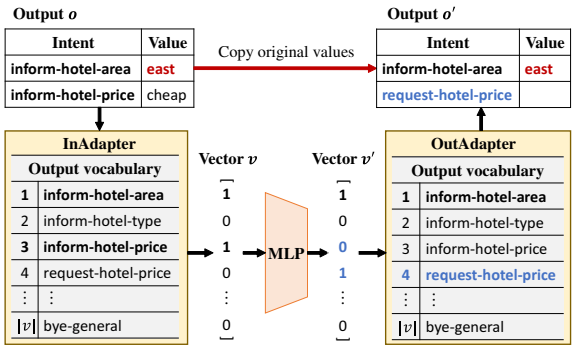


Figure 3: Procedure in which InAdapter converts output label o into vector v and OutAdapter restores vector v' into output label o' by using output vocabulary (in this case, output labels are DAs of NLU). Value information, which cannot be encoded in v, is copied directly from o when creating o'.

each PPN needs to execute post-processing appropriate for all modules so that the entire system can improve overall dialogue performance. With this in mind, each PPN post-processes the target module's output while using the latest states of all modules in the system. Basically, each module's state is the latest output of each module. However, if a module can provide information that represents its state in more detail than the module's output, the PPN also uses that information (see Section 3.1). Figure 2 shows the architecture of PPNs applied to a pipeline system consisting of Module$_1$, ..., Module$_n$.

3.1 Post-processing Algorithm

The following equations describe the steps in which PPN$_i$ post-processes the output o^t_i of Module$_i$ at turn t, as in Figure 2.

$$o^t_i, s^t_i = \text{Module}_i(o'^t_{i-1}) \quad (1)$$
$$v^t_i = \text{InAdapter}_i(o^t_i) \quad (2)$$
$$s^t_{\text{All}} = [s^t_1; ...; s^t_i; s^{t-1}_{i+1}; ...; s^{t-1}_n] \quad (3)$$
$$v'^t_i = \text{MLP}_i([v^t_i; s^t_{\text{All}}]) \quad (4)$$
$$o'^t_i = \text{OutAdapter}_i(v'^t_i) \quad (5)$$

As in a general pipeline system, Module$_i$ first receives the output o'^t_{i-1} of the preceding Module$_{i-1}$ and outputs o^t_i as the result of its processing (Eq. (1)) (e.g., for the NLU module, it receives the user's utterance as input and outputs the user's DAs). At the same time, Module$_i$ outputs its additional information s^t_i obtained in the processing, which is related to the state of Module$_i$ (Eq. (1)). Basically, s^t_i is the same as o^t_i. However, if Module$_i$ can provide more detailed information about its state obtained in the processing (e.g., for the NLU module, it typically outputs confidence scores of predicted user's DAs), Module$_i$ outputs that information as s^t_i.

Next, o^t_i is input to PPN$_i$. In PPN$_i$, InAdapter$_i$ creates a multi-binary vector v^t_i, which is a vector representation of o^t_i (Eq. (2)). The left half of Figure 3 shows a concrete example of an InAdapter converting a module output into a multi-binary vector. The InAdapter$_i$ is created by handcrafted rules using the output vocabulary set of Module$_i$. At the same time as creating v^t_i, $s^t_{\text{All}} = [s^t_1; ...; s^t_i; s^{t-1}_{i+1}; ...; s^{t-1}_n]$, which is a concatenation of the latest states of Module$_1$, ..., Module$_n$, are also created (Eq. (3)). Note that s^{t-1} is used for states of Module$_{i+1}$, ..., Module$_n$ because modules after Module$_i$ have not produced their states in turn t.

The v^t_i and s^t_{All} created thus far are combined and input to multi-layer perceptron (MLP) MLP$_i$, which outputs a multi-binary vector v'^t_i (Eq. (4)). The dimensions of v'^t_i are the same as the vocabulary size of Module$_i$. At this point, the changes in the original vectors v^t and v'^t become the result of post-processing. That is, the dimension, the value in v^t_i of which is 1 and value in v'^t_i of which is 0, is the information deleted by MLP$_i$, and the reverse is the information added by MLP$_i$. Finally, OutAdapter$_i$ converts v'^t_i into o'^t_i, the output label representation of Module$_i$. Some of the value information is directly copied from o^t_i when creating

$o_i'^t$ since these values are not given by $v_i'^t$. If there is no need to fill in the value, it is left empty. The right half of Figure 3 shows a concrete example of an OutAdapter converting a multi-binary vector into a label representation of a module's vocabulary. As with InAdapter$_i$, OutAdapter$_i$ is created by hand-crafted rules using the output vocabulary set of Module$_i$.

At runtime, in the initial turn, the states of some modules that have never processed yet are initialized with zero vector (i.e., $s^0 = \mathbf{0}$). In the subsequent turn t, as mentioned above, PPN$_i$ uses the preceding modules' states $[s_1^t, ..., s_i^t]$ and the succeeding modules' states $[s_{i+1}^{t-1}, ..., s_n^{t-1}]$.

With our method, the MLPs of all PPNs are optimized jointly by using RL via interaction with users (see Section 3.3). To apply PPNs to a system, we only need the vocabulary set of each module to implement an InAdapter and OutAdapter for conversion. Therefore, our method can be applied to both differentiable and non-differentiable implementations of the modules. Since we want first to verify the idea of PPNs, we only used MLPs and focused on NLU, DST, and Policy in this study. Once the verification is complete, we aim to apply PPNs to more complex modules, such as NLG.

3.2 Pre-training with Imitation Learning

It is not easy to optimize an MLP from scratch by using RL. Many studies have shown that model performance can be improved by imitation learning, which is a scheme for learning to imitate the behavior of experts before RL is conducted (Argall et al., 2009; Rajeswaran et al., 2017). We considered the actual output o_i of Module$_i$ to be the behavior of the expert for PPN$_i$ and conducted supervised learning so that PPN$_i$ copies o^t before RL. This should allow each PPN to focus only on "how to modify the module's output o" during RL.

With our method, a pipeline system consisting of Module$_1$,..., Module$_n$ first executes dialogue sessions for sampling training data. In each dialogue, we sample the $[s_{\mathrm{All}}, v]$ of each module for all turns. At this stage, no PPNs execute post-processing, and no MLPs are used. When training MLPs by imitation learning, supervised learning is carried out using the sampled data. We train all MLPs to execute a multi-labeling task in which the input is $[v; s_{\mathrm{All}}]$ and the output label is v. Binary cross-entropy is used to update the MLP to minimize the difference between v and $v' = \mathrm{MLP}([v; s_{\mathrm{All}}])$.

3.3 Optimization with Reinforcement Learning

The goal with PPNs is to improve dialogue performance (e.g., task success) by each PPN post-processing the output of each module. Therefore, the MLP of each PPN needs to be optimized using RL for maximizing the rewards related to dialogue performance. We use proximal policy optimization (PPO) (Schulman et al., 2017) as the RL algorithm, which is a stable and straightforward policy-gradient-based RL algorithm.

The following steps show the learning algorithm of a PPN for each iteration:

Step. 1 The pipeline system with PPNs interacts with a user. Each PPN post-processes and samples the s_{All}^t, v^t, v'^t, and reward r^t of each MLP in turn t. The sampled $(s_{\mathrm{All}}^t, v^t, v'^t, r^t)$ are added to the post-processing history (called *trajectory*) of each PPN. As an r^t, we give the same value to all PPNs. These trials are repeated until the trajectory reaches a predetermined size (called *horizon*).

Step. 2 The PPN to be updated in this iteration is selected on the basis of the *PPN-selection strategy*, which is a rule for selecting PPNs to be updated in each iteration. We have three strategies described in the next paragraph.

Step. 3 The MLPs of the PPNs selected in Step. 2 are updated using the PPO algorithm. Each MLP is updated for multiple epochs using the trajectory sampled in Step. 1 as training data.

Since it is not apparent which modules' PPN should be updated and in what order, we prepared the following three PPN-selection strategies: ALL (select all PPNs in every iteration), RANDOM (randomly select one or more PPNs in each iteration), and ROTATION (select one PPN at each iteration in order). In the following experiments, we examined which strategy is the best.

4 Experiments

To confirm the effectiveness of our method, we applied PPNs to several different pipeline systems and evaluated dialogue performance using dialogue simulation. We also carried out a human evaluation.

4.1 Dataset

We evaluated PPNs using modules and a user simulator implemented using the MultiWOZ dataset (Budzianowski et al., 2018), which is a task-oriented dialogue dataset between a clerk and tourist at an information center. MultiWOZ contains 10,438 dialogues; one to three domains (seven domains in total in the dataset) appear simultaneously in each dialogue.

4.2 Platform and User Simulator

ConvLab-2[2] (Zhu et al., 2020) is a platform for multi-domain dialogue systems, which provides pre-implemented models of each module in the pipeline system and tools for end-to-end evaluation of the dialogue system.

We used the user simulator implemented in ConvLab-2. The simulator interacts with the dialogue system in natural language on the basis of the user goal given for each dialogue session. The simulator consists of a BERT (Devlin et al., 2019)-based NLU (Chen et al., 2019), an agenda-based Policy (Schatzmann et al., 2007), and a template-based NLG. The agenda-based Policy models a user's behavior in MultiWOZ by using a stack-like agenda created using hand-crafted rules. A user goal for each dialogue is randomly generated: the domains are randomly selected from one to three domains (out of all seven domains) on the basis of the domains' frequency in MultiWOZ; the slots are also randomly selected on the basis of the slots' frequency in MultiWOZ.

4.3 Evaluation Metrics

In evaluating each dialogue, we used the number of turns[3] (Turn) to measure the efficiency of completing each dialogue; the smaller the Turn is, the better the system performance. We also measured whether the system responds to the requested slot by the user without excess or deficiency (Inform F1) and whether the entity presented by the system met the condition of the user goal (Match Rate). We also used Task Success as a result of Match Rate and Inform Recall being equal to 1 within 20 turns. The above four metrics are the major ones for dialogue evaluation and have been used in many studies using ConvLab-2 (Li et al., 2020; Takanobu et al., 2020; Hou et al., 2021).

4.4 Implementation

4.4.1 System Configurations

To select the modules that make up a pipeline system, we referred to Takanobu et al. (2020), who developed and evaluated various combinations of modules using ConvLab-2. For the models of each module (NLU, DST, Policy, and NLG), we included both classical rule-based and recent neural-based models. Note that, since this study focused on whether PPNs can be used to optimize pipeline systems consisting of non-trainable modules, we did not update modules even if the modules may be trainable. Each of the models[4] we prepared are as follows.

NLU We used BERT NLU (Chen et al., 2019) for the NLU module. This model estimates DAs by tagging which domain-intent-slot each token in a user utterance represents by using a pre-trained BERT (Devlin et al., 2019). The InAdapter/OutAdapter are created using the DA set defined in BERT NLU (see Figure 3 for an illustration of an InAdapter-processing example by using a DA set). We used the estimated probabilities of each DA as BERT NLU's state s.

DST We used two models for the DST module: Rule DST (Zhu et al., 2020) and TRADE (Wu et al., 2019). Rule DST updates the dialogue state consisting of belief state, database search results, current user DAs, and previous system DAs at each turn by directly using the DAs estimated by the NLU. On the contrary, TRADE is a neural-based model that directly extracts slot-value pairs and generates belief states using the dialogue history as input. For DST modules, a belief state is subject to post-processing. Therefore, we created an InAdapter/OutAdapter on the basis of the slot types defined in the belief state on ConvLab-2. As states of Rule DST and TRADE, an entire dialogue state is converted into a multi-binary vector by using a vectorizer implemented in ConvLab-2.

Policy We used four models for the Policy module: Rule Policy (Zhu et al., 2020), MLE Policy, PPO Policy (Schulman et al., 2017), and LaRL Policy (Zhao et al., 2019). Rule Policy is a model based on hand-crafted rules. MLE Policy is a model trained on state-action pairs in MultiWOZ using supervised learning. PPO Policy is

[2] https://github.com/thu-coai/ConvLab-2

[3] One user utterance and its system response form one turn.

[4] For models, we used the best ones provided by ConvLab-2 as of October 20, 2021

Module	Models	$\|s\|$	$\|v\|$
NLU	BERT	175	175
DST	Rule, TRADE	340	24
Policy	Rule, MLE, PPO	209	209
	LaRL	0	0
NLG	Template, SC-LSTM	0	0

Table 1: Dimensions $|s|$ of state s output from each module and $|v|$ of vector v processed by PPN of each module. Number of output vocabularies defined for each module and $|v|$ are equal.

a fine-tuned model based on MLE Policy using the PPO RL algorithm. Unlike the other Policy models, LaRL Policy is an LSTM-based model trained to directly generate system utterances instead of system DAs by using RL. We created an InAdapter/OutAdapter using the DA set defined in each model. For states of MLE Policy and PPO Policy, we used the estimated probability of each DA. For Rule Policy's state, we used a binary vector representation of DAs. Since the output of LaRL is a natural language, it was not subject to post-processing in this study.

NLG We used two models for the NLG module: Template NLG and SC-LSTM (Wen et al., 2015). Template NLG creates system responses by inserting values into templates of utterances manually created in advance for each DA. SC-LSTM is an LSTM-based model that generates utterances on the basis of DAs. For the same reason as for LaRL Policy, we did not implement PPNs for Template NLG and SC-LSTM in these experiments.

Table 1 shows the dimensions of each module's state s described above and the number of dimensions of the multi-binary vector o of each PPN (i.e., the vocabulary of each module). Note that for the DST modules, the dimensions of s and v are different. This is because s is a vector representation of a dialogue state, which includes a belief state, database search results, user's DAs, etc., and v is a vector representation of a belief state only.

4.4.2 Training

Throughout all experiments, the data used for imitation learning of each pipeline system was sampled by simulating 10,000 turns, corresponding to approximately 1,000 dialogue sessions. In RL for each system, we trained 200 iterations, where one iteration consists of approximately 100 dialogue sessions. Following Takanobu et al. (2019), we gave a reward of -1 for each turn, and when the task was a success, we gave the maximum number of turns $\times$ 2 at the end of the dialogue session, i.e., 40 in our case. See Section A.1 of the appendix for more training details.

To test each system, we ran 1,000 dialogues using a system that achieved the best Task Success during the RL training. Throughout all experiments, we trained with five different random seeds and reported the average of their scores as the final performance.

PPN-selection strategy	Success	Inform	Match	Turn
ALL	64.2	**71.9**	76.6	9.20
RANDOM	**66.1**	71.5	**78.7**	**8.61**
ROTATION	60.4	70.5	73.2	9.10

Table 2: Performance after PPN training with each PPN-selection strategy

4.5 Experimental Procedure

We conducted four experiments. The first experiment was conducted to determine which of the PPN-selection strategies (see Section 3.3) is appropriate. We used a combination of BERT NLU, Rule DST, MLE Policy, and Template NLG as the system configuration. The reasons for using this combination are that (1) the Task Success of a system composed of this module combination is around 50%. Therefore, it would be easy to understand the impact of the PPNs, and (2) MLE Policy is used as the initial weight in many RL methods (Takanobu et al., 2019; Li et al., 2020), making it a reasonable starting point for RL. The second experiment was conducted to verify whether the PPNs work for any combination of modules; we combined some of the modules described in Section 4.4.1 to build pipeline systems and applied PPNs. The third experiment was conducted to investigate the contribution of the PPN of each module and s_{All} to the overall performance of the system. The final experiment was a human evaluation; we examined whether the proposed method is effective not only for a simulator but also for humans.

4.6 Comparison of Post-processing-network-selection Strategies

Figure 4 shows the learning process in the three PPN-selection strategies. Task Success and Inform F1 at 50 iterations show that ALL reached the highest score about 100 iterations earlier than RANDOM and ROTATION. This is a reasonable result

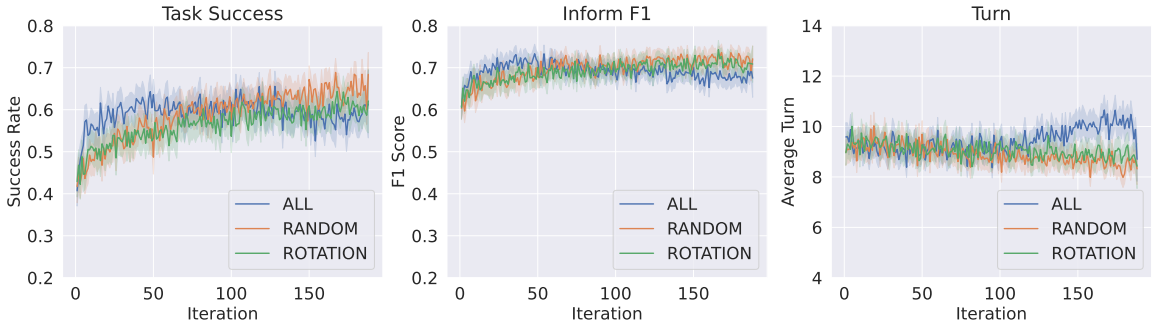


Figure 4: Scores of each evaluation metric in learning process with three PPN-selection strategies

System	Model Combination				w/ PPN	Task Success	Inform F1	Match Rate	Turn
	NLU	DST	Policy	NLG					
SYS-RUL	BERT	Rule	Rule	Template		84.1	87.4	90.2	5.92
					✓	84.0	86.3	**92.4**	6.33
SYS-MLE	BERT	Rule	MLE	Template		43.3	62.4	27.8	9.03
					✓	**66.1**	**71.5**	**78.7**	**8.61**
SYS-PPO	BERT	Rule	PPO	Template		54.9	65.5	55.2	8.41
					✓	**68.8**	**72.1**	**77.8**	**8.37**
SYS-SCL	BERT	Rule	Rule	SC-LSTM		38.3	57.5	56.7	13.53
					✓	**44.2**	**71.7**	**71.8**	**11.04**
SYS-TRA	TRADE		Rule	Template		19.0	45.6	36.4	12.08
					✓	18.8	**49.2**	31.6	12.14
SYS-LAR	BERT	Rule	LaRL			21.6	44.9	27.6	13.24
					✓	**23.9**	**50.9**	**34.1**	**12.77**

Table 3: Combination of models for each pipeline system and scores before and after applying PPNs to each system. 'w/ PPN' indicates whether PPNs are applied to the system. Scores that have been improved using PPNs are in bold.

since the number of updates for each MLP in ALL was up to four times that for the other strategies. However, it was unstable after 50 iterations, and the scores of Task Success, Inform F1, and Turn all worsened as the learning process progressed. This is probably because the gradients of each MLP were calculated simultaneously in the PPO update algorithm, which caused each MLP to update in a different gradient direction, making it difficult for each MLP to coordinate with one another.

Although the learning speed of ROTATION and RANDOM was slow, all metrics consistently improved. Turn and Inform F1 also showed stable improvements compared with ALL. For RANDOM and ROTATION, each MLP computed its gradient after the other MLPs computed and updated their gradients one by one, which probably prevented significant discrepancies among MLPs and stabilized learning.

Table 2 shows the final performance of each strategy. RANDOM outperformed ALL in all the final scores, and ROTATION was inferior to ALL in Task Success, Inform F1, and Match Rate. Since the learning was stable and the final performance was generally better than the other strategies, we decided to use RANDOM in the following experiments.

4.7 Comparison of Model Combinations

We built six pipeline systems with different model combinations. Table 3 summarizes the comparison of the scores when PPNs were applied to each system. For a fair comparison, systems without PPNs were also evaluated on the average scores[5] of 1,000 dialogues conducted with five different random seeds.

Table 3 shows that Task Success improved for most of the systems. In addition, all systems improved in Inform F1 or Match Rate. These results indicate that post-processing with PPNs can improve the dialogue performance of a pipeline system without touching the module internals. However, neither Task Success nor Turn improved for SYS-RUL and SYS-TRA. The common feature of these two systems is that they use Rule Policy and Template NLG. These modules are carefully designed by hand and originally have high accu-

[5]Although we used the latest models implemented in ConvLab-2, we could not reproduce the scores reported in https://github.com/thu-coai/ConvLab-2#end-to-end-performance-on-multiwoz

System	w/ s_{All}	Success	Inform	Match	Turn
SYS-MLE		43.3	62.4	27.8	9.03
$+\text{PPN}_{\text{NLU}}$		59.6	**73.1**	65.8	9.59
$+\text{PPN}_{\text{DST}}$		46.7	65.1	36.7	9.41
$+\text{PPN}_{\text{Policy}}$		59.9	67.3	67.9	9.20
$+\text{PPN}_{\text{All}}$		59.7	68.0	69.9	9.84
$+\text{PPN}_{\text{NLU}}$	✓	62.2	72.1	64.0	9.36
$+\text{PPN}_{\text{DST}}$	✓	47.9	66.1	40.2	9.21
$+\text{PPN}_{\text{Policy}}$	✓	65.8	67.6	76.9	**8.56**
$+\text{PPN}_{\text{All}}$	✓	**66.1**	71.5	**78.7**	8.61
+Fine-tuned Policy		71.9	74.3	80.4	7.88

Table 4: Impact analysis of PPNs. Subscripts (i.e., NLU, DST, Policy, and All) indicate that PPN was applied to that one specific module or all modules. 'w/ s_{All}' indicates whether s_{All} was used. Row of Fine-tuned Policy shows scores when SYS-MLE's Policy was fine-tuned using RL.

racy, leading to little room for improvement in this configuration.

In general, there were large differences in performance among the systems regardless of whether PPN was used. As mentioned above, this is due to the performance differences among the modules comprising the systems. For example, SYS-RUL is considered to have significantly higher performance than the other systems due to the use of elaborately designed rules and templates.

4.8 Impact of Post-processing Networks

We investigated the impact of each module's PPN and s_{All}. We used SYS-MLE as a base configuration for this experiment since its performance was most improved with our method (see Table 3); we considered it appropriate to measure the impact of PPNs. In Table 4, the results of applying PPNs to only one of the NLU, DST, and Policy are shown, as well as the results of applying PPNs without using s_{All}. The system performance consistently improved when only a single module's PPN was applied. In particular, $+\text{PPN}_{\text{Policy}}$ achieved the best performance (Task Success improved by more than 20%), indicating that the PPN of Policy contributed the most to dialogue performance. When s_{All} was not used, most of the scores decreased. This indicates that each PPN can execute post-processing more appropriately by using the states of all modules in the system.

To confirm the degree of performance improvement achieved with the PPNs, the method of fine-tuning the modules by using RL was used as the upper bound of post-processing. Only the Policy module was fine-tuned, as is common with conventional methods (Liu et al., 2018; Lin et al., 2021).

System	Success	Turn	Und.	App.	Sat.
SYS-MLE	39.0	11.0	2.93	3.12	2.46
$+\text{PPN}_{\text{NLU}}$	53.7	11.1	3.10	3.37	2.93
$+\text{PPN}_{\text{DST}}$	60.0	10.4	**3.30**	**3.43**	**3.28***
$+\text{PPN}_{\text{Policy}}$	**62.5***	**8.20***	2.93	3.03	3.00
$+\text{PPN}_{\text{All}}$	57.5	9.00	2.83	3.00	2.95

Table 5: Results of human evaluation for each system configuration. Asterisks indicate statistically significant differences ($p < 0.05$) over SYS-MLE.

The bottom row of Table 4 shows the results when the Policy of SYS-MLE was fine-tuned by PPO (Schulman et al., 2017) (see Section A.2 of the appendix for training details). The difference between $+\text{PPN}_{\text{All}}$ and +Fine-tuned Policy is small with 5.8%. This is a promising result considering that our proposed method does not touch on the internal architecture of Policy.

4.9 Human Evaluation

Five systems (SYS-MLE and four systems with our proposed method, i.e., $+\text{PPN}_{\text{NLU}}$, $+\text{PPN}_{\text{DST}}$, $+\text{PPN}_{\text{Policy}}$, and $+\text{PPN}_{\text{All}}$) in Table 4 were used for the human evaluation. Not that s_{All} was used in all four systems. About forty Amazon Mechanical Turk (AMT) crowd workers were recruited to interact with each of the five systems and judged on Task Success. As in the simulation experiments (see Section 4.2), user goals were randomly generated for each dialogue. After the interaction, the workers also evaluated the system's ability to understand the language (Und.), accuracy of the system's responses (App.), and overall satisfaction with the interaction (Sat.) on a 5-point Likert scale. See Section B of the appendix for the procedures taken by the workers.

Table 5 shows the results. All four systems with our proposed method performed better than SYS-MLE, which is similar to the result in Table 4. Wilcoxon rank-sum tests were conducted using the top score in each evaluation metric and the score of SYS-MLE, and statistically significant differences were confirmed for Task Success and Turn in $+\text{PPN}_{\text{Policy}}$ and interaction satisfaction in $+\text{PPN}_{\text{DST}}$. In contrast, there were no significant differences in scores for language understanding and responses' appropriateness. This is probably because RL was conducted with rewards that only relied on Task Success and Turn. The performance of $+\text{PPN}_{\text{NLU}}$ did not improve as much as in Table 4. A possible reason is the overfitting of $+\text{PPN}_{\text{NLU}}$ with the user simulator. The same over-

fitting might have occurred in the NLU's PPN in $+\mathrm{PPN}_{\mathrm{All}}$, which resulted in a smaller improvement in scores of $+\mathrm{PPN}_{\mathrm{All}}$.

We also investigated how PPNs executed post-processing by analyzing the actual dialogue logs collected in this experiment. A specific case study is described in Section C of the appendix. Generally, in the dialogue of $+\mathrm{PPN}_{\mathrm{Policy}}$, we observed that $\mathrm{PPN}_{\mathrm{Policy}}$ added necessary DAs when the original Policy failed to output them.

5 Conclusions and Future Work

We proposed a method for optimizing pipeline dialogue systems with post-processing networks (PPNs). Through dialogue simulation and human evaluation experiments on the MultiWOZ dataset, we showed that the proposed method is effective for a pipeline system consisting of modules with various models.

For future work, we plan to design more sophisticated rewards in RL such as module-specific rewards. We also plan to extend PPNs to handle natural language generation by implementing them using Transformer-based models. We are also considering to apply PPNs to modules dealing with speech recognition and multi-modal processing.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 19H05692. We used the computational resource of the supercomputer "Flow" at Information Technology Center, Nagoya University. We thank Yuya Chiba and Yuiko Tsunomori for their helpful comments and feedback. Thanks also go to Ao Guo for his advice on the human evaluation experiment.

References

Brenna D. Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, pages 469–483.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, pages 25–35.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for Joint Intent Classification and Slot Filling. *arXiv preprint arXiv:1902.10909*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W. Black, Alexander Rudnicky, Jason Williams, Joelle Pineau, Mikhail Burtsev, and Jason Weston. 2019. The Second Conversational Intelligence Challenge (ConvAI2). *arXiv preprint arXiv:1902.00098*.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TPRO. *arXiv preprint arXiv:2005.12729*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural Approaches to Conversational AI. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374.

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, Dilek Hakkani-Tür, Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Minlie Huang, Jianfeng Gao, Shikib Mehri, Yulan Feng, Carla Gordon, Seyed Hossein Alavi, David Traum, Maxine Eskenazi, Ahmad Beirami, Cho Eunjoon, Paul A. Crook, Ankita De, Alborz Geramifard, Satwik Kottur, Seungwhan Moon, Shivani Poddar, and Rajen Subba. 2020. Overview of the Ninth Dialog System Technology Challenge: DSTC9. *arXiv preprint arXiv:2011.06486*.

Zhengxu Hou, Bang Liu, Ruihui Zhao, Zijing Ou, Yafei Liu, Xi Chen, and Yefeng Zheng. 2021. Imperfect also Deserves Reward: Multi-Level and Sequential Reward Modeling for Better Dialog Management. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2993–3001.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Hwaran Lee, Seokhwan Jo, Hyungjun Kim, Sangkeun Jung, and Tae-Yoon Kim. 2021. SUMBT+LaRL: Effective Multi-Domain End-to-End Neural Task-Oriented Dialog System. *IEEE Access*, pages 116133–116146.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-Sequence Architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1437–1447.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-End Task-Completion Neural Dialogue Systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 733–743.

Ziming Li, Sungjin Lee, Baolin Peng, Jinchao Li, Julia Kiseleva, Maarten de Rijke, Shahin Shayandeh, and Jianfeng Gao. 2020. Guided Dialogue Policy Learning without Adversarial Learning in the Loop. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2308–2317.

Weixin Liang, Youzhi Tian, Chengcai Chen, and Zhou Yu. 2020. MOSS: End-to-End Dialog System Framework with Modular Supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8327–8335.

Zichuan Lin, Jing Huang, Bowen Zhou, Xiaodong He, and Tengyu Ma. 2021. Joint System-Wise Optimization for Pipeline Goal-Oriented Dialog System. *arXiv preprint arXiv:2106.04835*.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2017. End-to-End Optimization of Task-Oriented Dialogue Model with Deep Reinforcement Learning. *arXiv preprint arXiv:1711.10712*.

Bing Liu, Gokhan Tür, Dilek Hakkani-Tür, Pararth Shah, and Larry Heck. 2018. Dialogue Learning with Human Teaching and Feedback in End-to-End Trainable Task-Oriented Dialogue Systems. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2060–2069.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured Fusion Networks for Dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*.

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. 2017. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. *arXiv preprint arXiv:1709.10087*.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 149–152.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of Advances in neural information processing systems*, pages 3104–3112.

Ryuichi Takanobu, Hanlin Zhu, and Minlie Huang. 2019. Guided Dialog Policy Learning: Reward Estimation for Multi-Domain Task-Oriented Dialog. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 100–110.

Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation. In *Proceedings of the 21st Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 297–310.

Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne. 2021. Transferable Dialogue Systems and User Simulators. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 152–166.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve Young. 2017. PyDial: A Multi-domain Statistical Dialogue System Toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 438–449.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020a. Task-Oriented Dialog Systems That Consider Multiple Appropriate Responses under the Same Context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9604–9611.

Zheng Zhang, Ryuichi Takanobu, Qi Zhu, MinLie Huang, and XiaoYan Zhu. 2020b. Recent advances and challenges in task-oriented dialog systems. *Science China Technological Sciences*, pages 1–17.

Tiancheng Zhao and Maxine Eskenazi. 2016. Towards End-to-End Learning for Dialog State Tracking and Management using Deep Reinforcement Learning. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–10.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents with Latent Variable Models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1208–1218.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149.

A Training Details

A.1 Training Post-processing Networks

Model All MLPs of the PPNs for all modules are implemented in three layers: one input layer, one hidden layer, and one output layer, and the dimensionality of the hidden layer is 128 for all layers. The number of dimensions of the input and output layers are $|o| + |s_{\text{All}}|$ and $|o_i|$, respectively. The activation functions are all ReLUs.

Imitation Learning The sampled data of 10,000 turns were split as $\text{training} : \text{validation} = 8 : 2$. All MLPs were trained on a batch size of 32 for 20 epochs using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 1e-3. The weights at the epoch with the highest accuracy for validation were used for the following RL.

Reinforcement Learning The hyperparameters shown in Table 6 were determined with reference to the implementation of PPO in ConvLab-2. We used Generalized Advantage Estimation (Schulman et al., 2015). Referring to Engstrom et al. (2020), the learning rate was annealed linearly in accordance with the current iteration. The computational resource used was a single NVIDIA Tesla V100 SXM2 GPU with 32GB RAM. In training, the trajectory was sampled in parallel by eight processes, and it took 5 to 17 hours, depending on the system, to complete the training of 200 iterations.

A.2 Fine-tuning of Policy

The MLE Policy of SYS-MLE in Section 4.8 was fine-tuned with PPO using the same user simulator used for training PPNs. The hyperparameters used for training were the same as those used in ConvLab-2, as shown in Table 6. To evaluate the fine-tuned Policy, training and testing (consisting of 1,000 dialogue sessions) were conducted with five random seeds.

Hyperparameters		PPN	Fine-tuned Policy
Number of iterations		200	200
Batch size		1024	1024
Epoch		5	5
Mini batch size		32	32
Discount factor γ		0.99	0.99
GAE factor λ		0.95	0.95
Optimizer	policy net	Adam	RMSprop
	value net		Adam
Learning rate	policy net	1e-4	1e-4
	value net		5e-5

Table 6: Hyperparameter settings in PPO

B Details of Human Evaluation

Referring to Takanobu et al. (2020), we designed the following experimental procedure. First, each worker is presented with an instruction for a randomly generated user goal. Next, the user interacts with one of the five systems in Table 5 for up to 20 turns. Workers determine whether the interaction succeeded or failed within 20 turns; after 20 turns, the interaction is automatically marked as failed. To ensure the quality of the workers, several qualifications were set; the eligible workers should (1) reside in an English-speaking country, (2) have a task accomplishment number on AMT greater than 10, (3) have a task-approval rate greater than 90%, and (4) correctly answer all the common sense questions (total of five questions) we designed. The time limit for the task was 10 minutes, and the reward was $1.7. To account for workers who may cancel the task amid the dialogue session, more than 40 workers were recruited for each system. At the end, 41 workers for SYS-MLE and $+\text{PPN}_{\text{NLU}}$ and 40 workers for $+\text{PPN}_{\text{DST}}$, $+\text{PPN}_{\text{Policy}}$, and $+\text{PPN}_{\text{All}}$ participated in the experiment.

C Case Study

Tables 7 and 8 show the user goal and example of a dialogue session between a worker and $+\text{PPN}_{\text{Policy}}$, respectively, in the human evaluation experiment. Table 8 also includes the outputs of the system's Policy and the post-processing result of PPNs. In S1, the original MLE Policy did not generate any DAs. At this time, $\text{PPN}_{\text{Policy}}$ generated additional DAs informing the user that "free parking is available at the hotel found in the database" and asked if there was anything else the user needs. In S2, the user requested a reservation, but MLE Policy could not make the reservation. In contrast, $\text{PPN}_{\text{Policy}}$ successfully made the reservation and provided a booking reference number.

Domain	Task	Slot
Hotel	Info	Area = West Parking = Yes
	Book	Day = Sunday Time = 11:00 People = 1
Restaurant	Fail Info	Area = West Food = Mediterranean
	Info	Area = West Food = Thai
	Reqt	Postcode Address

Table 7: User goal used in Table 8

U1	I need a hotel in the west with free parking for 7 people on saturday for 3 nights
S1	**Policy outputs:** **PPN$_{\text{Policy}}$ outputs:** Inform-Hotel-Parking-yes, reqmore-general **System response:** It offers free parking . Is there anything else I can help you with today ?
U2	can i book a room for 7 people for 3 nights on Saturday
S2	**Policy outputs:** reqmore-general-none **PPN$_{\text{Policy}}$ outputs:** Book-Booking-Ref-00000017, reqmore-general **System response:** Here is the booking information : Booking was successful . Reference number is : 00000017. You are welcome . Is there anything else I can help you with today ?
U3	I need a mediterranean restaurant in the west
S3	**Policy outputs:** NoOffer-Restaurant-Area-west, NoOffer-Restaurant-Food-Mediterranean, NoOffer-Restaurant, Request-Restaurant-Food **PPN$_{\text{Policy}}$ outputs:** Inform-Taxi-Car-ford, NoOffer-Restaurant-Food-Mediterranean, NoOffer-Restaurant, Request-Restaurant-Food **System response:** A ford is booked for you . There are no mediterranean restaurants unfortunately . No , I ' m sorry . The search did n't pull up any matches . What type of food would you like ?
U4	I need a thai restaurant in the west
S4	**Policy outputs:** Inform-Restaurant-Name-sala thong, Inform-Restaurant-Area-west, Inform-Restaurant-Food-thai, Inform-Restaurant-Price-expensive, Inform-Booking **PPN$_{\text{Policy}}$ outputs:** Inform-Restaurant-Name-sala thong, Inform-Restaurant-Area-west, Inform-Restaurant-Food-thai, Inform-Restaurant-Price-expensive, Inform-Hotel-Internet-yes, Inform-Booking **System response:** How does the sala thong sound ? It is in the west area . That is a thai restaurant . They are expensive. The place provides free wifi . Would you like for me to try and make a reservation ?
U5	Yes
S5	**Policy outputs:** Inform-Booking, Request-Booking-Day, Request-Booking-People, Request-Booking-Time, reqmore-general **PPN$_{\text{Policy}}$ outputs:** Inform-Hospital-Phone-01223216297, Inform-Booking, Request-Booking-Day, Request-Booking-People, Request-Booking-Time, reqmore-general **System response:** The phone is 01223216297 . Would you like me to book it ? What day would you like that reservation ? How many people will be with you ? What time would you like me to make your reservation ? You are welcome . Is there anything else I can help you with today ?
U6	yes. what is the address?
S6	**Policy outputs:** Request-Booking-Day, Request-Booking-Time, reqmore-general **PPN$_{\text{Policy}}$ outputs:** Inform-Hotel-Addr-144 thornton road, Inform-Hotel-Post-cb30nd, Book-Booking-Ref-00000017, Request-Booking-Day, Request-Booking-People, reqmore-general **System response:** They are located at 144 thornton road. The postal code for that hotel is cb30nd . Here is the booking information : Booking was successful . Reference number is : 00000017. What day would you like your booking for ? How many people will be with you ? Is there anything else I can help you with today ?
U7	No
S7	**Policy outputs:** reqmore-general **PPN$_{\text{Policy}}$ outputs:** reqmore-general **System response:** Did you need any further assistance today ?
U8	**Success**

Table 8: Example of dialogue history between worker and +PPN$_{\text{Policy}}$ in human evaluation experiment. DAs appropriately added by PPN$_{\text{Policy}}$ are in blue, and those inappropriately added are in red.

Reducing Model Churn: Stable Re-training of Conversational Agents

Christopher Hidey **Fei Liu** **Rahul Goel**
Google Assistant
{chrishidey, liufe, goelrahul}@google.com

Abstract

Retraining modern deep learning systems can lead to variations in model performance even when trained using the same data and hyper-parameters by simply using different random seeds. This phenomenon is known as *model churn or model jitter*. This issue is often exacerbated in real world settings, where noise may be introduced in the data collection process. In this work we tackle the problem of stable retraining with a novel focus on structured prediction for conversational semantic parsing. We first quantify the model churn by introducing metrics for *agreement* between predictions across multiple re-trainings. Next, we devise realistic scenarios for noise injection and demonstrate the effectiveness of various churn reduction techniques such as ensembling and distillation. Lastly, we discuss practical trade-offs between such techniques and show that co-distillation provides a sweet spot in terms of churn reduction with only a modest increase in resource usage.

1 Introduction

Deep learning systems can perform inconsistently across multiple runs, even when trained on the same data with the same hyper-parameters. Deployment in real-world environments presents a challenge, where constantly changing production systems require frequent re-training of models. For a conversational semantic parsing system such as Google Assistant or Amazon Alexa, where the goal is to convert users' commands into executable forms, this erratic behavior can have some unfortunate practical consequences. Some examples include irreproducibility, which limits the ability to make meaningful comparisons between experiments (Dodge et al., 2019, 2020), bias, which creates credibility issues if systems consistently struggle with members of a certain class (D'Amour et al., 2020), and user frustration, which can arise due to unpredictable interactions over time.

Query	will i need snow tires to drive the sierra nevada mountains this afternoon?
Model Run 1	[in:get_weather [sl:weather_attribute snow tires] [sl:location sierra mountains] [sl:date_time this afternoon]]
Model Run 2	[in:get_info_road_condition [sl:road_condition snow tires] [sl:location sierra mountains] [sl:date_time this afternoon]]

Table 1: An example from the TOPv2 dataset (Chen et al., 2020a) where two model runs re-trained on the same data with the same hyper-parameters make different predictions. Only the first matches the gold target, but the second has an incorrect intent and slot.

The root cause of widely divergent behavior is underspecification (D'Amour et al., 2020), where there are many equivalent but distinct solutions to a problem. Non-determinism in model training (e.g. different data orders or weight initializations) can lead to finding local minima that obtain the same measurements on a held-out test set but make different predictions (also known as *model churn*).

Even in an academic setting, controlling for all non-determinism is unrealistic - Table 1 provides an example of churn from the TOPv2 dataset (Chen et al., 2020a). In this case, re-training the same model twice with the same data and hyper-parameters results in two different predictions for the given query. While at the token level the slots and arguments overlap, the intents are different, resulting in a drastically different user experience. In this scenario, the dataset is static and yet we still observe model churn. In a real-world setting, the dataset may be constantly changing and noisy, necessitating frequent re-training. The goal, then, is to maintain consistency even in this scenario.

We thus conduct experiments to evaluate and reduce churn across multiple model re-training runs. Our contributions are as follows:

1. We extend the notion of *model churn* to structured prediction. To this end, we introduce

Proceedings of the SIGdial 2022 Conference, pages 14–25
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

new metrics for *agreement* and *exact match agreement* (Section 3).

2. We show that techniques such as ensembling (Dietterich, 2000) and distillation/co-distillation (Hinton et al., 2015; Kim and Rush, 2016; Anil et al., 2018), described in Section 4, reduce churn on the TOP (Gupta et al., 2018), TOPv2 (Chen et al., 2020a), MTOP (Li et al., 2021), and SNIPS (Coucke et al., 2018) datasets (Section 6).

3. We explore the effects of model churn in "real-world" environments, conducting experiments with a smaller model and two types of simulated noise (random and systematic)[1] to represent various sources of error (Sections 5 and 6).

4. We make practical recommendations based on resource usage (number of parameters) in addition to accuracy and agreement and observe that co-distillation with label smoothing provides the best tradeoff (Section 7).

To the best of our knowledge, we are the first to study model churn for the structured prediction task of spoken language understanding (SLU).

2 Background and Related Work

The problem of *model churn* (Milani Fard et al., 2016), defined as the difference in predictions observed across runs when re-training models, has traditionally been studied for classification tasks. In contrast with previous work, we study the problem of model churn for structured prediction, specifically for SLU. Shamir and Coviello (2020) introduced "anti-distillation" to increase diversity in ensemble predictions and Shamir et al. (2020) introduced the smooth-relu activation function; however, in our initial experiments we did not find significant improvement using these methods when applied to structured prediction. Other work has explored forms of smoothing to reduce churn, either by computing soft labels using the nearest neighbors (Bahri and Jiang, 2021) or by weighting the loss term of individual examples using the predicted probabilities from a teacher model (Jiang et al., 2022). As these methods were developed for classification, we leave the task of adapting them to structured prediction for future work.

Other research has focused on related problems such as reproducibility (McCoy et al., 2020) and calibration (Guo et al., 2017; Mosbach et al., 2021). Nie et al. (2020) argue that this phenomenon is due to underlying task complexity and annotator disagreement. D'Amour et al. (2020) claim that reproducibility is primarily due to underspecification, where there are many distinct solutions to the same problem. While these problems are related to churn, both reproducibility and calibration metrics are computed relative to a target, rather than accounting for agreement across re-training runs.

It has been well known that ensembling increases reproducibility and model calibration (Hansen and Salamon, 1990; Lakshminarayanan et al., 2017). Since ensembles increase inference times, distillation (Hinton et al., 2015) is commonly used to train a student model with similar inference resource usage. Reich et al. (2020) show that ensemble distillation improves calibration for machine translation and named entitity recognition. For our distillation baselines, we follow the recipe by Chen et al. (2020b). For co-distillation, we follow the recipe developed by Anil et al. (2018). In our work, we look at the aforementioned approaches and compare them in terms of resource usage, churn reduction, and effectiveness on the task of conversational semantic parsing (Gupta et al., 2018; Cheng et al., 2020; Damonte et al., 2019; Aghajanyan et al., 2020; Lialin et al., 2020).

3 Task Definition and Evaluation

We follow recent work (Rongali et al., 2020) and treat conversational semantic parsing as sequence generation using auto-regressive neural models. The goal is to make a structured prediction given a user command such as the example in Table 1. For structured prediction, the task of *churn reduction* is, given an input, to predict the exact same sequence across multiple re-training runs. A re-training *run* refers to the model parameters that result from different random weight initialization and data order but the same data and hyper-parameters.

Our aim is to reduce churn across runs while maintaining high accuracy on the gold labels. Thus, we report **exact match accuracy** (EM) with the mean over N runs. While our goal is not to obtain the state of the art, we do want to show which methods reduce churn without a loss in performance.

[1]Datasets can be found at https://github.com/google/stable-retraining-conversational-agents

To measure churn, we need a way to compare predictions across runs, independent of the gold labels. While previous work (Shamir et al., 2020) has used metrics such as prediction difference (similar to Hamming distance), the focus was on classification tasks only, making it necessary to compute an alternative measure. Metrics such as edit distance or multiple sequence alignment would be appropriate for sequence generation tasks such as machine translation or paraphrasing, where churn across output may differ locally by only a few tokens. Comparatively, the meaning of these metrics is unclear for structured prediction tasks such as semantic parsing. For example, computing a token-level distance between a prediction such as "[in:unsupported]" and "[in:get_event [sl:date_time this weekend]]" would not be a useful measure. Thus, we report sequence-level model **agreement** (AGR) across N runs, where each example has a score of 1 if all N runs agree on the exact same predicted sequence and 0 otherwise. However, it is possible for all runs to agree but make an incorrect prediction; the goal ultimately is to consistently make correct predictions. Consequently, we further extend this metric to include the case where the predictions from all N runs agree *and* the predictions match the target. We refer to this metric as **exact match agreement** (EM@N).

4 Methods for Churn Reduction

For our experiments, we explore three techniques which have been effective on related problems such as model calibration: **ensembling**, which combines the predictions of multiple models, **distillation**, which pre-trains a *teacher* model and uses its predictions to train a *student*, and **co-distillation**, which trains two or more *peer* models in parallel and allows each model to learn from the predictions of the other. Figure 1 displays these techniques.

4.1 Ensembling

We create ensembles by uniformly averaging the probabilities of each model to obtain a point estimate. As our semantic parser is an auto-regressive sequence-to-sequence model, at every timestep we create the ensemble distribution over the vocabulary from a mixture of K distributions, as in Reich et al. (2020):

$$p(y_t|y_0 \ldots y_{t-1}, X) = \frac{1}{K} \sum_{k=1}^{K} p_k(y_t|y_0 \ldots y_{t-1}, X) \quad (1)$$

During inference, the next token at each timestep is determined as usual by taking the $argmax$ (in the case of a greedy decoding approach) or using an algorithm such as beam search.

4.2 Distillation

As ensembling increases model size, distillation (Hinton et al., 2015) was introduced to compress the knowledge of an ensemble into a single model. With distillation, a *teacher* model[2] provides a fixed distribution used to train a *student*. The distillation loss from the teacher can be combined with a loss over the target distribution given by gold labels:

$$\mathcal{L}_{student} = \mathcal{L}_{NLL}(\theta, \mathcal{D}) + \lambda * \mathcal{L}_{KD}(p_\theta, q, \mathcal{D}) \quad (2)$$

where $\mathcal{D}$ is the training dataset, $\mathcal{L}_{NLL}$ is negative log-likelihood loss, and $\mathcal{L}_{KD}$ is knowledge distillation loss. While $\mathcal{L}_{KD}$ may be any dissimilarity measure, we use cross-entropy loss between teacher probabilities q and student probabilities p_θ.

For a sequence generation task, computing the exact probabilities $q(Y|X)$ and $p(Y|X)$ for a given X is intractable as it would require a computation over the space of all possible Y. One way to address this problem is with *sequence-level* distillation (Kim and Rush, 2016), which approximates these probabilities with M samples. However, in practice, increasing training time by a factor of M is often infeasible. Instead, we perform *token-level* distillation, computing token probabilities q_i and p_i at each timestep.

The teacher probability q_i of a token i is computed using the "softmax" of its logit z_i,[3] adjusted by a *temperature* T:

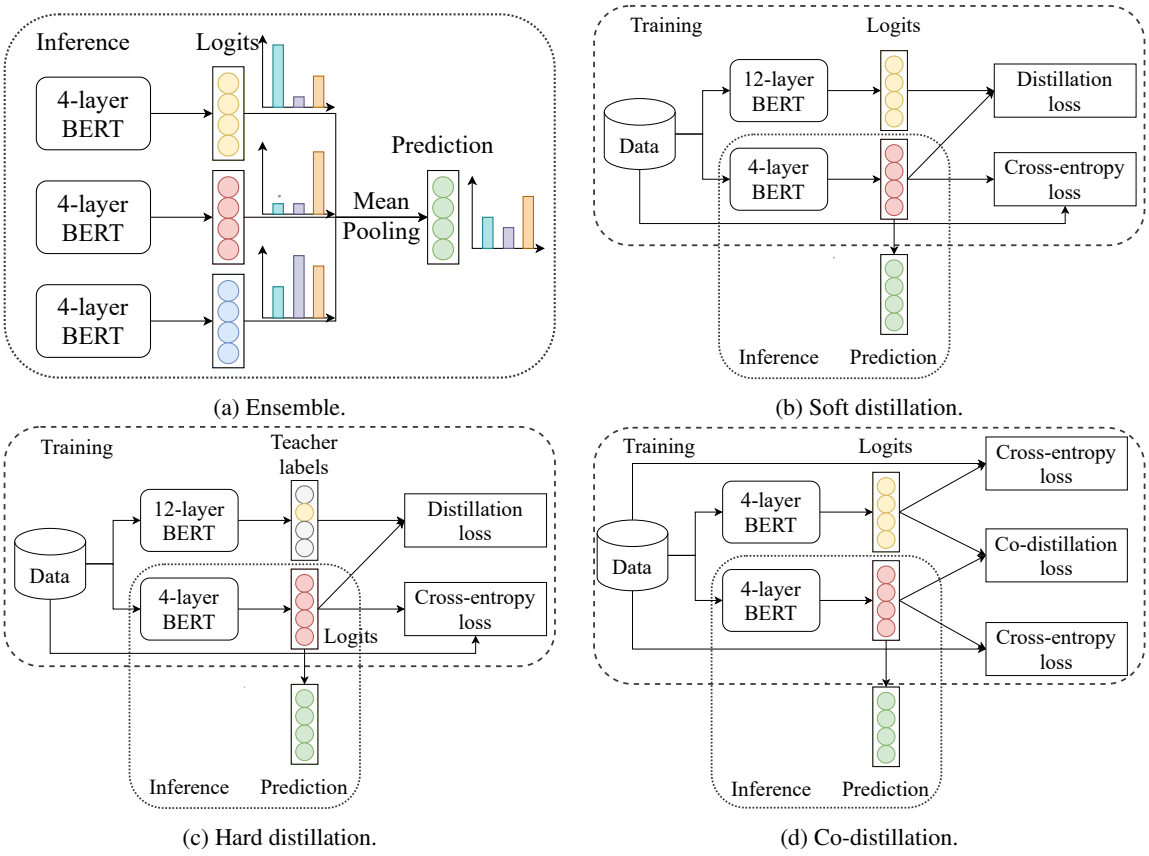
$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)} \quad (3)$$

While T usually is set to 1, the temperature can be used to control the entropy of the distribution, where a high temperature increases uniformity. As the temperature approaches 0, the probability mass is increasingly concentrated on a single token, eventually becoming equivalent to the $argmax$ (a technique known as **hard distillation**). Otherwise, the method is referred to as **soft distillation**.

One challenge for distillation is computing the sequence of targets prior to time t. One possibility is to perform inference with a method such as beam

[2] which is not required to be an ensemble

[3] When distilling from an ensemble, we average the probabilities as in Equation 1 and convert them back to logits.



(a) Ensemble. (b) Soft distillation.

(c) Hard distillation. (d) Co-distillation.

Figure 1: Overview of churn-reducing methods. Dashed and dotted lines indicate the training and inference stages. Rounded rectangular boxes represent seq2seq models with 4- or 12-layer BERT encoders. Ensembling and distillation techniques are applied to the decoder.

search to obtain model predictions. Alternatively, we can use teacher-forcing (Williams and Zipser, 1989; Reich et al., 2020) and condition on true targets through time $t-1$. For soft distillation, using model predictions would require expensive pre-computation and storage of logits or slower training by performing inference at every timestep. However, for hard distillation, only teacher labels are required, making it possible to pre-compute teacher predictions in a single training set pass.

4.3 Co-Distillation

In contrast to distillation, which requires sequential training of the teacher and student, Anil et al. (2018) introduced **co-distillation**, which involves training multiple *peer* models in parallel. While distillation as an abstract idea only requires logits as a signal, and thus the teacher may be a different architecture or even a different dataset, co-distillation has a few distinct features. First, the peer models share an architecture and training data so that the models can be trained online in parallel. Second, the distillation loss is used before the models have converged. Co-distillation loss is computed as:

$$\mathcal{L}_{peers} = \sum_{k=1}^{K} \mathcal{L}_{NLL}(\theta_k, \mathcal{D}) + \sum_{j \neq k} \lambda * \mathcal{L}_{KD}(p_{\theta_k}, q_j, \mathcal{D}) \quad (4)$$

where each of K models is trained with negative log likelihood loss ($\mathcal{L}_{NLL}$) on training data as well as distillation loss ($\mathcal{L}_{KD}$) on the predictions of all other models.

The main advantage of co-distillation is that inference time is equivalent to a single model as only one of the peers is needed. Training time and memory usage are implementation and resource dependent; however the worst case is a K-times increase and may be reduced by, e.g. model parallelism or asynchronous updates (Anil et al., 2018).

5 Experiment Setup

5.1 Datasets

We showcase the problem of model churn on 4 conversational semantic parsing datasets. The TOP

Dataset	Train	Test
TOP	31,279	9,042
TOPv2	124,597	38,785
MTOP	15,667	4,386
SNIPS	13,784	700

Table 2: Data statistics (# of utterances).

dataset (Gupta et al., 2018) consists of queries with hierarchical semantic parses in 2 domains. The TOPv2 (Chen et al., 2020a) and MTOP (Li et al., 2021) datasets expand to 6 more domains with both linear and nested intents and 5 more languages, respectively.[4] Table 1 gives an example of the data format shared across all 3 datasets. We further evaluate on SNIPS (Coucke et al., 2018), another popular semantic parsing dataset with utterances from 7 domains (including AddToPlaylist, BookRestaurant, GetWeather, and PlayMusic). Data statistics are shown in Table 2.

5.2 Noise Injection

We hypothesize that distillation combined with noise reduces churn without a loss in performance. On the one hand, adding noise is a common approach to improving model stability and robustness (Szegedy et al., 2016; Müller et al., 2019). On the other hand, real-world environments often unintentionally contain noise (due to labels collected from multiple sources, e.g., annotators, users, or distant supervision) and models should be resilient to unexpected changes. We explore both scenarios, reporting the results of experiments for **label smoothing** (Szegedy et al., 2016) for the former and **random and systematic noise** for the latter.

Label Smoothing Label smoothing is a widely-used technique for calibration of deep learning models, especially for distillation (Müller et al., 2019). Label smoothing can also be thought of as a noise injection method. This technique is applied by using a weighted average of the one-hot label at a specific timestep and a uniform distribution over all labels. Specifically, at time step t, we compute a new "soft" target:

$$(1-\alpha)\delta_{t,l} + \alpha\frac{1}{|L|} \quad (5)$$

where $\delta_{t,l}$ is the one-hot label if present, α is a parameter that controls the percentage of smoothing, and L is the set of all labels. We follow the recommendations of (Müller et al., 2019) in applying label smoothing only to student models. We set $\alpha = 0.1$ to match the random/systematic noise settings and hold constant the amount of noise across all experiments.

Random Noise To simulate noise that may occur in a real-world scenario, we create an artificial **random noise** dataset by randomly swapping 10% of labels from a weighted distribution. To construct this dataset, we first find all labels with the prefix "[in:" (intents) and compute their probabilities in training. Then, we randomly sample a replacement intent from this distribution. We repeat this process for slots ("[sl:").

Systematic Noise High-quality labeled data for SLU systems may be difficult to come by in large quantities. Conversational agents are therefore often trained using "distant-labeled" data from an earlier iteration. This process inevitably results in noisy data, as no SLU system will obtain 100% on all unseen examples. To simulate this distant supervision, we construct a **systematic noise** dataset. We train a baseline with a 4-layer BERT encoder (see Section 5.3) on 90% of each training set and label the remaining 10%. However, in order to obtain labels that are both (a) systematic and (b) incorrect, we select the prediction at the *second* beam position rather than the first.[5]

5.3 Implementation details

Baselines The pointer generator network of Rongali et al. (2020) obtained competitive performance on the TOP datasets using pre-trained encoders. We obtain similar results upon re-implementing this work as a baseline. As our goal is to reduce churn in a realistic environment, we use a "production-sized" encoder – the 4-layer BERT model of Turc et al. (2019) with 4 heads and 256 dimensions – to reflect what can reasonably be served to users at a robust query-per-second rate. We selected this model to evaluate distillation from a larger model of the same type, 12-layer BERT-base (Devlin et al., 2019), which differs only by the number of parameters. The 4-layer BERT was distilled from BERT-base and obtained only a small decrease on benchmark datasets compared to larger models.

[4] Although our work is limited to English only.

[5] In practice, this results in less than 10% of the training data being incorrect. However, on all datasets used in these experiments, the percentage of correct predictions at the second beam position is less than 5%, thus ensuring that at least 9.5% of the training data is noisy.

	TOP		TOPv2		MTOP		SNIPS	
Model	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR
BERT-4	80.65 (70.29)	75.48	83.88 (73.12)	78.15	79.31 (69.04)	73.64	86.90 (77.12)	80.29
Ensemble	84.60 (78.55)	86.18	86.42 (80.38)	88.17	84.59 (78.52)	84.39	87.69 (80.58)	84.60
SD (ensemble)	81.20 (70.80)	76.16	84.00 (73.47)	78.75	79.29 (67.40)	71.38	87.29 (79.71)	83.45
SD (BERT-12)	80.93 (71.14)	76.80	84.12 (73.87)	79.02	79.23 (68.71)	73.23	87.34 (78.27)	80.86
HD (BERT-12)	80.72 (70.01)	75.03	83.84 (72.57)	77.37	78.96 (68.61)	73.07	87.44 **(80.86)**	**84.75**
Co-distillation	**81.43 (73.56)**	**80.41**	**84.21 (76.10)**	**82.99**	**79.45 (69.73)**	**74.87**	**87.50 (80.86)**	**84.75**

(a) Original dataset (label smoothing with $\alpha = 0.1$).

	TOP		TOPv2		MTOP		SNIPS	
Model	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR
BERT-4	77.02 (65.81)	71.58	82.60 (71.03)	75.96	68.12 (45.88)	49.12	78.41 (57.12)	58.85
Ensemble	78.67 (72.21)	80.55	83.78 (76.53)	83.89	72.37 (58.78)	65.24	82.27 (67.23)	70.50
SD (ensemble)	79.44 (68.53)	73.78	83.22 (72.40)	77.71	67.75 (44.51)	47.23	77.89 (56.69)	58.99
SD (BERT-12)	77.11 (65.47)	71.51	82.73 (70.25)	74.65	66.67 (41.00)	43.62	78.11 (56.69)	58.85
HD (BERT-12)	77.33 (59.83)	63.14	82.40 (68.85)	72.76	67.99 (42.84)	44.51	77.89 (56.69)	58.99
Co-distillation	**80.21 (72.04)**	**78.86**	**83.18 (73.09)**	**78.85**	**73.50 (58.43)**	**62.22**	**82.00 (66.33)**	**68.92**

(b) 10% random noise.

	TOP		TOPv2		MTOP		SNIPS	
Model	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR
BERT-4	78.15 (61.36)	65.11	81.80 (67.20)	70.86	74.72 (57.09)	60.81	81.17 (58.42)	60.43
Ensemble	79.87 (68.78)	74.52	83.40 (73.60)	79.75	77.59 (68.55)	75.80	84.50 (71.22)	74.53
SD (ensemble)	79.85 (67.46)	72.36	**83.04 (71.50)**	**76.60**	74.84 (57.91)	**61.99**	81.96 (60.72)	63.02
SD (BERT-12)	79.28 (66.83)	71.70	81.84 (67.47)	71.10	74.97 (57.16)	61.01	81.67 (59.71)	62.45
HD (BERT-12)	79.12 (65.93)	70.37	81.36 (65.33)	68.47	74.51 (56.72)	60.37	80.23 (56.26)	58.71
Co-distillation	**80.83 (72.14)**	**78.45**	81.97 (70.12)	75.91	**75.03 (58.16)**	61.49	**83.66 (68.78)**	**72.23**

(c) 10% systematic noise.

Table 3: Model performance (over $N = 10$ runs) when trained on datasets with varying degrees of noise. All student models use 4-layer BERT. BERT-4/12: 4/12-layer BERT. Ensemble: 4-layer ensemble. SD: soft distillation. HD: hard distillation. EM: exact match (mean over 10 runs). EM@10: EM if all 10 models are correct. AGR: model agreement. **Bold**: best non-ensemble.

Experiments For our experiments, we explore different settings for ensembling and distillation. For both our **ensemble** and **ensemble distillation**, we use 4-layer BERT models with $K = 3$. We use soft distillation and obtain teacher probabilities with teacher forcing and Equation 1. While distilling from an ensemble may increase agreement by preventing the student from assigning too much probability to a single token and becoming overconfident, we also explore **soft distillation** from a 12-layer teacher. We hypothesize that the 12-layer model would have higher EM but lower AGR than the 4-layer ensemble and this setup allows us to explore any tradeoff between these measurements. In addition, we consider **hard distillation** from a 12-layer model. For this setting, we use beam search inference with a beam width of 3 to obtain predictions, so that we can compare to teacher forcing for soft distillation. We perform offline inference with the 12-layer model on the entire training set and use both the teacher-labeled data and the gold data for every example. Finally, we use **co-distillation** with $K = 2^6$ and $\lambda = 1$. We distill from model predictions using weights updated at every timestep.

Hyperparameters To reduce non-determinism, we use a single set of hyper-parameters for the 3 TOP datasets and all experiments. For SNIPS, we select a single set of hyper-parameters by tuning the baseline on 10% of the training data. Appendix B lists all hyper-parameters.

6 Results

We test the effectiveness of the methods described in Section 4 over $N = 10$ runs. We compile results in Table 3a for models trained on the original datasets with label smoothing. We also report re-

[6] as recommended by Anil et al. (2018).

sults for the 10% random/systematic noise setting (Tables 3b and 3c) as we assume this represents a "real-world" scenario where labels are 90% correct.

Ensemble superior at the cost of much increased computational cost First, ensemble sets a high bar in almost all settings regardless of artificial noise. While impressive, this approach requires significantly more computation at inference time and is sometimes deemed infeasible to deploy when accounting for resource usage (see Table 8).

Co-distillation best among distillation-based methods regardless of noise For label smoothing (Table 3a) and the random/systematic noise settings (Tables 3b and 3c), co-distillation clearly and consistently outperforms the baseline in EM, EM@10, and AGR. We also find that soft distillation from the ensemble occasionally obtains the best performance (TOPv2 with systematic noise) but more frequently performs worse than the baseline (MTOP/SNIPS with random noise). On the other hand, soft/hard distillation perform merely on-par with the baseline or worse. Surprisingly, in the 10% random/systematic noise setting, co-distillation not only narrows the gap for EM@10/AGR compared to the ensemble, but also occasionally outperforms the ensemble in EM for TOP/MTOP and TOP, respectively, which may be due to increased robustness to noise during training, rather than only during inference in the ensemble.

6.1 Effect of Task Difficulty

Table 4 shows the performance of the baseline models as we increase the task difficulty by reducing the model size or increasing noise in the data. As expected, EM decreases as the task becomes more difficult. However, AGR decreases more rapidly because with lower EM the model has more degrees of freedom to find solutions. These results also show that EM alone is not enough to measure reproducibility and validate the use of EM@10/AGR.

6.2 Effect of Label Smoothing

To better understand the effect of label smoothing, we conduct a study of TOPv2 for the baseline and co-distillation models (Table 5)[7]. On the base dataset in the baseline setting (BERT-4), label smoothing provides little to no benefit in all metrics. However, we observe a dramatic improvement for co-distillation with label smoothing vs without

[7] see Appendix D for the full results

Model and Setting	EM(@10)	AGR
BERT-12 (0% random noise)	**85.68** (76.11)	**81.30**
BERT-4 (0% random noise)	83.74 (73.18)	78.15
BERT-4 (10% random noise)	82.60 (71.03)	75.96
BERT-4 (25% random noise)	81.34 (69.04)	73.73
BERT-4 (50% random noise)	76.83 (62.87)	67.28

Table 4: Effect of Task Difficulty on TOPv2, varying baseline model size (4/12-layer BERT) and random noise. EM(@10): exact match (with all 10 runs correct). AGR: model agreement. **Bold**: best performance.

Model and Setting	EM(@10)	AGR
BERT-4 ($\alpha = 0$)	83.74 (73.18)	78.47
BERT-4 ($\alpha = 0.1$)	83.89 (73.12)	78.15
CD ($\alpha = 0$)	84.01 (73.96)	79.49
CD ($\alpha = 0.1$)	**84.21 (76.10)**	**82.99**
BERT-4 ($\alpha = 0$, 10% rand.)	82.60 (71.03)	75.96
BERT-4 ($\alpha = 0.1$, 10% rand.)	82.38 (71.11)	76.24
CD ($\alpha = 0$, 10% rand.)	**83.18 (73.09)**	78.85
CD ($\alpha = 0.1$, 10% rand.)	82.60 (73.06)	**79.33**
BERT-4 ($\alpha = 0$, 10% sys.)	81.80 (67.20)	70.86
BERT-4 ($\alpha = 0.1$, 10% sys.)	83.02 (72.27)	77.74
CD ($\alpha = 0$, 10% sys.)	81.97 (70.12)	75.91
CD ($\alpha = 0.1$, 10% sys.)	**83.19 (73.96)**	**80.50**

Table 5: Effects of Label Smoothing on TOPv2. BERT-4: baseline. CD: co-distillation. α: label smoothing wt. EM(@10): exact match (with all 10 runs correct) AGR: model agreement **Bold**: best performance.

in EM@10 (+2.14) and AGR (+3.5). On the other hand, on the dataset with 10% random noise, we do not observe any benefit with label smoothing for either the baseline or co-distillation, perhaps due to the noise already in the data. Finally, on the dataset with 10% systematic noise, we observe that label smoothing dramatically improves results for both the baseline - EM@10 (+5.07) and AGR (+6.88) - and co-distillation - EM@10 (+2.84) and AGR (+4.59). Overall, in the most realistic scenarios ("clean" or distant-labeled data), we find that co-distillation can be effectively combined with label smoothing. This result is in contrast to Müller et al. (2019), who found that training a teacher with label smoothing is not effective. When both models are teachers, it is clear that label smoothing helps.

7 Discussion

Qualitative Analysis To further understand what queries cause the model to churn, we analyze cases where multiple runs disagree. To keep the analysis simple we compare the baseline with co-distillation in Table 6 (additional examples in Appendix C).

Query	play new matchbox 20
Model Run 1	[in:play_music [sl:music_artist_name matchbox 20]]
Model Run 2	[in:play_music [sl:music_track_title matchbox 20]]
Query	**repeat closer**
Model Run 1	[in:replay_music [sl:music_track_title closer]]
Model Run 2	[in:loop_music]

Table 6: Churn examples from TOPv2 fixed by co-distillation. Model predictions are from the baseline. In both cases, only Model Run 1 matches the target, but Model Run 2 has an incorrect intent or slot.

The first row shows that the baseline model runs are confused by semantically similar slots – *music_artist_name* vs. *music_track_title*. The second row demonstrates baseline confusion between the intents *loop_music* vs. *replay_music*. In both cases the co-distilled models agree across all training runs. Due to the semantic similarity of the slots/intents, we can attribute this churn to under-specification (D'Amour et al., 2020), which is reduced by co-distillation.

We also explore the relation between agreement and the length of the structured output sequences. Figure 2 plots the number of models in agreement against the number of intents and slots. In making a structured prediction during inference, as length increases the model has more freedom to select incorrect tokens and therefore churn increases. Co-distillation increases agreement for longer sequences, but ensembling is especially robust. Table 7 reports the average target and prediction length where all N models disagree. Surprisingly, we observe that the models over-generate compared to the target; however, the difference is reduced with co-distillation/ensembling.

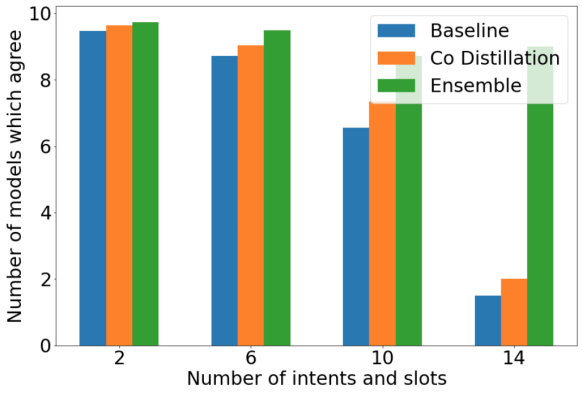


Figure 2: Agreement across trained models for various methods vs prediction complexity.

Method	Target	Prediction
Baseline	3.66	3.91
Co-distillation	3.77	3.82
4 layer ensemble	3.56	3.70

Table 7: Average # of slots and intents for cases where all N models disagree. When there is churn the model over-generates (i.e. prediction length > target length).

Practical considerations We roughly compare the methods along the resource usage dimension in Table 8. As resource usage may be implementation or architecture dependent, we report the number of parameters, which correlates strongly with training/inference time and memory. While ensembling is the strongest approach, it also comes with the most expensive inference. Although wall-clock inference time may be the same as the base model due to parallelization, computing power and memory scales by a factor of K. Further, while distillation methods have the same inference time due to similar sized outputs, they have different costs w.r.t. training the teacher.[8] For ensemble distillation, the teacher models can be trained in parallel, but still have Kx storage requirements. For large-model distillation, in practice our 12-layer teacher has about $P = 9$ times the number of parameters as the baseline. In both cases, the student *must be trained sequentially*. Overall, co-distillation performs consistently well across different datasets and noise settings in terms of EM and model agreement while striking a balance between computational cost and performance, rendering it an attractive approach for goal-oriented conversational semantic parsing.

Method	Training (actual)	Inference (actual)
Baseline	x	x
Ensemble	$P_e^* = 3x$	$P_e^* = 3x$
Ens. distillation	$P_e^* + x = 4x$	x
Large distillation	$P_l + x = 10x$	x
Co-distillation	$P_c^* = 2x$	x

Table 8: Overview of resource usage by number of parameters (relative to 4-layer baseline with x =~14 million parameters). $P_{e/l/c}$: Number of ensemble/teacher/peer parameters. * denotes parallelism.

8 Conclusion

Our experiments showed that there exists substantial churn across runs when re-training models on the same conversational semantic parsing datasets. We showed that for "production-sized" models, co-

[8] Hard/soft distillation have equal number of parameters.

distillation with label smoothing increases agreement without loss of accuracy. Furthermore, on noisy data simulating a real-world environment, the improvement is even more drastic. When we account for resource usage along with accuracy, we provide strong evidence that co-distillation provides the sweet spot compared to methods like hard/soft distillation and ensembling.

In future work, we plan to explore how other modeling decisions can increase or decrease model churn. In this work, we limited our focus to BERT encoders with different number of layers. Other questions to explore include whether the choice of pre-training technique affects churn or whether pre-trained encoder-decoders show the same effects. Finally, we will examine whether alternative decoding algorithms, such as non-autoregressive approaches (Babu et al., 2021; Oh et al., 2022), can reduce churn.

References

Armen Aghajanyan, Jean Maillard, Akshat Shrivastava, Keith Diedrick, Michael Haeger, Haoran Li, Yashar Mehdad, Veselin Stoyanov, Anuj Kumar, Mike Lewis, and Sonal Gupta. 2020. Conversational semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5026–5035, Online. Association for Computational Linguistics.

Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E. Dahl, and Geoffrey E. Hinton. 2018. Large scale distributed neural network training through online distillation. In *International Conference on Learning Representations*.

Arun Babu, Akshat Shrivastava, Armen Aghajanyan, Ahmed Aly, Angela Fan, and Marjan Ghazvininejad. 2021. Non-autoregressive semantic parsing for compositional task-oriented dialog. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2969–2978, Online. Association for Computational Linguistics.

Dara Bahri and Heinrich Jiang. 2021. Locally adaptive label smoothing improves predictive churn. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 532–542. PMLR.

Xilun Chen, Asish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020a. Low-resource domain adaptation for compositional task-oriented semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5090–5100, Online. Association for Computational Linguistics.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2020b. Distilling knowledge learned in BERT for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905, Online. Association for Computational Linguistics.

Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, Jason D. Williams, Hong Yu, Diarmuid Ó Séaghdha, and Anders Johannsen. 2020. Conversational semantic parsing for dialog state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8107–8117, Online. Association for Computational Linguistics.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 16–23, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexander D'Amour, Katherine A. Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, et al. 2020. Underspecification presents challenges for credibility in modern machine learning. *CoRR*, abs/2011.03395.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thomas G. Dietterich. 2000. Ensemble methods in machine learning. In *MULTIPLE CLASSIFIER SYSTEMS, LBCS-1857*, pages 1–15. Springer.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Elliot Karro, and D. Sculley, editors. 2017. *Google Vizier: A Service for Black-Box Optimization*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *ICML*, pages 1321–1330.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2787–2792, Brussels, Belgium. Association for Computational Linguistics.

L. K. Hansen and P. Salamon. 1990. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10):993–1001.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Heinrich Jiang, Harikrishna Narasimhan, Dara Bahri, Andrew Cotter, and Afshin Rostamizadeh. 2022. Churn reduction via distillation. In *International Conference on Learning Representations*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

Vladislav Lialin, Rahul Goel, Andrey Simanovsky, Anna Rumshisky, and Rushin Shah. 2020. Continual learning for neural semantic parsing. *CoRR*, abs/2010.07865.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Mahdi Milani Fard, Quentin Cormier, Kevin Canini, and Maya Gupta. 2016. Launch and iterate: Reducing prediction churn. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9131–9143, Online. Association for Computational Linguistics.

Geunseob Oh, Rahul Goel, Christopher Hidey, Shachi Paul, Aditya Gupta, Pararth Shah, and Rushin Shah. 2022. Improving top-k decoding for non-autoregressive semantic parsing via intent conditioning. *CoRR*, abs/2204.06748.

Steven Reich, David Mueller, and Nicholas Andrews. 2020. Ensemble Distillation for Structured Prediction: Calibrated, Accurate, Fast—Choose Three. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5583–5595, Online. Association for Computational Linguistics.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. *Don't Parse, Generate! A Sequence to Sequence Architecture for Task-Oriented Semantic Parsing*, page 2962–2968. Association for Computing Machinery, New York, NY, USA.

Gil I. Shamir and Lorenzo Coviello. 2020. Anti-distillation: Improving reproducibility of deep networks. *CoRR*, abs/2010.09923.

Gil I. Shamir, Dong Lin, and Lorenzo Coviello. 2020. Smooth activations and reproducibility in deep networks. *CoRR*, abs/2010.09931.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *CoRR*, abs/1908.08962.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ronald J. Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation*, 1(2):270–280.

A Ethics

The TOP and SNIPS datasets used in this experiments are intended for research purposes only. We verified that the datasets do not contain personally identifiable information. The risks of dual use for task-oriented conversational semantic parsers are low as we are not performing open-ended generation; however, the models are likely to overfit to certain demographic groups and underperform on others.

B Hyper-parameter Search and Settings

We run our experiments on the TPU v2 available through Google Cloud.[9]

We use the same hyper-parameters for all 3 TOP datasets and SNIPS, except for SNIPS we use a different number of training steps and learning rate. The hyper-parameters were selected using the Google Cloud black box optimizer (Golovin et al., 2017). We tuned the parameters using 64 re-runs over the settings described in Table 9. For SNIPS, we held out 10% of the training data for tuning the training steps (100000) and learning rate (0.000031) and trained the final models on 100% of the training data with the selected hyper-parameters. For distillation experiments we adjusted the learning rate to $1e-5$ and the batch size to 128 to prevent overfitting.

We train all models (including teacher and student) for 300000 steps on the TOP datasets and 100000 on SNIPS. We use the Adam optimizer with weight decay (Loshchilov and Hutter, 2017) and the relu activation function. To follow the pointer generator approach of Rongali et al. (2020), we embed the output vocabulary in 128-dimensional vectors and project the BERT embeddings from the input to 128 dimensions as well. For our transformer decoder (Vaswani et al., 2017), we use 2 heads and 2 layers (see Table 9) with 256 dimensions for the attention and feed forward layers. We also use a maximum output length of 51. We use dropout on the input wordpiece embeddings, after the contextual BERT embeddings, and on the output embeddings before the softmax layer.

Hyper-parameter	Range/Set	Selected Value
Learning rate	$[2e-5, 2e-4]$	4e-5
Decoder Heads	$\{2, 4, 8\}$	2
Decoder Layers	$\{2, 4, 8\}$	4
Batch Size	$\{128, 256\}$	256
Dropout	$[0.01, 0.1]$	0.0316

Table 9: Tuned Hyper-parameters and their Possible Values

C Additional Examples

Table 10 provides additional examples where ensembling fixes errors still present in co-distilled models. In these cases, the co-distilled models over-generate (the phenomenon indicated in Table 7) whereas the lengths of the ensemble predictions are correctly calibrated to the target lengths.

D Additional Results

We present the full set of results from Table 5 in Table 11. The results in Table 11a provide strong evidence that co-distillation with label smoothing (Table 11b) is clearly preferable. When we examine the full set of datasets and methods combined with label smoothing in the random/systematic noise setting, we also see that soft distillation from an ensemble performs well. However, in some cases soft ensemble distillation performs worse than the baseline; swapping occasionally slightly better performance for occasionally much worse performance would not be an acceptable tradeoff in most cases. Co-distillation is more stable in terms of consistently outperforming the baseline. Furthermore, co-distillation requires fewer resources and can be trained in parallel.

[9] https://cloud.google.com/tpu

Query	Ground Truth	Model predictions
play new matchbox 20	[in:play_music [sl:music_artist_name matchbox 20]]	[in:play_music [sl:music_track_title matchbox 20]] [in:play_music [sl:music_artist_name matchbox 20]]
repeat closer	[in:replay_music [sl:music_track_title closer]]	[in:replay_music [sl:music_track_title closer]] [in:loop_music]
Churn examples fixed by co-distillation. Model predictions are from the baseline model		
show me alarms for tomorrow	[in:get_alarm [sl:date_time for tomorrow]]	[in:get_alarm [sl:alarm_name [in:get_time [sl:date_time for tomorrow]]]] [in:get_alarm [sl:date_time for tomorrow]]
take out my wednesday alarm.	[in:delete_alarm [sl:alarm_name [in:get_time [sl:date_time wednesday]]]]	[in:delete_alarm [sl:alarm_name [in:get_time [sl:date_time wednesday]]]] [in:silence_alarm [sl:alarm_name [in:get_time [sl:date_time wednesday]]]]
Churn examples further fixed by ensembling. Model predictions from the co-distilled model		

Table 10: Qualitative comparison on TOPv2 of the types of errors fixed by co-distillation and ensembling.

	TOP		TOPv2		MTOP		SNIPS	
Model	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR
BERT-4	**81.51 (72.14)**	77.85	83.74 (73.18)	78.47	**80.13 (68.71)**	72.54	86.83 (75.25)	78.42
Ensemble	84.60 (78.55)	86.18	86.42 (80.38)	88.17	84.59 (78.52)	84.39	87.69 (80.58)	84.60
SD (ensemble)	81.36 (71.63)	77.25	83.73 (72.62)	77.72	79.50 (68.11)	71.97	86.80 (75.11)	77.84
SD (BERT-12)	81.31 (71.16)	76.43	83.51 (72.13)	77.10	79.87 (67.36)	70.76	86.37 (73.96)	76.69
HD (BERT-12)	81.33 (70.91)	75.92	83.56 (72.15)	76.99	79.66 (67.09)	70.39	86.93 (77.12)	80.29
Co-distillation	81.31 (72.04)	**77.98**	**84.01 (73.96)**	**79.49**	79.55 (68.48)	**72.95**	**87.39 (79.28)**	**82.59**

(a) Original dataset (no noise)

	TOP		TOPv2		MTOP		SNIPS	
Model	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR
BERT-4	77.90 (64.28)	68.76	82.39 (71.11)	76.24	70.01 (44.97)	46.63	76.59 (51.08)	52.95
Ensemble	78.67 (72.21)	80.55	83.78 (76.53)	83.89	72.37 (58.78)	65.24	82.27 (67.23)	70.50
SD (ensemble)	**80.14 (70.59)**	**76.45**	**83.51 (74.31)**	**80.46**	71.14 (48.89)	51.27	80.96 (61.01)	63.17
SD (BERT-12)	78.71 (66.96)	72.05	82.71 (70.75)	75.50	69.83 (45.26)	47.09	78.59 (53.09)	55.54
HD (BERT-12)	77.71 (64.77)	69.54	81.11 (60.39)	63.08	69.63 (44.78)	46.52	76.70 (47.34)	48.92
Co-distillation	78.91 (68.42)	74.54	82.60 (73.07)	79.34	**73.74 (57.64)**	**61.22**	**82.50 (68.35)**	**71.80**

(b) 10% random noise and label smoothing with $\alpha = 0.1$.

	TOP		TOPv2		MTOP		SNIPS	
Model	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR	EM (@10)	AGR
BERT-4	79.66 (65.28)	70.17	83.03 (72.27)	77.74	74.58 (58.50)	62.86	84.19 (68.20)	70.94
Ensemble	79.87 (68.78)	74.52	83.40 (73.60)	79.75	77.59 (68.55)	75.80	84.50 (71.22)	74.53
SD (ensemble)	**81.02** (71.71)	77.87	**83.85 (74.46)**	**80.68**	74.97 (58.87)	63.30	82.24 (57.12)	59.14
SD (BERT-12)	80.75 (71.22)	77.15	83.25 (73.19)	78.97	75.01 (59.28)	63.30	82.59 (63.02)	65.90
HD (BERT-12)	79.49 (64.51)	69.10	82.93 (72.27)	77.94	75.21 (57.11)	60.51	81.57 (59.71)	62.88
Co-distillation	80.84 **(73.61)**	**81.27**	83.19 (73.96)	80.50	**76.98 (63.64)**	**68.09**	**85.49 (72.09)**	**76.26**

(c) 10% systematic noise and label smoothing with $\alpha = 0.1$.

Table 11: Model performance (over $N = 10$ runs) when trained on datasets with varying degrees of noise. All student models use 4-layer BERT. BERT-4/12: 4/12-layer BERT. Ensemble: 4-layer ensemble. SD: soft distillation. HD: hard distillation. EM: exact match (mean over 10 runs). EM@10: EM if all 10 models are correct. AGR: model agreement. **Bold**: best non-ensemble.

Knowledge-Grounded Conversational Data Augmentation with Generative Conversational Networks

Yen-Ting Lin ytl@ieee.org
Alexandros Papangelis {papangea,seokhwk,hakkanit}@amazon.com
Seokhwan Kim {papangea,seokhwk,hakkanit}@amazon.com
Dilek Hakkani-Tur {papangea,seokhwk,hakkanit}@amazon.com

Abstract

While rich, open-domain textual data are generally available and may include interesting phenomena (humor, sarcasm, empathy, etc.) most are designed for language processing tasks, and are usually in a non-conversational format. In this work, we take a step towards automatically generating conversational data using Generative Conversational Networks, aiming to benefit from the breadth of available language and knowledge data, and train open domain social conversational agents. We evaluate our approach on conversations with and without knowledge on the Topical Chat dataset using automatic metrics and human evaluators. Our results show that for conversations without knowledge grounding, GCN can generalize from the seed data, producing novel conversations that are less relevant but more engaging and for knowledge-grounded conversations, it can produce more knowledge-focused, fluent, and engaging conversations. Specifically, we show that for open-domain conversations with 10% of seed data, our approach performs close to the baseline that uses 100% of the data, while for knowledge-grounded conversations, it achieves the same using only 1% of the data, on human ratings of engagingness, fluency, and relevance.

1 Introduction

Conversational Artificial Intelligence has progressed a lot in the recent past, partly due to advances in large pre-trained language models (PLM) and partly due to commercial conversational agents (Alexa, Siri, Cortana, Google Assistant, and others). It is evident, however, that many challenges still remain, such as handling idioms, humour, expressing empathy, processing unstructured knowledge, and so on. One big factor for this is the lack of large and rich conversational data that include these complex aspects of human communication. While the research community is making great efforts in collecting such data (e.g. empathetic dialogues (Rashkin et al., 2019), persuasion (Wang et al., 2019), and others), these are still small compared to the amount of data needed to train deep neural networks. Furthermore, these expensive data collections usually target a single phenomenon at a time, and hence do not necessarily scale to the richness of human conversations. Another challenge for real world applications is privacy, preventing the use of much of the publicly available conversational data.

In this work, we take a first step into automatically generating conversational data from unstructured textual knowledge (e.g. web sources) using Generative Conversational Networks (GCN) (Papangelis et al., 2021). GCN is a meta-learning method initially proposed for intent detection and slot tagging; we extend that approach and demonstrate that we can learn how to generate responses grounded in unstructured knowledge. Specifically, GCN learns how to generate labelled, diverse, and targeted data that are optimised with Reinforcement Learning (RL). This is achieved by using a generator model that produces new data which is used to train a separate learner model. The performance of the learner model is used as a reward signal to train the generator, so that over time the quality of the generated data increases. This reward signal can allow us to guide the data generation towards dimensions of interest, for example, knowledge-grounded, empathetic, or polite dialogues and can be derived from automatic metrics or human feedback if the system is deployed. In our case, the generator produces open-domain dialogues and the learner is a conversational agent that is trained on that data. Selecting an appropriate reward signal can be difficult, since we want to generate good quality dialogues that do not exist in the training data, but dialogue evaluation is a challenging open problem. We therefore investigate a combination of multiple metrics that capture different aspects: BLEU (Papineni et al., 2002)

Proceedings of the SIGdial 2022 Conference, pages 26–38
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

and ROUGE (Lin, 2004) to ensure some similarity with the reference data, BERTScore (Zhang et al., 2020a)[1] to encourage good quality dialogues, and Knowledge F1[2] (Shuster et al., 2021) to encourage knowledge integration. It should be noted that while the focus in this work is knowledge grounding in open-domain response generation, our approach is extensible to other conversational phenomena with appropriate reward signals.

Our main contributions are: a) we generate knowledge-grounded conversational data from unstructured textual knowledge (e.g. the kind of knowledge available on the web); b) we improve response generation quality over a baseline that uses fine-tuning on seed data, eliminating the need for additional human-human data collection; and c) we demonstrate improved performance on knowledge-grounded response generation on Topical Chat, as measured by KF1 and human evaluations.

2 Related Work

Language Data Augmentation Approaches. There are a lot of recent works on data augmentation, but most of them are geared towards individual language processing tasks rather than training complete conversational agents. Due to lack of space we only mention the ones that are most relevant to our work.

PROTODA (Kumar et al., 2021) uses prototypical networks to augment data for intent classification while GenSF (Mehri and Eskenazi, 2021) uses DialoGPT (Zhang et al., 2020b) for zero-shot slot tagging; DINO (Schick and Schütze, 2021) uses PLM to generate data for semantic textual similarity; Campagna et al. (2020) focus on zero-shot dialogue state tracking and use an abstract dialogue model to generate data. SOLOIST (Peng et al., 2021) uses a PLM fine-tuned on large dialogue corpora and is designed for transactional (goal-oriented) dialogues. Mohapatra et al. (2020) use PLM to train user simulators from crowd-generated conversations and their instructions. Lin et al. (2021a) train domain-independent user simulators for transactional dialogues. Chang et al. (2021) augment data for Data-To-Text NLG by generating text in two steps: replacing values with alternatives and using GPT-2 to produce surface text. They then do automatic labelling and enforce cycle-consistency (make sure text can be generated from data and vice versa). Stahlberg and Kumar (2021) focus on data generation for Grammatical Error Correction and propose a method that can generate an erroneous sentence given a correct sentence and an error tag. Chen and Yu (2021) use data augmentation to improve out of scope (OOS) detection models. Specifically, they extract utterances from a different dataset than the one they are targeting that can be labelled as OOS and then do some smart filtering to select good candidates. Kim et al. (2021) propose NeuralWOZ, a framework to generate dialogue state tracking data given goal descriptions and API calls. NeuralWOZ has a data generator and a data labeler that annotates the data. GCN does not need a separate labeler model and has the added option of being continually trained with RL. PromDA (Wang et al., 2022b) is a soft-prompt learning method for low-resource NLP tasks, that addresses the problem of overfitting (memorizing) when fine-tuning a PLM with a very small number of examples. The authors generate data for sequence classification and labelling. However, this approach is not tested on full dialogues which require significantly more context in the input. Bayer et al. (2022) propose a three step method, where they first fine-tune a PLM and then generate new data-points by adjusting the temperature of the generation. They then filter the generated data by putting a threshold on embedding similarity with respect to the target class centroid. GCN uses RL to guide the generation process, alleviating the need for explicit post-processing. Wang et al. (2022a) present a data augmentation approach for aspect-based sentiment analysis that can generate data along two dimensions: aspects and polarity. The resulting data are then used in a contrastive learning setting to train a sentiment classifier. Similarly to other approaches, it is not clear how it would perform in knowledge-grounded dialogue generation, with large inputs (context and available knowledge). For a more comprehensive review of data augmentation for language tasks, please see (Feng et al., 2021; Li et al., 2021; Sahin, 2022).

Regarding data augmentation for conversational agents, one of the most prominent methods is User Simulation (Schatzmann et al., 2007; Asri et al., 2016; Liu and Lane, 2018; Papangelis et al., 2019; Lin et al., 2021b; Shah et al., 2018, e.g.). These approaches, however, have been designed to work

[1] Data driven evaluation metrics tend to favor dialogues similar to the ones used during their training and we found that we cannot solely rely on such metrics.

[2] KF1 measures the token level F1 score between a knowledge piece and an utterance.

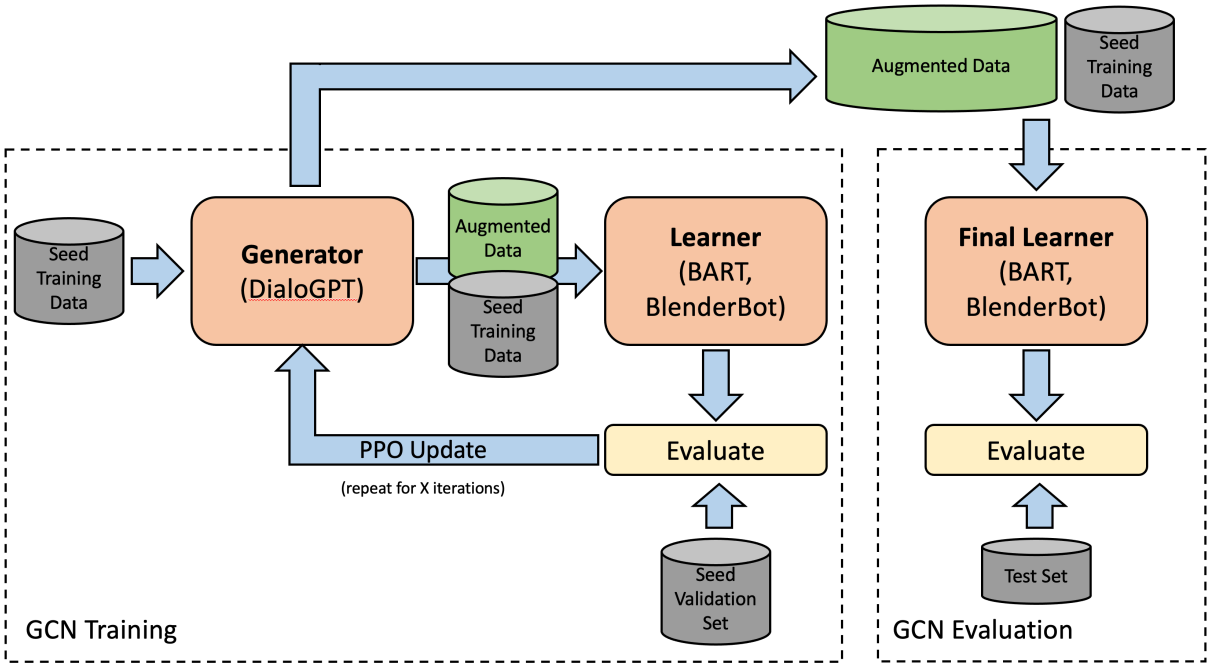


Figure 1: The architecture of our approach using Generative Conversational Networks for knowledge-grounded dialogues. The generator is first fine-tuned with seed data and produces an augmented dataset and those data are used to train a learner. The performance of the learner on a held-out validation set (along with auxiliary metrics) is used as a reward to update the generator.

with well-structured databases whereas we are concerned with grounding open-domain conversational responses in unstructured knowledge. DG^2 (Wu et al., 2021) focuses on data augmentation for document-grounded dialogues, using Doc2Dial (Feng et al., 2020). The authors use an agent bot and a user bot to conduct simulated conversations and generate data. However, unlike GCN, the bots are not continually updated and may not generalise well to produce novel content. The code was not available for a direct comparison on our dataset, however, in the few-shot learning experiments, they demonstrate good performance with as little as 25% of the data (869 Doc2Dial dialogues), whereas we demonstrate competitive performance by only using 1% of the training data (86 Topical Chat dialogues).

Few-Shot Approaches. Another line of related work is based on few-/zero-shot transfer learning for dialogue tasks. Again due to space we only mention the most relevant works. Earlier studies have focused on improving the generalizability of natural language understanding problems such as intent classification (Chen et al., 2016) and slot filling (Bapna et al., 2017; Shah et al., 2019) for unseen labels or domains. Then, focus was placed on other dialogue problems including dialogue state tracking (Wu et al., 2019; Rastogi et al., 2020), next action prediction (Mosig et al., 2020), and natural language generation (NLG) (Peng et al., 2020). Bapna et al. (2017) and Shah et al. (2019) utilized slot descriptions for improving the zero-shot slot filling performance. Rastogi et al. (2020) used slot, intent, and task-specific API descriptions for schema-guided dialogue state tracking. Mosig et al. (2020) based on a structural schema in graph representations instead of textual descriptions for zero-shot action prediction and NLG. Peng et al. (2020) pre-trained on massive text data followed by dialog act labeled dialogue utterances. Madotto et al. (2020) used a large-scale pre-trained language model as a few-shot learner with task-specific prompting. All the methods presented above, however, are geared towards specific tasks and are not shown to generalize to open-domain social or knowledge-grounded conversation.

3 Notation

We conduct experiments under two settings: conversations without explicit knowledge-grounding (we call them *open-domain*) and knowledge-grounded conversations.

3.1 Open-domain conversations

We define a multi-turn conversation as a list of utterances: U_1, U_2, ..., U_N where U_i is the utterance at turn i, and N is the number of turns in the conversation. Each utterance is composed of words w_1, ..., w_M, where M is the number of words in the utterance. Conversational agents are given a subset of the dialog context, for example the t most recent turns U_{N-t-1}, ..., U_{N-1} and generate the response U_N.

3.2 Knowledge-grounded conversations

To formulate knowledge-grounded responses, conversational systems need two steps (sometimes taken jointly): knowledge selection and response generation (Dinan et al., 2019). The conversational agent should therefore first select relevant knowledge pieces from the sources provided with respect to the current dialog context and then generate a response that incorporates the selected knowledge. A knowledge piece in our case is defined as a fact consisting of one or more sentences (see Table 8 for some examples). To select a knowledge retrieval method, we conducted preliminary experiments comparing TF-IDF, BM25, and BERTScore and we saw that the more sophisticated parsing and dense retrieval methods did not outperform TF-IDF. We therefore represent conversation context and knowledge using TF-IDF vectors and utilize TF-IDF-based retrieval over documents as our knowledge selection mechanism. We select the most relevant knowledge using cosine similarity with the context $C = U_{N-t-1}, ..., U_{N-1}$:

$$k_N = \underset{k}{\operatorname{argmax}}\{\cos(t_C, t_k)\} \quad (1)$$

where t_C is the TF-IDF vector corresponding to the context and t_k is the vector corresponding to knowledge piece k. Knowledge-grounded conversational agents are given not only the dialog context C but also the selected knowledge k_N (or multiple pieces of knowledge as in our case) and are asked to generate a response U_N that incorporates k_N.

4 Generative Conversational Networks

GCN (Papangelis et al., 2021) (Figure 1) consist of two models in a meta-learning architecture: a data generator and a learner. The generator creates a labeled dataset that is used to train a new learner (a conversational agent in our case) in a supervised fashion. The learner is then evaluated on an external validation set and its performance is used as a proxy for the quality of the dataset. This quality measure is used as a reward in a RL setup that trains the generator. Over time, the generator learns to create data of better and better quality, with respect to the learner's task, leading the learner to perform well. To avoid overfitting the validation set, we can limit the number of meta-iterations or include domain-independent performance metrics, such as fluency, perplexity, or even human feedback. When deployed, the generator is directly optimized on the test set (i.e. real interactions). Both models can be pre-trained with seed data, if available, and paired with reward estimation, GCN can be used for continuous learning from user feedback. This approach has been proven to work well for intent detection and slot tagging in goal-oriented conversations (Papangelis et al., 2021) and we here apply it to train social conversational agents. Different from Generative Adversarial Networks (Goodfellow et al., 2014)[3] where the model tries to mimic the data, GCN models are guided by an external reward signal - that does not need to be differentiable - and can therefore generalize better. Depending on the optimization criteria, we can set the direction towards which the models will go, for example more polite conversations, more technical terminology, different dialect, knowledge grounding, and even directions that are not easily quantifiable (e.g. engagingness ratings from humans).

For open-domain conversations, as a proof of concept, we conduct few-shot experiments using 10% of the data and for knowledge-grounded conversations which is the main focus of this work, we use 1%, 5%, and 10% of the data; we call these the seed data (D_{seed}). At the beginning of training, we sample D_{seed} from the data D, fine-tune the generator on D_{seed} (see $G.train(D_{seed})$, line 4 in Algorithm 1), and then start the outer loop meta-iterations. Along with the training data, we sample the corresponding percentage of validation data D_{val}. Once the training is complete, we spawn a new learner, train it on the seed and synthetic data, and evaluate it on D_{test} which has been unseen so far. As described earlier, each meta-iteration has four phases: data generation, learner fine-tuning, learner evaluation, and generator update. Algorithm 1 summarizes the process.

[3] A direct comparison with GAN approaches is out of scope for this work and we leave it for the future.

4.1 Data generation

In the first phase of the process, the generator G is given some dialog context sampled from D_{seed} and, in the knowledge-grounded condition, top-m retrieved knowledge pieces k from the TFIDF retriever. Specifically, we give the last two turns as context and the top-3 matching knowledge pieces, and ask the generator to predict the next system response. At each turn i, the context C_i is used to retrieve relevant knowledge k_i that is then used as input to the generator which produces the next turn response U_i:

$$\begin{aligned} U_i &= G(C_i, k_i) \\ &= \bigcup_{w=0}^{n} \{sample(P_{LM}(w|w_{n-1}, ..., w_0, c_i, k_i)\} \end{aligned} \quad (2)$$

where P_{LM} is the probability of the underlying language model generating each word w of the response U_i, and *sample* is the method we use to sample from the PLM, (greedy, nucleus, etc). This way, the generator produces a synthetic dataset D_{synth} of size L, where each datapoint is a triplet of context C_i, knowledge k_i, and response U_i:

$$D_{synth} = \{(C_i, k_i, U_i), i = 1, ..., L\} \quad (3)$$

In essence, to create D_{synth}, instead of taking the human response from the data as a target, we use the generated response U as a target and feed that along with C and k to fine-tune the learner.

4.2 Learner fine-tuning and evaluation

Since the learner's task is knowledge-grounded dialogue, it does not have access to the TFIDF retriever and, as k may contain multiple relevant knowledge pieces, it will learn to perform its own implicit knowledge selection, not knowing what the exact knowledge piece used to produce U was.

At every iteration, we create a new learner (based on a pre-trained model) and fine-tune it on $D_{seed} \cup D_{synth}$ (see line 10 in Algorithm 1). The knowledge-grounded learners are trained using a combination of cross entropy loss and knowledge retrieval score, specifically, Knowledge F1 (KF1) (Shuster et al., 2021) which measures the F1 score between the produced utterance and the selected knowledge piece. The trained learner is then evaluated (see line 11 in Algorithm 1) and a numerical reward is computed by combining several metrics.

Algorithm 1 GCN training procedure.

1: **procedure** TRAIN(D_{seed}, D_{val}, D_{test}, ϵ)
2: Initialize Generator G
3: **if** D_{seed} **then**
4: G.train(D_{seed})
5: **end if**
6: Performance$_{meta} \leftarrow 0$
7: **while** Performance$_{meta} < 1 - \epsilon$ **do**
8: $D_{synth} \leftarrow G$.generate()
9: Sample and initialize new Learner l
10: l.train($D_{seed} \cup D_{synth}$)
11: Performance$_{meta} \leftarrow l$.evaluate(D_{val})
12: ▷ Performance$_{meta} \in [0, 1]$
13: G.update(Performance$_{meta}$)
14: **end while**
15: $D_{synth} \leftarrow G$.generate()
16: Sample and initialize new final Learner L
17: L.train($D_{seed} \cup D_{synth}$)
18: L.evaluate(D_{test}) ▷ or other evaluator
19: **end procedure**

4.3 Generator update

Following (Ziegler et al., 2019) and (Papangelis et al., 2021), we use Proximal Policy Optimization (PPO) (Schulman et al., 2017) with the following modified reward R to train the generator using the learner's validation performance r:

$$R(C, U) = r(C, U) - \beta \log \frac{G(U|C)}{G_{ref}(U|C)} \quad (4)$$

where C represents the context including the knowledge if applicable, U represents the model's response, and β is a constant that prevents G from diverging too much from a reference generator G_{ref}.

In the open-domain condition, the generator uses multiple losses to calculate r: BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and BERTScore (Zhang et al., 2020a) which measure the similarity of the learner-produced utterance and the utterance in the data (D_{seed} or D_{synth}). We evaluate each learner on the validation set D_{val} and compute the above metrics using the human responses in D_{val} as references. The weighted sum of the NLG metrics comprises the reward for the generator training. The weights were determined via grid search: 0.1, 0.01, 0.95, for BLEU, ROUGE-L and BERTScore, respectively. In the knowledge-grounded condition, we use a combination of BLEU-1 and KF1 (with weights 0.75 for

BLEU-1 and 0.25 for KF1) as we found via grid search that it produced better results.

After the meta-iterations are finished, we pick the best performing generator checkpoint (measured by the learners' performance on D_{val} at each meta-iteration) and create a final synthetic set D_{final_synth} that is 5 times the size of the seed. We then create a new learner as our final learner (i.e. the conversational agent) and fine-tune it on $D_{seed} \cup D_{final\_synth}$ (lines 15-18 in Algorithm 1). If D_{final_synth} is of good quality, we should expect the final learner to outperform the baseline, as it is trained with more data. The results presented next are all computed on the final learners, trained for 3 epochs, evaluated on D_{test}, and averaged over 3 runs (as are our baselines).

5 Experiments

To evaluate GCN as a data augmentation method for conversations with and without knowledge, we conduct few-shot experiments on Topical Chat (TC) (Gopalakrishnan et al., 2019). TC is a set of human-human conversations, without explicitly defined roles for each participant, collected over Amazon Mechanical Turk. Each participant had access to a set of facts or articles with some conversations being symmetric (participants had access to the same knowledge) and some being asymmetric. All experiments were conducted on 2 Tesla V100 GPUs with 32GB memory each.

5.1 Model ablations

To quantify the effect of data augmentation and RL in both conditions, we train BART (Lewis et al., 2020) or BlenderBot-small (BBs)[4] (Roller et al., 2021) models for no-knowledge and knowledge-grounded conversations respectively, under the following conditions:

- **Baseline (BART/BBs)**: In this condition, we train BART or BBs on the seed data. This will give us a lower bound on performance (if the augmented data is good, it should help performance).

- **Data augmentation without RL (GCN-RL)**: In this condition, we pre-train a DialoGPT-small[5] (Zhang et al., 2020b) generator with the seed data, and use that to generate 5x more data. We then use the seed and generated data to train a final BART or BBs (learner) model depending on the task.

- **Data augmentation with RL (GCN+RL)**: In this condition, we take the GCN-RL generator and iteratively update it using RL, as described in section 4. This is the full GCN framework. At the end of the meta-iterations, we take the best-performing generator and use it to create 5x more data. We use the seed and generated data to train a final BART or BBs model.

- **Generator direct evaluation (G±RL generator)**: For the knowledge-grounded condition, in addition to the above three models, we evaluate the generator by having it directly interact with humans instead of generating data to train a learner.

5.2 Open-domain conversations

For the open-domain conversations, we sample 10% of TC as seed for GCN and use DialoGPT-small and BART as initial models for the generator and the learner, respectively. We compare the performance of the GCN learner and 3 baselines using automated metrics, and also conduct human evaluations. Our baselines are: BART trained with the same seed data (BART 10%), BART trained with the entire training set (BART 100%), and a GCN learner trained on seed and synthetic data but without updating the generator via RL (GCN-RL). Last, we also compare against the human responses that appear in the data ("Data" in Tables 1 and 3).

5.3 Knowledge-grounded conversations

For knowledge grounded conversations, we sample 1%, 5%, and 10% of TC as seed data for GCN. Again we use DialoGPT-small as a generator but we use BBs as our learner. We compare the performance of GCN against similar baselines to the open-domain condition: BBs trained on the seed or the entire data, GCN without RL, human responses from the data, and we also evaluate the generators themselves if we were to use them directly as conversational agents (G±RL generator). Even though KF1 is the metric of choice in related work on knowledge-grounded conversations, we did not find works that report KF1 for TC.

[4] 90M parameters
[5] 117M parameters

Model	BLEU	Rouge(1/2/L)	BScore	Engaging.	Fluency	Relevance	Overall
Data	-	-	-	3.85	4.55	3.77	4.06
BART (100%)	3.1	20.3/6.1/17.8	0.861	3.80	4.58	3.68	4.02
BART (10%)	**2.0**	**18.5/4.2/16.0**	**0.858**	3.63	4.50	**3.62**	3.92
GCN-RL	1.1	15.0/2.1/12.6	0.850	3.70	4.47	3.47	3.88
GCN+RL	1.3	15.8/2.7/13.6	0.851	**3.79**	4.49	3.58	**3.96**

Table 1: Automatic and human evaluation results. Human evaluators rate responses on a scale of 1 to 5. BScore stands for BERTScore. Bold indicates statistically significant difference (t-test assuming unequal variance). BART (100%) and BART (10%) are BART trained on 100% and 10% of the data, GCN-RL is GCN without RL, and GCN+RL is GCN with RL training.

	1% data			5% data			10% data		
Model	PPL	KF1	BL-4	PPL	KF1	BL-4	PPL	KF1	BL-4
BBs	23.39	0.10	0.07	23.52	0.17	0.09	21.69	0.17	0.09
GCN-RL	26.47	0.15	0.08	24.54	0.18	0.09	23.11	0.18	0.09
GCN+RL	27.11	**0.20**	0.08	24.60	**0.25**	0.14	23.67	**0.28**	0.10

Table 2: Results of automated evaluation on knowledge-grounded conversations. All models try to maximize KF1, and the baseline is the same model as the GCN learners (BBs: BlenderBot-small, 90M parameters).

Model	Eng.	Flu.	Rel.	Avg
Data	3.74	3.98	3.57	3.76
BBs (100%)	3.69	3.99	3.57	3.75
BBs (1%)	3.64	3.86	3.42	3.64
G-RL generator	3.47	3.35	3.23	3.35
G-RL learner	3.58	3.85	**3.48**	3.64
G+RL generator	3.37	3.27	3.40	3.35
G+RL learner	**3.73**	**3.97**	**3.48**	**3.73**
	Wins Percentage			
Combinations	**Base**	**G-RL**	**G+RL**	**Tie**
BBs VS G-RL	40.0	**44.3**	-	15.7
BBs VS G+RL	44.7	-	**47.7**	7.6
All 3 models	29.3	25.7	**45.0**	-

Table 3: Human evaluation results (top) for knowledge-grounded conversations. Human evaluators rate responses with the same conversation context on a scale of 1 to 5. In a different evaluation (bottom), they were asked to choose the best response from two options. BBs: BlenderBot-small (90M), G-RL: GCN without RL, G+RL: GCN with RL.

6 Results

6.1 Automatic evaluation

We report perplexity (PPL), BLEU-4 (Papineni et al., 2002) with the "method 7" smoothing function from (Chen and Cherry, 2014) as it has higher correlation with human ratings, and KF1. We calculate these metrics on the TC "frequent" test set, (Tables 1 and 2). In the open-domain condition, we see that BART 10% outperforms GCN agents on all automated metrics. In knowledge-grounded conversations, we see that GCN+RL is able to incorporate more knowledge as evidenced by the higher KF1.

6.2 Human evaluation

Due to the intrinsic one-to-many property of conversation, reference-based metrics may not correlate with human ratings; our generated conversation may be appropriate for the dialogue context but different from the reference responses. For this reason, we also conduct human evaluation (following sub-section). Human evaluators rate the output of the GCN learner, the baselines, and the ground truth. Specifically, they rate how engaging, fluent, and relevant each response is, on a scale from 1 to 5. We generate 1,000 samples for each condition using the same context and make sure we have 3 ratings per sample per condition. Tables 1 (right) and 3 show the results of the evaluation, where we see that in the open-domain condition, the GCN learner produces engaging but less relevant conversations. This is likely because the model inserts facts or other output that is not entirely relevant, but is perceived as more engaging (e.g. information on a somewhat relevant subject, fun fact, etc.). Consistent with prior work, (Papangelis et al., 2021), this shows that GCN can generalize from the data. When it comes to knowledge-grounded conversations, where GCN is explicitly trained to optimize KF1 (among other metrics), then relevance is indeed higher than the baseline. Overall, averaging the three metrics, GCN+RL outperforms BART 10% and is close to BART 100%'s performance. All models are outperformed by the human responses, which may be due to the size of our models or the number of training iterations.

Iterations	PPL	KF1	BL-4
1	30.8	0.146	0.179
2	31.1	0.147	0.182
3	30.7	0.146	0.186
5	30.8	0.163	0.190
10	27.1	0.238	0.085

Table 4: Performance of GCN+RL for varying number of meta-iterations. Here, we generate 3x the seed data and use 1% of TC.

Data Mult.	PPL	KF1	BL-4
1	26.5	0.201	0.082
2	27.4	0.213	0.084
3	28.6	0.17	0.083
5	22.2	0.25	0.154
10	22.9	0.27	0.106

Table 5: Performance of GCN+RL for varying size of generated data (as a multiplier of the seed). Here, we do 5 meta-iterations and use 1% of TC.

For knowledge-grounded conversations (Table 3) we see that GCN+RL produces more engaging and fluent conversations and overall outperforms both baselines while again being close to BBs trained on all the data. In pair-wise comparisons, GCN+RL is generally preferred more than the other models. Overall, for the GCN conditions, given that we generate 5x the seed data, the total amount of data is about 6% of the size of TC and our results show that the generated data is indeed of high quality, since the same model (BlenderBot-small) using the generated data performs close to the one that uses 100% of the human-human data and close to the data itself. It should be noted that GCN achieves this performance using small models (in the order of 100M parameters each).

In Figure 2 in the appendix, we show the Amazon Mechanical Turk setup that we used during our human evaluations.

6.3 Generated data diversity

In this section we further analyze the performance of GCN, specifically its performance with respect to the number of meta-iterations (Table 4) and the amount of generated data (Table 5). In Table 4, we see that KF1 increases as we have more meta-iterations, meaning that the generator actually leads the learner to learn to produce more knowledgable responses. BLEU naturally drops as these more knowledgable responses may not appear in the data.

Data %	BBs	GCN-RL	GCN+RL
1%	8.1%	17.4%	25.1%
5%	8.5%	12.1%	24.5%
10%	5.9%	9.2%	13.6%

Table 6: Out-Of-Vocabulary (OOV) rates for various seed percentages.

We observe similar trends in Table 5, where we vary the amount of synthetic generated data (as a multiplier of the size of the seed data). Regarding data diversity, Table 6 presents out of vocabulary rates for all three conditions when using 1%, 5%, and 10% of the data as seed. Higher rates mean more diversity but may also mean that the generated data is farther from the seed data. Paired with the results in Tables 1-3, however, we can see that GCN+RL produces more diverse data that are still in-domain and useful.

6.4 Examples

In Table 7, we provide example responses for conversations without knowledge grounding, and in Table 8 we show example responses for knowledge-grounded conversations. We see that in both cases GCN+RL tries to insert knowledge or fun facts and that appear to be received well by the human judges. However, since there are no knowledge-grounding abilities, the model hallucinates and this is what likely drives relevance ratings down. In the knowledge-grounded example, we see that GCN+RL can use the knowledge pieces more effectively than the other models.

7 Conclusion

We presented a meta-learning method based on GCN to generate conversational data grounded on unstructured textual knowledge such as what can be found on the web. We show that given a small seed (1% of TC or 86 examples) our approach can generate high quality data that can be used to train a competitive conversational agent able to do knowledge selection and grounding. Lower reference-based metric scores (Table 1, left and Table 2) indicate that our method generates diverse data. Based on the good scores in the human evaluation (Table 1, right and Table 3), we see that the generated data is also useful for the task at hand, a result in line with the results on intent detection and slot tagging in (Papangelis et al., 2021). As future work, we are conducting thorough analyses and evaluations

Context	
	Hello, do you watch football?
	Hi, yes. I love football! I did not realize the season lasted 17 weeks!
	Yeah pretty long. I guess with 32 teams playing it takes that much time
	Did you see they only rest one weekend during the season?
	That is crazy, so much work goes in there. Did you know U of Iowa painted their visitor locker room pink?
	Oh yes,that is so funny. they really want to intimidate the other team, but is more like a joke
	Wow, what an idea. Did you know bowlers used to make more than football players in the 60s
Responses	
Baseline	that's a good point. i wonder how many of the top bowlers make twice as much as top football stars. do you have a favorite
GCN-RL	that's a good point. i'm not a big fan of football, but i do like to watch it. do you like football?
GCN+RL	i didn't know that, that's interesting. did you know there are more than 30 million players in the nfl? i wonder how many of those players are still playing?

Table 7: Example responses for open-domain conversations. The models here are **not** trained to optimize knowledgable responses and therefore may hallucinate. This may be the reason for lower relevance scores for GCN+RL (Table 1); however, GCN+RL's tendency to provide fun facts is what may be leading to more engaging responses.

Context	
	Did you know that the University of Iowa's locker room is painted pink? I wonder why?
	I think I did hear something about that. I imagine it is an attempt to psych the other team out.
	So, it would be in the visiting team's locker room but not their own?
Retrieved Knowledge (subset)	
	the University of Iowa's locker room for visiting football teams is completely painted pink in the 1960's top bowlers made twice as much as top football stars
	Football is a family of team sports that involve, to varying degrees, kicking a ball [...]
	in 39 out of 50 states, the highest paid state employee is either a college football or basketball coach.
Responses	
Baseline	i think it would be in the visiting team's locker room. i don't know why they would want to psych the other team out
GCN-RL	that's interesting. i've never heard that before. do you have a favorite team?
GCN+RL	i'm not too familiar with iowa's locker room for visiting football teams, but i do know that they're the highest paid employees in the state of iowa. do you know why they are paid so much?

Table 8: Example responses for knowledge-grounded conversations. In this setting, models are trained to optimize knowledgable responses which explains in part why GCN+RL is preferred in human judgements (Table 3).

of the generated data, exploring more model architectures, and are working on more conversational phenomena.

References

Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, pages 1151–1155. ISCA.

Ankur Bapna, Gökhan Tür, Dilek Hakkani-Tür, and Larry P. Heck. 2017. Towards zero-shot frame semantic parsing for domain scaling. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 2476–2480. ISCA.

Markus Bayer, Marc-André Kaufhold, Björn Buchhold, Marcel Keller, Jörg Dallmeyer, and Christian Reuter. 2022. Data augmentation in natural language processing: a novel text generation approach for long and short text classifiers. *International Journal of Machine Learning and Cybernetics*, pages 1–16.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Association for Computational Linguistics*.

Ernie Chang, Xiaoyu Shen, Dawei Zhu, Vera Demberg, and Hui Su. 2021. Neural data-to-text generation

with lm-based text augmentation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 758–768. Association for Computational Linguistics.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.

Derek Chen and Zhou Yu. 2021. GOLD: improving out-of-scope detection in dialogues using data augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 429–442. Association for Computational Linguistics.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6045–6049. IEEE.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Song Feng, Kshitij Fadnis, Q Vera Liao, and Luis A Lastras. 2020. Doc2dial: a framework for dialogue composition grounded in documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13604–13605.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard H. Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 968–988. Association for Computational Linguistics.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tür, and Amazon Alexa AI. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. In *INTERSPEECH*, pages 1891–1895.

Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. 2021. NeuralWOZ: Learning to collect task-oriented dialogue via model-based simulation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3704–3717, Online. Association for Computational Linguistics.

Manoj Kumar, Varun Kumar, Hadrien Glaude, Cyprien de Lichy, Aman Alok, and Rahul Gupta. 2021. Protoda: Efficient transfer learning for few-shot intent classification. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 966–972. IEEE.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Bohan Li, Yutai Hou, and Wanxiang Che. 2021. Data augmentation approaches in natural language processing: A survey. *CoRR*, abs/2110.01852.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gašić. 2021a. Domain-independent user simulation with transformers for task-oriented dialogue systems. *arXiv preprint arXiv:2106.08838*.

Hsien-Chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gasic. 2021b. Domain-independent user simulation with transformers for task-oriented dialogue systems. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2021, Singapore and Online, July 29-31, 2021*, pages 445–456. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2018. End-to-end learning of task-oriented dialogs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 67–73.

Andrea Madotto, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Language models as few-shot learner for task-oriented dialogue systems. *arXiv preprint arXiv:2008.06239*.

Shikib Mehri and Maxine Eskenazi. 2021. Gensf: Simultaneous adaptation of generative pre-trained models and slot filling. *SIGDial*.

Biswesh Mohapatra, Gaurav Pandey, Danish Contractor, and Sachindra Joshi. 2020. Simulated chats for task-oriented dialog: Learning to generate conversations from instructions. *arXiv e-prints*, pages arXiv–2010.

Johannes EM Mosig, Shikib Mehri, and Thomas Kober. 2020. Star: A schema-guided dialog dataset for transfer learning. *arXiv preprint arXiv:2010.11853*.

Alexandros Papangelis, Karthik Gopalakrishnan, Aishwarya Padmakumar, Seokhwan Kim, Gokhan Tur, and Dilek Hakkani-Tur. 2021. Generative conversational networks. In *SIGDial*.

Alexandros Papangelis, Yi-Chia Wang, Piero Molino, Gokhan Tur, and AI Uber. 2019. Collaborative multi-agent dialogue model training via reinforcement learning. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 92.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. SOLOIST: building task bots at scale with transfer learning and machine teaching. *Trans. Assoc. Comput. Linguistics*, 9:907–824.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 172–182.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 300–325. Association for Computational Linguistics.

Gözde Gül Sahin. 2022. To augment or not to augment? A comparative study on text augmentation techniques for low-resource NLP. *Comput. Linguistics*, 48(1):5–42.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152.

Timo Schick and Hinrich Schütze. 2021. Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6943–6951. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Darsh Shah, Raghav Gupta, Amir Fayazi, and Dilek Hakkani-Tur. 2019. Robust zero-shot cross-domain slot filling with example values. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5484–5490.

Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Building a conversational agent overnight with dialogue self-play. *arXiv preprint arXiv:1801.04871*.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3784–3803. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47.

Bing Wang, Liang Ding, Qihuang Zhong, Ximing Li, and Dacheng Tao. 2022a. A contrastive cross-channel data augmentation framework for aspect-based sentiment analysis. *arXiv preprint arXiv:2204.07832*.

Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy. Association for Computational Linguistics.

Yufei Wang, Can Xu, Qingfeng Sun, Huang Hu, Chongyang Tao, Xiubo Geng, and Daxin Jiang. 2022b. Promda: Prompt-based data augmentation for low-resource NLU tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4242–4255. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Qingyang Wu, Song Feng, Derek Chen, Sachindra Joshi, Luis A Lastras, and Zhou Yu. 2021. Dg2: Data augmentation through document grounded dialogue generation. *arXiv preprint arXiv:2112.08342*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020a. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020b. Dialogpt: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Association for Computational Linguistics: System Demonstrations*, pages 270–278.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

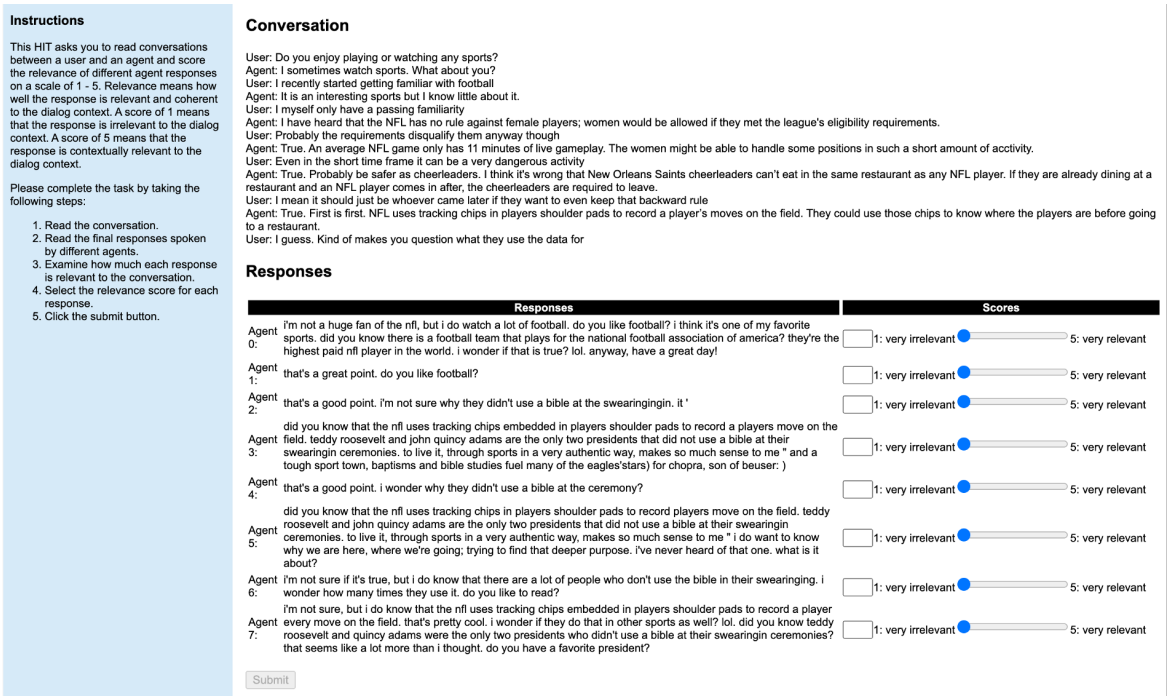


Figure 2: Our human evaluation setup.

A Limitations and Risks

Given that GCN relies on Reinforcement Learning, it may be difficult to tune for more complex applications. Also, due to its meta-learning nature it can be computationally expensive, although we show it here working with small versions of each model.

As with any machine learning approach - especially ones that rely on PLMs - there are inherent risks as to the language that will be produced, which may be biased in many ways. Our approach works with any machine learning model and therefore inherits the risks of the models used. Reinforcement Learning can also play a role in learning biased models if we are not careful.

B Amazon Mechanical Turk Setup

In Figure 2 we show a screenshot of our Amazon Mechanical Turk setup for human evaluation.

Guiding the Release of Safer E2E Conversational AI through Value Sensitive Design

A. Stevie Bergman
Responsible AI, Meta
bergman.as@gmail.com

Gavin Abercrombie
Heriot-Watt University

Shannon Spruit
Independent Ethics Advisor
Populytics, Netherlands

Dirk Hovy
Bocconi University

Emily Dinan
FAIR, Meta

Y-Lan Boureau
FAIR, Meta

Verena Rieser
Heriot-Watt University
Alana AI

Abstract

Over the last several years, end-to-end neural conversational agents have vastly improved their ability to carry unrestricted, open-domain conversations with humans. However, these models are often trained on large datasets from the Internet and, as a result, may learn undesirable behaviours from this data, such as toxic or otherwise harmful language. Thus, researchers must wrestle with how and when to release these models. In this paper, we survey recent and related work to highlight tensions between values, potential positive impact, and potential harms. We also provide a framework to support practitioners in deciding whether and how to release these models, following the tenets of value-sensitive design.

1 Introduction

The social impact of natural language processing and its applications has received increasing attention within the NLP community (e.g. Hovy and Spruit, 2016) with Large Language Models (LLMs) as one of the recent primary targets (e.g. Bender et al., 2021; Bommasani et al., 2021; Weidinger et al., 2021). This paper examines what considerations are salient when designing and releasing *conversational AI* (ConvAI) models. We focus on neural conversational response generation models that are trained on open-domain dialog data and lack a domain-specific task formulation, but instead are designed to freely and engagingly converse about a wide variety of topics. These models are typically trained in the popular encoder-decoder paradigm, which was first introduced for this task by Vinyals and Le (2015); Shang et al. (2015); Serban et al. (2016). We call conversational models trained in this paradigm *end-to-end* (E2E) systems because they learn a hidden mapping between input and output without an interim semantic representation. An important benefit of E2E ConvAI models trained in this paradigm is that they can be adapted to new domains or taught new skills just by fine-tuning a pre-trained model on datasets of interest (e.g. Roller et al., 2020; Smith et al., 2020; Solaimon and Dennison, 2021). Releasing these pre-trained models thus allows different groups of researchers to build on the work of others, which can increase reproducibility and progress. Unfortunately, releasing a model can also have harmful impacts.

We discuss a subset of ethical challenges related to the release and deployment of these models, which we summarise under the term "safety," and highlight tensions between potential harms and benefits resulting from such releases. This is particularly salient in light of recently proposed AI regulation in the European Union (European Commission, 2021). While several recent efforts have been made to describe and mitigate unsafe behaviour of conversational models (e.g. Dinan et al., 2019; Xu et al., 2021; Ouyang et al., 2022; Thoppilan et al., 2022; Perez et al., 2022; Dinan et al., 2022), this work aims to provide a framework to help practitioners think through the conflicts and tensions that arise when designing a conversational model and deciding whether or not to release it, and how.

Releasing models "safely" is particularly challenging for the research community. The concept of "safe language" varies from culture to culture and person to person. It may shift over time as language evolves and significant cultural or personal events provide new context for the usage of that language. In addition, the downstream consequences may not be fully known *a priori*, and may not even be felt for years to come. This is particularly true for large interactive E2E models, where the space of possible generated replies is both extremely vast and highly dependent on context, and can therefore not be exhaustively explored before release. Researchers are then left with the task of trying to arbitrate between uncertain, changing, and conflicting values when making decisions about creating

Proceedings of the SIGdial 2022 Conference, pages 39–52
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

and releasing these models.

We propose ways to conceptualise the interaction of values at play in conversational models (section 3). Based on that understanding, we present a conceptual analytical framework to guide researchers and practitioners towards making better-informed decisions about model release (section 4). We aim to move away from a notion of safety that is based on "the absence of risk" to a more resilience-based notion of safety that is focused on the ability of sociotechnical systems (i.e., users, developers, and technology combined) to anticipate new threats and value changes.

2 Safety problems and mitigations in E2E conversational AI models

We first illustrate some possible sources of safety concerns for ConvAI models through concrete examples grounded in references to existing work – pointing out similarities and differences in issues shared with LLMs. We mainly distinguish ConvAI and generative LLMs by their usage: We refer to ConvAI models if they are used interactively and take an active role as the interlocutor in a dialogue, whereas we refer to LLMs if models are mainly used to generate text, e.g., via text completion or via prompting.

2.1 Training models

While we focus mainly on model release, many of our considerations also apply to earlier stages of training a model, particularly as early choices can have downstream effects that impact elements of the cost-benefit analyses of the researchers. For example, for LLMs and ConvAI systems alike, the type of data used during training might influence what populations could benefit from or be harmed by release of a model (Bender et al., 2021). In addition, training large neural networks on vast amounts of data, leading to high energy consumption and environmental costs (Strubell et al., 2019; Bender et al., 2021). Furthermore, the data used to train models can be insufficiently protected, leading to the leakage of sensitive information through model generations and privacy breaches as happened recently with commercial chatbot Lee-Luda (Jang, 2021). Similar privacy problems are observed for LLMs (e.g. Nasr et al., 2019; Shokri et al., 2017; Carlini et al., 2019, 2020).

2.2 Offensive content

Once trained, a conversational generative model can give rise to safety sensitive situations, by directly generating toxic or otherwise harmful content, by agreeing with offensive statements uttered by the conversation partner (Dinan et al., 2022), or by responding defensively or dismissively when provided with corrective feedback by the conversation partner (Ung et al., 2021). While the first case is shared with LLMs, the latter two are unique to ConvAI systems. Generating this type of content can cause harm to users, and poses a reputational risk to the organisation releasing the model, for instance when the bot voices undesirable or controversial opinions, e.g., Tay's anti-semitic stances (Miller et al., 2017).

The boundaries of what is offensive or not are both subjective and culturally dependent. This makes it especially important to consider what community norms are applicable when deploying a model (Jurgens et al., 2019; Sap et al., 2019; Kiritchenko and Nejadgholi, 2020; Liang et al., 2022), and whether the use of labels might not be a risk in itself (Thylstrup and Waseem, 2020).

Many existing mitigations rely on the ability to detect problematic content – often centred on content written by humans on social media platforms, such as Twitter (e.g. Waseem and Hovy, 2016; Wang et al., 2020; Zampieri et al., 2019, 2020; Zhang et al., 2020), Facebook (Glavaš et al., 2020; Zampieri et al., 2020), or Reddit (Han and Tsvetkov, 2020; Zampieri et al., 2020). However, of course, conversational systems may not necessarily have the same patterns as social media content (Cercas Curry et al., 2021). Existing work on conversational systems often relies on identification of keywords (Ram et al., 2017; Cercas Curry et al., 2018; Fulda et al., 2018; Khatri et al., 2018; Paranjape et al., 2020), or uses human labels such as flagging of a post to train classifiers (Larionov et al., 2018; Cercas Curry et al., 2018). These first-pass classifiers can then be augmented adversarially as done in Dinan et al. (2019); Xu et al. (2020).

In addition, work on building safer LLMs explores fine-tuning on curated data (Solaimon and Dennison, 2021) or directly controlling the generations of the model (Dathathri et al., 2019; Liu et al., 2021; Schick et al., 2021; Xu et al., 2020). Conditioning generations on certain types of context, such as personas of diverse historically marginalised demographics, has also been shown

to decrease the generation of harmful responses (Sheng et al., 2021).

2.3 Mitigating the risks of mitigations

LLMs and ConvAI models often rely on a classifier to detect and mitigate unsafe model outputs. However, these classifiers themselves can have issues with bias, e.g., by learning undesirable correlations that tie toxicity to identity terms (Dixon et al., 2018; Nozza et al., 2021, 2022), or language varieties, such as African American English (Liu et al., 2019; Sap et al., 2019). Possible mitigations include using race and dialect priming (Sap et al., 2019), using adversarial training techniques (Xia et al., 2020), adding fairness constraints (Gencoglu, 2020), or relabeling data used during training (Zhou et al., 2021).

2.4 Interacting with users

There are some additional challenges which are unique to ConvAI system arising from the direct interaction with users. This includes the possibility of an involuntary anthropomorphic relationship arising between a conversational model and a human interacting with it (Abercrombie et al., 2021), and the fact that model generations are inherently dependent on the unknown inputs of a conversation partner who will be repeatedly interacting with the systems and steering them in unpredictable directions. Some users have been observed to behave in an adversarial way, as happened for instance with Tay (Miller et al., 2017).

Another empirical pattern is that user utterances in their conversations with chatbots are often abusive (Cercas Curry and Rieser, 2018; Cercas Curry et al., 2021). Thus, the safety implications of the system needs to be considered within the expected conversational context, including adversarial inputs. For example, publicly available chatbots have been shown to agree with sexist or racist utterances (Lee et al., 2019b). Automatically detecting unsafe user utterances is still a challenge, both for system directed abuse (Cercas Curry et al., 2021) and general toxic statements (Xu et al., 2020). A recent report by UNESCO points out that the inability to respond appropriately to system-directed abuse may reinforce negative gender stereotypes (West et al., 2019), especially paired with their anthropomorphic and feminised design cues (cf. Abercrombie et al. (2021)).

The possibility of adversarial interaction and, more generally, the unpredictability of a system used far outside the training distribution, make it particularly important to not exclusively rely on mitigations such as cleaning up training data to avoid exposing the system to offensive content, as it has been shown to still leave models prone to generating toxic content in response to specific prompts (Gehman et al., 2020) or inadequate responses to abuse from users (Cercas Curry and Rieser, 2018).

2.5 Use in unsafe applications

Conversational and language models can also prove unsafe if they are used for medical advice or emergency situations (self-harm, crime, natural disasters, etc) (e.g. Palanica et al., 2019; Bickmore et al., 2018). Conversational systems designed for discussing health issues tend to not be generative models and use expert-produced rather than generic data (e.g. Brixey et al., 2017; Fadhil and AbuRa'ed, 2019; Vaira et al., 2018; Pereira and Díaz, 2019).

A mitigation avenue for E2E ConvAI models is to recognise topics that do not lend themselves to automated conversation, and steer the conversation away from them (Dinan et al., 2022). When using such mitigations, considerations for release might then usefully include how effective the context detection is, and the costs of false negatives (i.e., failing to steer away from an unsafe context), false positives (i.e., refusing to talk about safe topics), and lost opportunity to provide safe benefits, e.g., safe general medical advice such as that generally offered on public health websites.[1]

3 Tensions between values, potential positive impact, and potential harm

After highlighting some existing barriers to the creation of safe ConvAI (as well as possible mitigations), we lay out some important tensions between values, positive impact and potential harm. These considerations establish a foundational understanding of the system, after which we can consider release decisions (discussed in section 4).

There is a growing understanding that computing systems encode values, and will do so whether or not the parties involved in designing and releasing the system are explicitly aware of those values (Friedman et al., 2008; van de Poel, 2018). Reflecting more deliberately on values throughout model development can help surface potential problems and opportunities early on, identify what informa-

[1] For a recent, taxonomy of harms and risks from LLMs, see Weidinger et al. (2021).

tion might be important to communicate as part of a model release, and allow practitioners and downstream users to make better-informed decisions.

We use the broad definition of values employed in Friedman et al. (2008): "what a person or group of people consider important in life." With this definition, values extend beyond the use of the term akin to moral tenets, to the more general *things of value*. Examples relevant to conversational agents could be: getting or providing education, companionship, or comfort, preserving privacy, widening access to more populations through automation – or trust, friendship, accessibility, and universality.

Throughout this section, we employ the scenario of a hypothetical companion: a potential chatbot that leverages the constant availability and scalability of automated systems to provide companionship to people who feel lonely. However, it could raise privacy and consent concerns, e.g., if the conversations are recorded for subsequent improvement of the model without informing the user. Deeper concerns would be that the system might displace human companionship in a way that creates an unhealthy reliance on a bot, a decreased motivation to engage with humans, and a lower tolerance to the limited availability and patience of humans.

3.1 How values conflict

Determining how to best arbitrate between different values requires the consideration of multiple types of conflicts. For example:

Conflicts between values. Some values can be in direct conflict: for example, lowering privacy protections to harvest more detailed intimate conversation data to train a powerful artificial "close friend" system pits privacy against relieving loneliness. These conflicts require deciding on a value trade-off. But even values that are not directly in conflict can require trade-offs, through competition for limited resources and prioritisation of certain goals or values: the resources invested to uphold a given value might have instead enabled a better implementation of another value. Thus, *opportunity costs* (Palmer and Raftery, 1999) need to be considered along with absolute costs.

Conflicts arising from distributional disparities. Besides values in a local setting (i.e., for a single stakeholder, at a single point in time), another source of conflict arises from disparities between stakeholders: who bears the costs and who reaps the rewards? This raises issues of distributional justice (Bojer, 2005). In intertemporal conflicts, the same person may pay a cost and reap a benefit at different points in time. For example, a user electing to contribute their private information now to enable systems they expect to benefit from later.

Arbitrating conflicts. For conflict within an individual stakeholder, the individual should theoretically be able to arbitrate the decision themselves, given relevant information. However, that arbitration would still be subject to ordinary cognitive and motivational biases. These include favouring instant gratification (Ainslie, 2001), and resorting to frugal heuristics to make faster decisions (Kahneman, 2011). Thus, practitioners need to grapple with additional tensions between prioritising users' autonomy (i.e., letting people choose, even if they are likely to choose something they will regret) or users' satisfaction with outcomes of their choices (i.e., protecting people from temptations). In the example of a companion chatbot, one could imagine a system that always tells people what they most want to hear, even if it reinforces unhealthy addictive patterns: would this require regulation like a drug, or would people best be left as the sole autonomous judges of how they want to use such a system? Clever defaults and nudges can help resolve this kind of tension, making it easier for people to choose what may ultimately be better for them (Thaler and Sunstein, 2009).

If costs and benefits allocate to different stakeholder groups, things become even more complex. Values are then compared in terms of the distribution of costs and benefits among stakeholders. For example, the value of fairness demands that distributions not be overly skewed. Utilitarian and rights-based approaches favour different trade-offs between increasing the benefits of a system for a large majority of people at the cost of harming a few, and emphasising preservation of the rights of as many people as possible (Velasquez et al., 2015). If a companion conversational system provides a great amount of comfort to millions of people, but harms a handful, different ethical systems will weigh the good and the bad in different ways and reach dissimilar conclusions. Next, we discuss what processes can achieve a particular desired balance of values and costs, regardless of what that desired balance is.

3.2 Additional Challenges

There are two additional challenges when aiming to balance values: First, *human judgements of risks, costs, and benefits can vary considerably across groups*. These include cognitive heuristics – such as the fact that people tend to have trouble comprehending large numbers and have more of a response to representative narratives (Slovic, 2010) – but also population biases in risk estimation, where white men are often outliers in how they (under)estimate risks (Finucane et al., 2000; Flynn et al., 1994). This discrepancy makes it especially important to pay attention to the demographic make-up of the sample of stakeholders providing a risk estimate. Other related issues is the asymmetry between perception of costs and benefits, where Baumeister et al. (2001) find "bad [events] to be stronger than good in a disappointingly relentless pattern," and that "bad events wear off more slowly than good events." This effect is especially pronounced in algorithmic systems, where people apply higher standards than in their interaction with other humans (Dietvorst et al., 2015). These findings mean that the balance between costs and benefits needs to be strongly tilted towards benefits to appeal to humans subjectively.

The other challenge stems from the *inherent uncertainty and change in safety related concepts*. Early estimates of costs and benefits are often plagued by uncertainty. This includes uncertainty about future use (malicious misuse or unintended use, broader or smaller adoption than planned, etc.), and uncertainty about interaction with an evolving society and other innovations. Beyond uncertainty, van de Poel (2018) draws attention to *value change* and its sources, from the emergence of new values in society to changes in how different values are weighed. As advocated in van de Poel (2018), systems should be designed with a focus on adaptability, robustness, and flexibility. In practical terms for conversational models, this entails the use of rapidly adaptable techniques (e.g., fine-tuning, inference-time control, etc.). It also highlights the importance of continually questioning assumptions on what evaluation methods measure and investing in methods that can evolve from ongoing feedback.

3.3 Value-sensitive design

Value-sensitive design (Friedman et al., 2008) incorporates human values throughout the design process. It adopts an iterative process of **conceptual exploration**, i.e., thinking about relevant values and how they manifest, about who the stakeholders are, and what the tradeoffs between values ought to be); **empirical investigations**, including surveys, interviews, empirical quantitative behavioural measurements, and experimental manipulations; and **technical investigation**, i.e., evaluating how a given technology supports or hinders specific values. Friedman et al. (2017) survey several techniques to help practitioners implement value-sensitive design, such as the *"value dams and flows"* heuristic (Miller et al., 2007). *Value dams* remove parts of the possible universe that incur strong opposition from even a small fraction of people. In contrast, *value flows* attempt to find areas where many people find value. An example of *value dams* would be thresholds on some features, as a way to translate values into design requirements (Van de Poel, 2013). This process is reminiscent of the machine learning practice of constrained optimisation, which combines satisficing constraints and maximising objectives. Van de Poel (2013) reviews how to operationalise values into design requirements.

4 A Framework for Researchers to Deliberate Model Release

The topic of when and how to release LLMs designed by research groups has been of increasing interest to the community (e.g. Solaiman et al., 2019; Crootof, 2019; Ovadya and Whittlestone, 2019; Partnership on AI, 2020; Partnership on AI , 2021; Liang et al., 2022). The case is similar for conversational models, with safety issues in particular posited as a reason for withholding the release of such models. For example, in a blog post about the ConvAI model Meena (Adiwardana et al., 2020) the authors cite safety challenges as a reason for not releasing the model.[2]

Within the broader context of value-sensitive design, and absent responsible release norms in the field (Ovadya and Whittlestone, 2019; Liang et al., 2022), we propose the following elements of a framework to aid researchers in deliberating safer release, and guidance to support learning during and after release.

We ground our discussion in two relevant, theoretical case studies:

[2] https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html accessed 10th May 2022.

- ***Case 1* – Open-sourcing a model:** Researchers train a several billion parameter Transformer encoder-decoder model on (primarily) English-language conversational data from the internet. They publish a peer-reviewed paper on this model. The researchers seek to open-source the weights of their model such that other researchers in the academic community can reproduce and build off of this work.
- ***Case 2* – Releasing a research demo of a model:** The researchers from *Case 1* would additionally like to release a small scale demo of their model through a chat interface on a website. Creating such a demo would allow non-expert stakeholders to interact with the model and gain a better sense of its abilities and limitations.

4.1 Intended use

Explicitly surfacing the intended use of the released model is a simple, but important, initial step. By stating their intentions early in the research, and re-evaluating at stages later in the process, the researchers can track whether their intentions have meaningfully drifted. In accordance with other elements of this framework, researchers can inquire: Is the intended use expected to have "positive impact," and what does that mean in the context of this model? To whom will these benefits accrue? Lastly, is releasing the model in the intended fashion necessary to fulfil the intended use?

At this stage, researchers might further consider uses that do not fall within their conception of the *intended use*. Explicitly deliberating on this might bring to the fore vulnerabilities and possible ethical tensions that could inform the release policies.

In *Case 1*, for example, the researchers' intention may be to advance the state of the art in the field and allow other researchers to reproduce and build off of their work (Dodge et al., 2019). Outside of the intended use, however, the researchers might imagine that – depending on the manner of the release – a user could build a product utilising the released model, resulting in unintended or previously unforeseen consequences. The researchers may then adopt a release policy designed to limit such an unintended use case. In *Case 2*, there are many possible intended uses for releasing such a demo. A primary intention might be to further research on human-bot communication by collecting data (with clear consent and privacy terms) to better understand the functioning and limitations of the model. Alternatively, it may be to simply increase awareness of the abilities and limitations of current neural models among the general public.

4.2 Audience

The consequences of a model being released beyond the research group depend largely on both the intended and unintended audiences of the release, as well as the policies that support and guardrail the research release (subsection 4.6). For conversational AI, the language(s) the model was trained on, the demographic composition and size of the intended audience, and the intended audience's familiarity with concepts and limitations of machine learning and NLP are all important considerations. Policies (subsection 4.6) may be designed to minimize access outside of the intended audience of the release where possible, so as to limit the potential harms of use outside the model's designed scope.

In both *Case 1* and *Case 2*, the model in question is trained primarily on English-language data, and so we might expect the audience to be primarily composed of English speakers, perhaps even those of a particular cultural community or dialect. This consideration is important both for user comprehension and due to the fact that different languages have different ways of expressing and responding to the same concept, like politeness, and different cultures might vary in their evaluation of the same concept. For example, Japanese requires the consideration of the social hierarchy and relations when expressing politeness (Gao, 2005), whereas English can achieve the same effect by adding individual words like "please." Arabic-speaking cultures, on the other hand, might find this use awkward, if not rude, in conversations among close friends (Kádár and Mills, 2011; Madaan et al., 2020).

Furthermore, in *Case 1*, the size of the audience may be hard to gauge *a priori*. On the other hand, in *Case 2*, the researchers/designers would have strict control over the size of the audience. Resulting policy decisions (section 4.6) will differ if the audience is on the scale of tens, hundreds, or millions of people interacting with this technology.

Lastly, in *Case 1*, access to the model may require deep technical knowledge of the programming language the model was implemented in, and as such, the audience would likely (although not definitely) be limited to folks with a working knowledge of machine learning and NLP, while in *Case 2* a more general audience may be able to access

the model. This is important, as a general audience may have different expectations and a different understanding of the limitations of systems (Bianchi and Hovy, 2021). If the targeted audience is the general public, a policy for releasing such a model might explicitly include a means for transparently communicating scope and expectations.

4.3 Envision Impact

The process of envisioning impact – including both potential harms and benefits – is not straightforward, as documented by Ovadya and Whittlestone (2019), Prunkl et al. (2021), Partnership on AI (2020), and Partnership on AI (2021), among others, and it may not always be possible to estimate impact. The goal is to get ahead of potential harms in order to direct tests, mitigation efforts, and design appropriate policies for mitigation and protection, however there must be caution against basing release decisions solely on envisioned harms rather than overall impact (subsection 3.2). This is the *conceptual* exploration of value sensitive design (subsection 3.3), similar in concept to the NeurIPS broader impact statement (NeurIPS, 2020). It benefits from consulting relevant community or domain experts (subsection 4.5). Again, considering the audience of the release matters here, e.g., considering to whom the benefits of the model will accrue and whether it might work less well for (or even harm) some members of the audience/community.

To begin, researchers from *Case 1* and *Case 2* might conduct a review of previous, similar domain research and the resulting impacts: If the research incrementally improves upon previous work, could the impacts be presumed similar to those of previous work? If not, how might those differences lead to divergent impacts (positive and negative)? Perhaps the model exhibits some issues described in section 2. Beyond these, it may be helpful to think outside the box, even constructing a fictional case study (CITP and UHCV) or thought experiment, such as asking: *How would a science fiction author turn your research into a dystopian story?* (Partnership on AI , 2021). Ovadya and Whittlestone (2019) recommend bringing in wider viewpoints (subsection 4.5), such as subject matter experts, for increased understanding of the risk landscape.

4.4 Impact Investigation

After the conceptual exploration of impacts, attempting to measure the *expected* impact can provide quantitative grounding. This means conducting a *technical investigation*, evaluating how the model supports or hinders the prioritised values. We reiterate that it is not always possible to accurately estimate impact, nevertheless, such empirical analyses may guide next steps or appropriate policies. Investigating benefits may be more application-dependent than investigating harms, so we encourage researchers to think through this for their own particular use cases.

The authors in *Case 1* and *Case 2* may estimate the frequency with which and the circumstances under which their model behaves inappropriately using human evaluators or automatic tooling, such as the toolkit provided by Dinan et al. (2022) to detect safety issues, for example. In *Case 2*, the authors may undergo a "dogfooding" process for their demo with a smaller audience that roughly matches the composition of their intended audience.

4.5 Wider Viewpoints

Input from community or domain experts relevant to the model application is highly recommended throughout the model development process, and indeed throughout this framework – from envisioning potential harms, to feedback for the purpose of model improvement – but particularly so in release deliberation to better understand the risk landscape and mitigation strategies (Martin Jr et al., 2020; Ovadya and Whittlestone, 2019; Bruckman, 2020). Researchers could further consider the burgeoning literature on participatory AI methodologies (e.g. Martin Jr et al., 2020; Lee et al., 2019a).

In *Case 1*, the researchers may seek feedback and discussions with researchers or potential users outside of their immediate institution, community, or more formal engagements through employment or a workshop on related topics. Researchers could reach out to stakeholder and advocacy groups for input, where possible. In *Case 2*, researchers might consider an explicit "dogfooding" step to gather feedback from users, as described in subsection 4.4, and expert representatives of social groups.

4.6 Policies

An important aspect of release is whether it is possible to design an effective guard-railing policy to both bolster/maintain the positive outcomes while mitigating any potential negative consequences.

For *Case 1*, in which a model is open-sourced to the research community, policies might include restrictive licensing or release by request only. If released only by request, then researchers who wish

to access the model would be required to contact the model owners. This method upholds the researchers values' of reproducibility while potentially limiting unintended uses, but incurs a possibly high maintenance cost if many researchers send in requests with detailed plans of use which would need to be examined and adjudicated. If multiple model versions exist which might be expected to have differing impacts, the researchers might consider adopting a *staged release* policy, as in Solaiman et al. (2019). This would allow further time and information to aid in technical investigations prior to releasing the version expected to have highest impact. Such a policy would be most effective if users had ample opportunity to provide feedback throughout the release stages.

For *Case 2*, releasing a small demo of a model on a chat interface, the researchers may limit access to the demo to a small group of people above a certain age. This could be enforced through password protection and cutting off access to the demo after a certain number of unique users have interacted with the model. Further, access might be revoked under certain circumstances, e.g., in case new potential for harm is detected and the model needs to be corrected, or abusive access by certain users.

4.7 Transparency

Striving for transparency can help researchers and model users reason through whether their use case is appropriate and worth the risk of engaging with the model (Diakopoulos, 2016). Consider the methodology laid down for Model Cards by Mitchell et al. (2019) to clarify the intended use cases of machine learning models and minimise their usages that fall outside of these parameters.

For *Case 1*, when open-sourcing the model, the authors may consider releasing it with a model card, following the content recommendations from Mitchell et al. (2019). In such a model card they might additionally report the outcome of any investigation into potential harms or benefits.

In *Case 2*, for a small-scale demo, a full model card with abundant technical details may not be effective (see discussion in subsection 3.2), however, the researchers might consider providing some easily-digestible model information – such as the institution responsible for the model, its intended use, any potential harms and policies in place to limit those harms, means for reporting or redress in case of error or harm, or other relevant details. In order to sustain the value of *informed consent*, the researchers might carefully craft the information such that the user is informed that they are interacting with an artificial conversational system, which may be unclear due to the anthropomorphic design cues from these models (Abercrombie et al., 2021).

4.8 Feedback to Model Improvement

Learning systems can produce unexpected outcomes, and thus unforeseen harms. Particularly as the environment (e.g., the world) in which the model is operating changes. Researchers can gain a better grasp on these with accessible and reliable mechanisms to capture unexpected outcomes and changes (e.g., a reporting form for the user to submit). Upon gathering feedback, researchers can then use this information to improve the model in future iterations, or consider how to design their model to be adaptable to changes in values.

In *Case 1*, for example, it may be hard to control or refer to the impact of open-sourcing the model. However, the researchers might consider providing access and encouraging reports of safety issues to a well-monitored GitHub Issues page. In *Case 2*, the researchers should consider how to design the demo UI to empower users to report problems.

Provided meaningful feedback about safety issues with the model in *Case 1* and *Case 2*, the researchers might release an updated version of the model, particularly if the model is designed in a way that makes it able to adapt easily to feedback.

5 Conclusion

Besides the overall challenges posed by large language models, conversational models present specific issues. They are inherently dependent on the unknown inputs of the users who will be repeatedly interacting with the systems and steering them in combinatorially unpredictable directions. The costs and benefits of releasing a model can thus be hard to determine, especially when they only appear after cascades of uncertain consequences at different time scales. Reckoning with these issues requires weighing conflicting, uncertain, and changing values. To aid in this challenging process, we provided a framework to support preparing for and learning from model release, following principles of value-sensitive design. We illustrate each of our proposed steps with concrete, hypothetical scenarios to help practitioners in their reflection.

While this is a theoretical paper, informed by

an interdisciplinary collaboration, we believe in the value of publishing it through an applied conference since this will maximise the chances of reaching our target audience.

Acknowledgements

Thanks to Chloé Bakalar, Miranda Bogen, and Adina Williams for their helpful comments.

Verena Rieser and Gavin Abercrombie's contributions were supported by the EPSRC project 'Gender Bias in Conversational AI' (EP/T023767/1) and Verena Rieser was also supported by the project 'AISEC: AI Secure and Explainable by Construction' (EP/T026952/1).

Dirk Hovy was supported by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (No. 949944, INTEGRATOR). He is a member of the Data and Marketing Insights Unit at the Bocconi Institute for Data Science and Analysis.

References

Gavin Abercrombie, Amanda Cercas Curry, Mugdha Pandya, and Verena Rieser. 2021. Alexa, Google, Siri: What are your pronouns? gender and anthropomorphism in the design and perception of conversational assistants. In *Proceedings of the 3rd Workshop on Gender Bias in Natural Language Processing*, pages 24–33, Online. Association for Computational Linguistics.

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

George Ainslie. 2001. *Breakdown of will*. Cambridge University Press.

Roy F Baumeister, Ellen Bratslavsky, Catrin Finkenauer, and Kathleen D Vohs. 2001. Bad is stronger than good. *Review of general psychology*, 5(4):323–370.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Federico Bianchi and Dirk Hovy. 2021. On the gap between adoption and understanding in NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3895–3901, Online. Association for Computational Linguistics.

Timothy W Bickmore, Ha Trinh, Stefan Olafsson, Teresa K O'Leary, Reza Asadi, Nathaniel M Rickles, and Ricardo Cruz. 2018. Patient and consumer safety risks when using conversational assistants for medical information: An observational study of siri, alexa, and google assistant. *J Med Internet Res*, 20(9):e11510.

Hilde Bojer. 2005. *Distributional justice: Theory and measurement*, volume 47. Routledge.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Kohd, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models.

Jacqueline Brixey, Rens Hoegen, Wei Lan, Joshua Rusow, Karan Singla, Xusen Yin, Ron Artstein, and Anton Leuski. 2017. SHIHbot: A Facebook chatbot for sexual health information on HIV/AIDS. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 370–373, Saarbrücken, Germany. Association for Computational Linguistics.

Amy Bruckman. 2020. 'Have you thought about...': Talking about ethical implications of research. *Communications of the ACM*, 63(9):38–40.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium*

(USENIX Security 19), pages 267–284, Santa Clara, CA. USENIX Association.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2020. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*.

Amanda Cercas Curry, Gavin Abercrombie, and Verena Rieser. 2021. ConvAbuse: Data, analysis, and benchmarks for nuanced abuse detection in conversational AI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7388–7403, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondřej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. 2018. Alana v2: Entertaining and informative open-domain social dialogue using ontologies and entity linking. *Alexa Prize Proceedings*.

Amanda Cercas Curry and Verena Rieser. 2018. # metoo: How conversational systems respond to sexual harassment. In *Proceedings of the Second ACL Workshop on Ethics in Natural Language Processing*, pages 7–14.

Princeton CITP and UHCV. Law enforcement chatbots, case study: 4.

Rebecca Crootof. 2019. Artificial intelligence research needs responsible publication norms. *Lawfare Blog*.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: a simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Nicholas Diakopoulos. 2016. Accountability in algorithmic decision making. *Communications of the ACM*, 59(2).

Berkeley J Dietvorst, Joseph P Simmons, and Cade Massey. 2015. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology: General*, 144(1):114.

Emily Dinan, Gavin Abercrombie, A. Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, and Verena Rieser. 2022. SafetyKit: First aid for measuring safety in open-domain conversational systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4113–4133, Dublin, Ireland. Association for Computational Linguistics.

Emily Dinan, Samuel Humeau, Bharath Chintagunta, and Jason Weston. 2019. Build it break it fix it for dialogue safety: Robustness from adversarial human attack. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4537–4546, Hong Kong, China. Association for Computational Linguistics.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2185–2194. Association for Computational Linguistics.

European Commission. 2021. Proposal for a regulation of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending cerntain union legislative acts. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELLAR:e0649735-a372-11eb-9585-01aa75ed71a1.

Ahmed Fadhil and Ahmed AbuRa'ed. 2019. OlloBot - towards a text-based Arabic health conversational agent: Evaluation and results. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 295–303, Varna, Bulgaria. INCOMA Ltd.

Melissa L Finucane, Paul Slovic, Chris K Mertz, James Flynn, and Theresa A Satterfield. 2000. Gender, race, and perceived risk: The'white male'effect. *Health, risk & society*, 2(2):159–172.

James Flynn, Paul Slovic, and Chris K Mertz. 1994. Gender, race, and perception of environmental health risks. *Risk analysis*, 14(6):1101–1108.

Batya Friedman, David G Hendry, and Alan Borning. 2017. A survey of value sensitive design methods. *Foundations and Trends in Human-Computer Interaction*, 11(2):63–125.

Batya Friedman, Peter H Kahn, and Alan Borning. 2008. Value sensitive design and information systems. *The handbook of information and computer ethics*, pages 69–101.

Nancy Fulda, Tyler Etchart, William Myers, Daniel Ricks, Zachary Brown, Joseph Szendre, Ben Murdoch, Andrew Carr, and David Wingate. 2018. Byueve: Mixed initiative dialog via structured knowledge graph traversal and conversational scaffolding. *Proceedings of the 2018 Amazon Alexa Prize*.

Fengping Gao. 2005. Japanese: A heavily culture-laden language. *Journal of Intercultural Communication*, 10:1404–1634.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Oguzhan Gencoglu. 2020. Cyberbullying detection with fairness constraints. *arXiv preprint arXiv:2005.06625*.

Goran Glavaš, Mladen Karan, and Ivan Vulić. 2020. XHate-999: Analyzing and detecting abusive language across domains and languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6350–6365, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xiaochuang Han and Yulia Tsvetkov. 2020. Fortifying toxic speech detectors against veiled toxicity.

Dirk Hovy and Shannon L Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598.

Heesoo Jang. 2021. A South Korean chatbot shows just how sloppy tech companies can be with user data. https://slate.com/technology/2021/04/scatterlab-lee-luda-chatbot-kakaotalk-ai-privacy.html. Accessed: 1st June 2021.

David Jurgens, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–3666, Florence, Italy. Association for Computational Linguistics.

Dániel Z Kádár and Sara Mills. 2011. *Politeness in East Asia*. Cambridge University Press.

Daniel Kahneman. 2011. *Thinking, fast and slow*. Macmillan.

Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, et al. 2018. Advancing the state of the art in open domain dialog systems through the Alexa prize. *arXiv preprint arXiv:1812.10757*.

Svetlana Kiritchenko and Isar Nejadgholi. 2020. Towards ethics by design in online abusive content detection.

George Larionov, Zachary Kaden, Hima Varsha Dureddy, Gabriel Bayomi T. Kalejaiye, Mihir Kale, Srividya Pranavi Potharaju, Ankit Parag Shah, and Alexander I Rudnicky. 2018. Tartan: A retrieval-based socialbot powered by a dynamic finite-state machine architecture.

Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. 2019a. Webuildai: Participatory framework for algorithmic governance. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW).

Nayeon Lee, Andrea Madotto, and Pascale Fung. 2019b. Exploring social bias in chatbots using stereotype knowledge. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 177–180, Florence, Italy. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Kathleen A. Creel, and Rob Reich. 2022. The time is now to develop community norms for the release of foundation models.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. On-the-fly controlled text generation with experts and anti-experts.

Haochen Liu, Jamell Dacon, Wenqi Fan, Hui Liu, Zitao Liu, and Jiliang Tang. 2019. Does gender matter? towards fairness in dialogue systems. *arXiv preprint arXiv:1910.10486*.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1869–1881, Online. Association for Computational Linguistics.

Donald Martin Jr, Vinodkumar Prabhakaran, Jill Kuhlberg, Andrew Smart, and William Isaac. 2020. Participatory Problem Formulation for Fairer Machine Learning Through Community Based System Dynamics. *CoRR*, abs/2005.07572.

Jessica K Miller, Batya Friedman, Gavin Jancke, and Brian Gill. 2007. Value tensions in design: the value sensitive design, development, and appropriation of a corporation's groupware system. In *Proceedings of the 2007 international ACM conference on Supporting group work*, pages 281–290.

K.W Miller, Marty J Wolf, and F.S. Grodzinsky. 2017. Why we should have seen that coming. *ORBIT Journal*, 1(2).

Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019. Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019*, pages 220–229. ACM.

Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 739–753.

Neural Information Processing Systems Conference NeurIPS. 2020. Getting started with NeurIPS 2020.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2021. HONEST: Measuring hurtful sentence completion in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406, Online. Association for Computational Linguistics.

Debora Nozza, Federico Bianchi, Anne Lauscher, and Dirk Hovy. 2022. Measuring harmful sentence completion in language models for LGBTQIA+ individuals. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 26–34, Dublin, Ireland. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Aviv Ovadya and Jess Whittlestone. 2019. Reducing malicious use of synthetic media research: Considerations and potential release practices for machine learning. *arXiv preprint arXiv:1907.11274*.

Adam Palanica, Peter Flaschner, Anirudh Thommandram, Michael Li, and Yan Fossat. 2019. Physicians' perceptions of chatbots in health care: Cross-sectional web-based survey. *J Med Internet Res*, 21(4):e12887.

Stephen Palmer and James Raftery. 1999. Opportunity cost. *Bmj*, 318(7197):1551–1552.

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*.

Partnership on AI . 2021. Managing the risks of ai research: Six recommendations for responsible publication.

Partnership on AI. 2020. Publication norms for responsible ai: Ongoing initiative.

Juanan Pereira and Óscar Díaz. 2019. Using health chatbots for behavior change: A mapping study. *Journal of Medical Systems*, 43(5).

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models.

Carina Prunkl, Carolyn Ashurst, Markus Anderljung, Helena Webb, Jan Leike, and Allan Dafoe. 2021. Institutionalizing ethics in AI through broader impact requirements. *Nature Machine Intelligence*.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, Eric King, Kate Bland, Amanda Wartick, Yi Pan, Han Song, Sk Jayadevan, Gene Hwang, and Art Pettigrue. 2017. Conversational AI: The science behind the Alexa Prize. In *Proceedings of Workshop on Conversational AI*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *CoRR*, abs/2103.00453.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, volume 16, pages 3776–3784.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586, Beijing, China. Association for Computational Linguistics.

Emily Sheng, Josh Arnold, Zhou Yu, Kai-Wei Chang, and Nanyun Peng. 2021. Revealing persona biases in dialogue systems. *CoRR*, abs/2104.08728.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18.

Paul Slovic. 2010. If i look at the mass i will never act: Psychic numbing and genocide. In *Emotions and risky technologies*, pages 37–59. Springer.

Eric Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Irene Solaimon and Christy Dennison. 2021. Process for adapting language models to society (palms) with values-targeted datasets.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Richard H Thaler and Cass R Sunstein. 2009. *Nudge: Improving decisions about health, wealth, and happiness*. Penguin.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. Lamda: Language models for dialog applications.

Nanna Thylstrup and Zeerak Waseem. 2020. Detecting 'dirt'and 'toxicity': Rethinking content moderation as pollution behaviour. *Available at SSRN 3709719*.

Megan Ung, Jing Xu, and Y-Lan Boureau. 2021. Saferdialogues: Taking feedback gracefully after conversational safety failures. *arXiv preprint arXiv:2110.07518*.

Lucia Vaira, Mario A. Bochicchio, Matteo Conte, Francesco Margiotta Casaluci, and Antonio Melpignano. 2018. Mamabot: a system based on ML and NLP for supporting women and families during pregnancy. In *Proceedings of the 22nd International Database Engineering & Applications Symposium, IDEAS 2018, Villa San Giovanni, Italy, June 18-20, 2018*, pages 273–277. ACM.

Ibo Van de Poel. 2013. Translating values into design requirements. In *Philosophy and engineering: Reflections on practice, principles and process*, pages 253–266. Springer.

Ibo van de Poel. 2018. Design for value change. *Ethics and Information Technology*, pages 1–5.

Manuel Velasquez, Claire Andre, Thomas Shanks, and Michael J Meyer. 2015. Thinking ethically. *Issues in Ethics,(August)*, pages 2–5.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *Proceedings of the 31st International Conference on Machine Learning, Deep Learning Workshop*, Lille, France.

Kunze Wang, Dong Lu, Caren Han, Siqu Long, and Josiah Poon. 2020. Detect all abuse! toward universal abusive language detection models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6366–6376, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from Language Models. *arXiv e-prints*, page arXiv:2112.04359.

Mark West, Rebecca Kraut, and Han Ei Chew. 2019. I'd blush if i could: closing gender divides in digital skills through education. Technical Report GEN/2019/EQUALS/1 REV, UNESCO.

Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. Demoting racial bias in hate speech detection. *arXiv preprint arXiv:2005.12246*.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2950–2968, Online. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

Yangjun Zhang, Pengjie Ren, and Maarten de Rijke. 2020. Detecting and classifying malevolent dialogue responses: Taxonomy, data and methodology. *arXiv preprint arXiv:2008.09706*.

Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah Smith. 2021. Challenges in automated debiasing for toxic language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3143–3155, Online. Association for Computational Linguistics.

Controllable User Dialogue Act Augmentation for Dialogue State Tracking

Chun-Mao Lai* **Ming-Hao Hsu*** **Chao-Wei Huang** **Yun-Nung Chen**
National Taiwan University, Taipei, Taiwan
{b09901186, b09502138}@ntu.edu.tw
f07922069@csie.ntu.edu.tw y.v.chen@ieee.org

Abstract

Prior work has demonstrated that data augmentation is useful for improving dialogue state tracking. However, there are many types of user utterances, while the prior method only considered the simplest one for augmentation, raising the concern about poor generalization capability. In order to better cover diverse dialogue acts and control the generation quality, this paper proposes controllable user dialogue act augmentation (CUDA-DST) to augment user utterances with diverse behaviors. With the augmented data, different state trackers gain improvement and show better robustness, achieving the state-of-the-art performance on MultiWOZ 2.1.[1]

1 Introduction

Dialogue state tracking (DST) serves as a backbone of task-oriented dialogue systems (Chen et al., 2017), where it aims at keeping track of user intents and associated information in a conversation. The dialogue states encapsulate the required information for the subsequent dialogue components. Hence, an accurate DST module is crucial for a dialogue system to perform successful conversations.

Recently, we have seen tremendous improvement on DST, mainly due to the curation of large datasets (Budzianowski et al., 2018; Eric et al., 2020; Rastogi et al., 2020) and many advanced models. They can be broadly categorized into 3 types: span prediction, question answering, and generation-based models. The question answering models define natural language questions for each slot to query the model for the corresponding values (Gao et al., 2020; Li et al., 2021). Wu et al. (2019) proposed TRADE to perform zero-shot transfer between multiple domains via slot-value embeddings and a state generator. SimpleTOD (Hosseini-Asl et al., 2020) combines all components in a task-oriented dialogue system with a pre-trained language model. Recently, TripPy (Heck et al., 2020) categorizes value prediction into 7 types, and designs different prediction strategies for them. This paper focuses on generalized augmentation covering all categories.

Another research line leverages data augmentation techniques to improve performance (Song et al., 2021; Yin et al., 2020; Summerville et al., 2020; Kim et al., 2021). Most prior work used simple augmentation techniques such as word insertion and state value substitution. With recent advances in pre-trained language models (Devlin et al., 2019; Radford et al., 2019; Raffel et al., 2020), generation-based augmentation has been proposed (Kim et al., 2021; Li et al., 2020). These methods have demonstrated impressive improvement and zero-shot adaptability (Yoo et al., 2020; Campagna et al., 2020), while our work focuses on data augmentation with in-domain data.

The closest work is CoCo (Li et al., 2020), a framework that generates user utterances given augmented dialogue states. The examples are shown in Figure 1, where the main differences between CoCo and ours are that 1) CoCo only augments user utterances in slot and value levels, but dialogue acts and domains are fixed, making augmented data limited. Our method can augment reasonable user utterances with diverse dialogue acts and domain switching scenarios. 2) Boolean slots and referred slots are not handled by CoCo due to its higher complexity, while our approach can handle all types of values for better generalization.

This paper proposes **CUDA-DST** (**C**ontrollable **U**ser **D**ialogue **A**ct augmentation), a generalized framework of generation-based augmentation for improving DST. Our contribution is 2-fold:

- We present CUDA which generates diverse user utterances via controllable user dialogue acts augmentation.
- Our augmented data helps most DST mod-

*Equal contribution.

[1]The source code is available at https://github.com/MiuLab/CUDA-DST.

Proceedings of the SIGdial 2022 Conference, pages 53–61
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

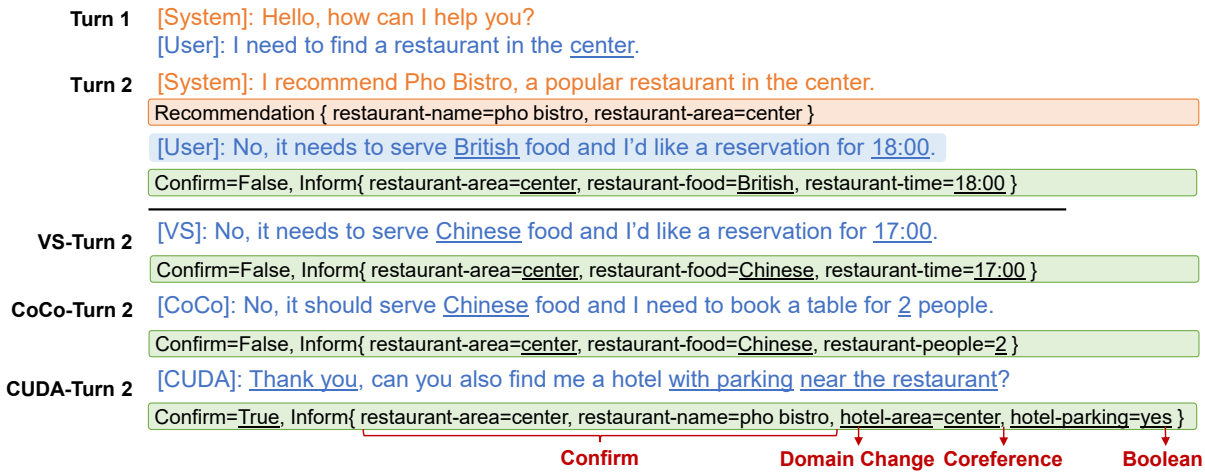


Figure 1: Augmented user utterances with the associated user dialogue acts and states from three methods.

els improve their performance. Specifically, CUDA-augmented TripPy model achieves the state-of-the-art result on MultiWOZ 2.1.

2 Controllable User Dialogue Act Augmentation (CUDA)

The goal of our method is to augment more and diverse user utterances that fit the dialogue context, and then the augmented data can help DST models learn better. More formally, given a system utterance U_t^{sys} in the turn t and dialogue history H_{t-1} before this turn, our approach focuses on augmenting a user dialogue act and state, $\hat{A}_t$, and generating the corresponding user utterance $\hat{U}_t^{\text{usr}}$. Note that each user utterance can be augmented.

To achieve this goal, we propose CUDA with three components illustrated in Figure 2: 1) a user dialogue act generation process for producing $\hat{A}_t$, 2) a user utterance generator for producing $\hat{U}_t^{\text{usr}}$, and 3) a state match filtering process.

2.1 User Dialogue Act Generation

Considering that a user dialogue act represents the core meaning of the user's behavior (Goo and Chen, 2018; Yu and Yu, 2021), we focus on simulating reasonable user dialogue acts given the system context for data augmentation. After analyzing task-oriented user utterances, user behaviors contain the following user dialogue acts:

1. **Confirm**: The system provides recommendation to the user, and the user confirms if accepting the recommended item.
2. **Reply**: The system asks for a user-desired value of the slots, and the user replies the corresponding value.
3. **Inform**: The user directly informs the desired slot values to the system.

Heck et al. (2020) designed their dialogue state tracker that tackle utterances with different dialogue acts in different ways and achieved good performance, implying that different dialogue acts contain diverse behaviors in the interactions. To augment more diverse user utterances, we introduce a random process for each user dialogue act. Unlike the prior work CoCo that did not generate utterance whose dialogue act different from the original one, our design is capable of simulating diverse behaviors for better augmentation illustrated in Figure 2.

Confirm When the system provides recommendations, our augmented user behavior has a probability of P_{confirm} to accept the recommended values. When the user confirms the recommendation, the suggested slot values are added to the augmented user dialogue state $\hat{A}_t$ as shown in Figure 1. In the example, the augmented user dialogue act is to confirm the suggested restaurant, and then includes it in the state (restaurant-name=pho bistro, restaurant-area=center).

Reply When the system requests a constraint for a specific slot, e.g. "*which area do you prefer?*", the user has a probability of P_{reply} to give the value of the requested slot. P_{reply} may not be 1, because users sometimes revise their previous requests without providing the asked information.

Inform In anytime of the conversation, the user can provide the desired slot values to convey his/her preference. As shown in the original user utterance of Figure 1, the user rejects the recommendation

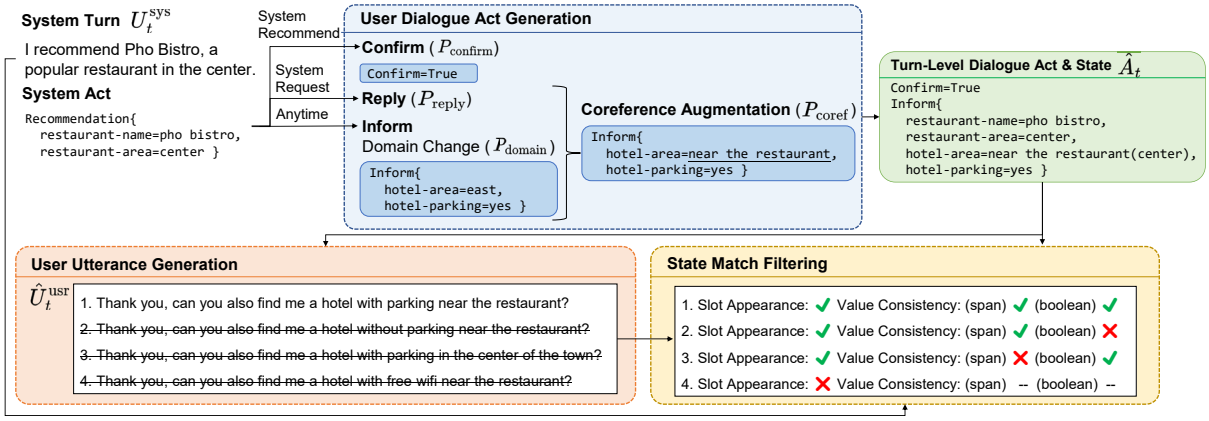


Figure 2: The overview of the proposed CUDA augmentation process.

and then directly informs the additional constraints (food and time). The number of additional informed values is randomly chosen, and then the slots and values are randomly sampled from the pre-defined ontology and dictionary. Note that the confirmed and replied information cannot be changed during additional informing. Considering that a user may change the domain within the dialogue, our algorithm allows the user to change the domain with a probability of P_{domain}, and then the informed slots and values need to be sampled from the new domain's dictionary. The new domain is selected randomly from all the other domains.

Coreference Augmentation In the generated user dialogue act and state, all informed slot values are from the pre-defined dictionary. However, it is natural for a user to refer the previously mentioned information, e.g., "*I am looking for a taxi that can arrive by the time of my reservation*". To further enhance the capability of handling coreference, our algorithm has a probability of P_{coref} to switch the slot value from the generated user dialogue state. Since not all slots can be referred, we define a coreference list containing all referable slots and the corresponding referring phrases, e.g., "*the same area as*" listed in Appendix A.

With the generated user dialogue acts and the system action, we form the corresponding turn-level dialogue act and state based on the confirmed suggestions and referred slot values as shown in the green block of Figure 2.

2.2 User Utterence Generation

To generate the user utterance associated with the augmented user dialogue act and state, we adopt a pre-trained T5 (Raffel et al., 2020) and fine-tune it on the MultiWOZ dataset by a language modeling objective formulated below:

$$\mathcal{L}_{\text{gen}} = -\sum_{k=1}^{n_t} \log p_\theta(U_{t,k}^{\text{usr}} \mid U_{t,<k}^{\text{usr}}, U_t^{\text{sys}}, H_{t-1}, A_t),$$

where $U_{t,k}^{\text{usr}}$ denotes the k-th token in the user utterance, H_{t-1} represents the all dialogue history before turn t, and A_t is the user dialogue act and state in the t-th turn. With the trained generator, we can generate the augmented user utterance by inputting the augmented user dialogue act and state $\hat{A}_t$ as shown in the green block of Figure 2. In decoding, we apply beam search so that we can augment diverse utterances for improving DST.

2.3 State Match Filtering

To make sure the generated user utterance well reflects its dialogue state, we propose two modules to check the state matching: a *slot appearance* classifier and a *value consistency* filter, where the former checks if the given slots are included and the latter focuses on ensuring the value consistency between dialogue states and user utterances.

Slot Appearance Following Li et al., we employ a BERT-based multi-label classification model to predict whether a slot appears in the given t-th turn. The augmented user utterances are eliminated if they do not contain all slots in the user dialogue state predicted by the model.

Value Consistency The slot values can be categorized into: 1) span-based, 2) boolean, and 3) *dontcare* values. It is naive to check if the span-based values are mentioned in the utterances, but boolean and *dontcare* values cannot be easily identified. To handle the slots with boolean and *dontcare* values, we propose two slot-gate classifiers

Dataset	CUDA	MultiWOZ
Span	100.00	64.61
Confirm (True)	5.27	5.84
Confirm (False)	0.44	0.32
Dontcare	0.67	2.46
Coreference	8.15	3.70
Multi-domain	13.10	24.48
#Turns	*54,855*	*69,673*

Table 1: Slot distribution in user utterances (%).

MultiWOZ	TripPy	TRADE	SimpleTOD
Original	57.72	44.08	49.19
VS	59.48	43.76	**50.50**
CoCo	60.46	43.53	50.25
CUDA	61.28†	**44.86**†	50.14
CUDA (*-coref*)	**62.93**†	42.98	49.64

Table 2: Joint goal accuracy on MultiWOZ 2.1 (%). † indicates the significant improvement over all baselines with $p < 0.05$.

motivated by Heck et al. (2020). Each boolean slot, e.g. *internet* or *parking*, is assigned to one of the classes in $C_{\text{bool}} = \{none, dontcare, yes, no\}$, while other slots are assigned to one of the classes in $C_{\text{span}} = \{none, dontcare, value\}$, where *value* indicates the span-based value. Then for all slots classified as span-based value, we check if all associated values are mentioned in the generated utterance. In addition, we use the coreference keywords, e.g., *same area*, to handle the coreference cases. We apply BERT (Devlin et al., 2019) to encode the t-th turn in a dialogue as:

$$R_t^{\text{CLS}} = \text{BERT}([\text{CLS}] \oplus U_t^{\text{sys}} \oplus [\text{SEP}] \oplus U_t^{\text{usr}} \oplus [\text{SEP}]),$$

where R_t^{CLS} denotes the output of the [CLS] token, which can be considered as the summation of the turn t. We then obtain the probability of the value types as

$$p_{s,t}^{\text{bool}} = \text{softmax}(W_s^{\text{bool}} \cdot R_t^{\text{CLS}} + b_s^{\text{bool}}) \in \mathbb{R}^4,$$

for each boolean slots, and

$$p_{s,t}^{\text{span}} = \text{softmax}(W_s^{\text{span}} \cdot R_t^{\text{CLS}} + b_s^{\text{span}}) \in \mathbb{R}^3,$$

for each span-based slots. Our multi-task BERT-based slot-gate classifier is trained with the cross entropy loss.

The neural-based filters are trained on the original MultiWOZ data, and the prediction performance in terms of slots (for both appearance and value consistency) is 92.9% in F1 evaluated on the development set. In our CUDA framework, we apply the trained filters to ensure the quality of the augmented user utterances as shown in Figure 2.

3 Experiments

To evaluate if our augmented data is beneficial for improving DST models, we perform three popular trackers, TRADE (Wu et al., 2019), SimpleTOD (Hosseini-Asl et al., 2020), and TripPy (Heck et al., 2020), on MultiWOZ 2.1 (Eric et al., 2020).

3.1 Experimental Setting

Our CUDA generator is trained on the training set of MultiWOZ 2.3 (Han et al., 2020) due to its additional *coreference* labels. Note that all dialogues are the same as MultiWOZ 2.1. We then generate the augmented dataset for the training set of MultiWOZ 2.1 for fair comparison with the prior work. The predifined slot-value dictionary is taken from CoCo's *out-of-domain* dictionary and the defined coreference list is shown in Appendix A.

In user dialogue act generation, the parameters are set as $(P_{\text{confirm}}, P_{\text{reply}}, P_{\text{domain}}, P_{\text{coref}}) = (0.7, 0.9, 0.8, 0.6)$, which can be flexibly adjusted to simulate different user behaviors. We report the distribution of slot types in our augmented data and the original MultiWOZ data in Table 1, where it can be found that our augmented slots cover diverse slot types and the distribution is reasonably similar to the original MultiWOZ. Different from the prior work, CoCo, which only tackled the span-based slots, our augmented data may better reflect the natural conversational interactions. Additionally, we perform CUDA with $P_{\text{coref}} = 0$ to check the impact of coreference augmentation.

We train three DST models on the augmented data and evaluate the results using joint goal accuracy. The compared augmentation baselines include value substitution (VS) and CoCo (Li et al., 2020) with the same setting.

3.2 Effectiveness of CUDA-Augmented Data

Table 2 shows that CUDA significantly improves TripPy and TRADE results by 3.6% and 0.8% respectively on MultiWOZ, and even outperforms the prior work CoCo. In addition, our CUDA augmentation process has 78% success rate, while CoCo only has 57%, demonstrating the efficiency of our augmentation method and the great data utility. Interestingly, CUDA without *coreference* achieves slightly better performance for TripPy while the performance of TRADE and SimpleTOD degrade,

CoCo+(rare)	TripPy	TRADE	SimpleTOD
Original	28.38	16.65	19.20
VS	39.42	16.42	26.26
CUDA	**48.83**	**17.79**	**29.32**
CUDA (*-coref*)	48.67	16.80	28.66
CoCo	*56.50*	*18.01*	*30.60*

Table 3: Joint goal accuracy on CoCo+ (rare) (%).

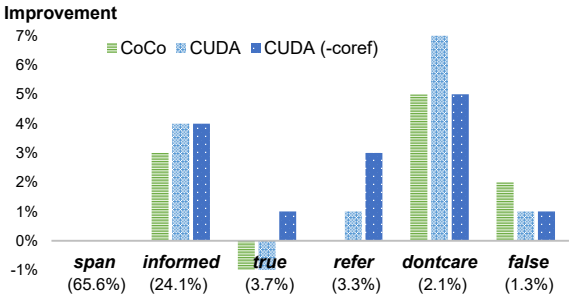


Figure 3: Performance gain across slots on TripPy.

achieving the new state-of-the-art performance on MultiWOZ 2.1. The probable reason is that TripPy already handles coreference very well via its refer classification module, so augmenting coreference cases may not help it a lot. In contrast, other generative models (TRADE and SimpleTOD) can benefit more from our augmented coreference cases. Another reason may be the small distribution of coreference slots in MultiWOZ shown as Table 1, implying that augmented data with too many coreference slots does not align well with the original distribution and hurts the performance.

3.3 Robustness to Rare Cases

We also evaluate our models on *CoCo+ (rare)*[2], a test set generated by CoCo's algorithm (Li et al., 2020), to examine model robustness under rare scenarios. Table 3 presents the results on CoCo+ (rare), which focuses rare cases for validating the model's robustness. It is clear that the model trained on our augmented data shows better generalization compared with the one trained on the original MultiWOZ data, demonstrating the effectiveness on improving robustness of DST models. The performance of CoCo is listed as reference, because comparing with its self-generated data is unfair.

3.4 Slot Performance Analysis

To further investigate the efficacy for each slot type, Figure 3 presents its performance gain on TripPy. Comparing with CoCo, CUDA improves more on *informed*, *refer*, and *dontcare* slots. It implies that CUDA augments diverse user dialogue acts for helping *informed* and *refer*, and the proposed slot-gate can better ensure value consistency for improving *dontcare* slots, even though they are rare cases in MultiWOZ. Our model can also keep the same performance for frequent *span* slots, demonstrating great generalization capability across diverse slot types from our controllable augmentation. The qualitative study can be found in Appendix B.

[2]CoCo+ (rare) applies *CoCo* and *value substitution (VS)* with a *rare* slot-combination dictionary.

4 Conclusion

We introduce a generalized data augmentation method for DST by utterance generation with controllable user dialogue act augmentation. Experiments show that our approach improves results of multiple state trackers and achieves state-of-the-art performance on MultiWOZ 2.1. Further study demonstrates that trackers' robustness and generalization capabilities can be improved by diverse generation covering different user behaviors.

Acknowledgements

We thank reviewers for their insightful comments. This work was financially supported from the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grants 111-2628-E-002-016 and 111-2634-F-002-014.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. MultiWOZ-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yun-Nung Chen, Asli Celikyilmaz, and Dilek Hakkani-Tur. 2017. Deep learning for dialogue systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 8–14.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of

deep bidirectional transformers for language understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. ACL.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *the 12th Language Resources and Evaluation Conference*, pages 422–428. European Language Resources Association.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From machine reading comprehension to dialogue state tracking: Bridging the gap. In *the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89. ACL.

Chih-Wen Goo and Yun-Nung Chen. 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 735–742. IEEE.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Wei Peng, and Minlie Huang. 2020. MultiWOZ 2.3: A multi-domain task-oriented dataset enhanced with annotation corrections and co-reference annotation. *arXiv preprint arXiv:2010.05594*.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44. ACL.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. 2021. NeuralWOZ: Learning to collect task-oriented dialogue via model-based simulation. In *the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3704–3717. ACL.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2020. CoCo: Controllable counterfactuals for evaluating dialogue state trackers. In *International Conference on Learning Representations*.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074. ACL.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Xiaohui Song, Liangjun Zang, and Songlin Hu. 2021. Data augmentation for copy-mechanism in dialogue state tracking. In *International Conference on Computational Science*, pages 736–749. Springer.

Adam Summerville, Jordan Hashemi, James Ryan, and William Ferguson. 2020. How to tame your data: Data augmentation for dialog state tracking. In *the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 32–37. ACL.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819. ACL.

Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dialog state tracking with reinforced data augmentation. In *the AAAI Conference on Artificial Intelligence*, volume 34, pages 9474–9481.

Kang Min Yoo, Hanbit Lee, Franck Dernoncourt, Trung Bui, Walter Chang, and Sang-goo Lee. 2020. Variational hierarchical dialog autoencoder for dialog state tracking data augmentation. In *the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3406–3425. ACL.

Dian Yu and Zhou Yu. 2021. Midas: A dialog act annotation scheme for open domain humanmachine spoken conversations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1103–1120.

A Reproducibility

Our CUDA generator is trained on the training set of MultiWOZ 2.3 (Han et al., 2020) due to its additional *coreference* labels. Note that all dialogues are the same as MultiWOZ 2.1. We then generate the augmented dataset using CUDA for the training set of MultiWOZ 2.1 for fair comparison with the prior work. The predifined slot-value dictionary is taken from CoCo's *out-of-domain* dictionary shown in Table 4 and the defined coreference list is shown in Table 5.

B Qualitative Study

The augmented data samples are shown in Figure 4. It can be found that the augmented user utterances can fluently switch the domain and include associated slot values that are aligned well with the dialogue states.

Slot Name	Possible Values
hotel-internet†	['yes', 'no', 'dontcare']
hotel-type	['hotel', 'guesthouse']
hotel-parking†	['yes', 'no', 'dontcare']
hotel-price	['moderate', 'cheap', 'expensive']
hotel-day	['march 11th', 'march 12th', 'march 13th', 'march 14th', 'march 15th', 'march 16th', 'march 17th', 'march 18th', 'march 19th', 'march 20th']
hotel-people	['20', '21', '22', '23', '24', '25', '26', '27', '28', '29']
hotel-stay	['20', '21', '22', '23', '24', '25', '26', '27', '28', '29']
hotel-area	['south', 'north', 'west', 'east', 'centre', 'dontcare']
hotel-stars	['0', '1', '2', '3', '4', '5', 'dontcare']
hotel-name	['moody moon', 'four seasons hotel', 'knights inn', 'travelodge', 'jack summer inn', 'paradise point resort']
restaurant-area	['south', 'north', 'west', 'east', 'centre', 'dontcare']
restaurant-food	['asian fusion', 'burger', 'pasta', 'ramen', 'taiwanese', 'dontcare']
restaurant-price	['moderate', 'cheap', 'expensive', 'dontcare']
restaurant-name	['buddha bowls', 'pizza my heart', 'pho bistro', 'sushiya express', 'rockfire grill', 'itsuki restaurant']
restaurant-day	['monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday', 'sunday']
restaurant-people	['20', '21', '22', '23', '24', '25', '26', '27', '28', '29']
restaurant-time	['19:01', '18:06', '17:11', '19:16', '18:21', '17:26', '19:31', '18:36', '17:41', '19:46', '18:51', '17:56', '7:00 pm', '6:07 pm', '5:12 pm', '7:17 pm', '6:17 pm', '5:27 pm', '7:32 pm', '6:37 pm', '5:42 pm', '7:47 pm', '6:52 pm', '5:57 pm', '11:00 am', '11:05 am', '11:10 am', '11:15 am', '11:20 am', '11:25 am', '11:30 am', '11:35 am', '11:40 am', '11:45 am', '11:50 am', '11:55 am']
restaurant-food	['asian fusion', 'burger', 'pasta', 'ramen', 'taiwanese', 'dontcare']
taxi-arrive	['17:26', '19:31', '18:36', '17:41', '19:46', '18:51', '17:56', '7:00 pm', '6:07 pm', '5:12 pm', '7:17 pm', '6:17 pm', '5:27 pm', '11:30 am', '11:35 am', '11:40 am', '11:45 am', '11:50 am', '11:55 am']
taxi-leave	['19:01', '18:06', '17:11', '19:16', '18:21', '7:32 pm', '6:37 pm', '5:42 pm', '7:47 pm', '6:52 pm', '5:57 pm', '11:00 am', '11:05 am', '11:10 am', '11:15 am', '11:20 am', '11:25 am']
taxi-depart	['moody moon', 'four seasons hotel', 'knights inn', 'travelodge', 'jack summer inn', 'paradise point resort']
taxi-dest	['buddha bowls', 'pizza my heart', 'pho bistro', 'sushiya express', 'rockfire grill', 'itsuki restaurant']
train-arrive	['17:26', '19:31', '18:36', '17:41', '19:46', '18:51', '17:56', '7:00 pm', '6:07 pm', '5:12 pm', '7:17 pm', '6:17 pm', '5:27 pm', '11:30 am', '11:35 am', '11:40 am', '11:45 am', '11:50 am', '11:55 am']
train-leave	['19:01', '18:06', '17:11', '19:16', '18:21', '7:32 pm', '6:37 pm', '5:42 pm', '7:47 pm', '6:52 pm', '5:57 pm', '11:00 am', '11:05 am', '11:10 am', '11:15 am', '11:20 am', '11:25 am']
train-depart	['gilroy', 'san martin', 'morgan hill', 'blossom hill', 'college park', 'santa clara', 'lawrence', 'sunnyvale']
train-dest	['mountain view', 'san antonio', 'palo alto', 'menlo park', 'hayward park', 'san mateo', 'broadway', 'san bruno']
train-day	['march 11th', 'march 12th', 'march 13th', 'march 14th', 'march 15th', 'march 16th', 'march 17th', 'march 18th', 'march 19th', 'march 20th']
train-people	['20', '21', '22', '23', '24', '25', '26', '27', '28', '29']
attraction-area	['south', 'north', 'west', 'east', 'centre', 'dontcare']
attraction-name	['grand canyon', 'golden gate bridge', 'niagara falls', 'kennedy space center', 'pike place market', 'las vegas strip']
attraction-type	['historical landmark', 'aquaria', 'beach', 'castle', 'art gallery', 'dontcare']

Table 4: The pre-defined slot-value dictionary, where † indicates a binary slot.

Slot Name	Referred Slot Name	Referred Key Value
hotel-price	*restaurant-price*	['same', 'same price', 'same price range']
hotel-day	*train-day*	['same', 'same day']
	restaurant-day	['same', 'same day']
hotel-people	*train-people*	['same', 'same group', 'same party']
	restaurant-people	['same', 'same group', 'same party']
hotel-area	*restaurant-area*	['same', 'same area', 'same part', 'near the restaurant']
	attraction-area	['same', 'same area', 'same part', 'near the attraction']
restaurant-area	*hotel-area*	['same', 'same area', 'same part', 'near the hotel']
	attraction-area	['same', 'same area', 'same part', 'near the attraction']
restaurant-price	*hotel-price*	['same', 'same price', 'same price range']
restaurant-day	*train-day*	['same', 'same day']
	hotel-day	['same', 'same day']
restaurant-people	*train-people*	['same', 'same group', 'same party']
	hotel-people	['same', 'same group', 'same party']
taxi-depart	*hotel-name*	['the hotel']
	restaurant-name	['the restaurant']
	attraction-name	['the attraction']
taxi-dest	*hotel-name*	['the hotel']
	restaurant-name	['the restaurant']
	attraction-name	['the attraction']
taxi-arrive	*restaurant-time*	['the time of my reservation', 'the time of my booking']
train-day	*restaurant-day*	['same', 'same day']
	hotel-day	['same', 'same day']
train-people	*restaurant-people*	['same', 'same group', 'same party']
	hotel-people	['same', 'same group', 'same party']
attraction-area	*hotel-area*	['same', 'same area', 'same part', 'near the hotel']
	restaurant-area	['same', 'same area', 'same part', 'near the restaurant']

Table 5: The coreference list. The slots that is not referable will not be displayed in the above table.

	Dialogues	CUDA Turn label
Example 1 Confirm Coref Domain	[sys]:Byard Art is near where you will be eating. The entrance fee is free. [cuda]:Thank you. Can you help me find a train that leaves after 19:16 on the same day as the restaurant?	attraction-name: Byard Art train-leaveat: 19:16 train-day: monday
Example 2 Confirm Domain	[sys]:I suggest the broughton house gallery. Is there anything else I could help you with ? [cuda]:Yes, I'm looking for an expensive restaurant. I 'd like to make a reservation for Saturday.	attraction-name: the broughton house gallery restaurant-book day: saturday restaurant-pricerange: expensive
Example 3 Reply Binary	[sys]:There are five hotels that meet your requirements. Is there a certain part of town you are looking in ? [cuda]:Yes, I 'd like to stay in the east, and I need it to have free wifi. I 'll need it for 24 people.	hotel-area: east hotel-internet: yes hotel-book people: 24
Example 4 Dontcare	[sys]: No. I am sorry. there isn't. something else perhaps? [cuda]:How about an art gallery? I don't care what area it is in.	attraction-type: art gallery attraction-area: dontcare
Example 5 Confirm Coref Domain	[sys]: Okay , we have the cambridge university botanic gardens in the centre of town . Will that work for you ? [cuda]:Yes, I need a taxi to get me to itsuki restaurant by the time of my reservation.	attraction-area: centre attraction-name: cambridge university botanic gardens taxi-destination: itsuki restaurant taxi-arriveby: 15:45

Figure 4: The CUDA-generated examples. The red tags indicate the strategies implemented by CUDA.

Developing an argument annotation scheme based on a semantic classification of arguments

Lea Kawaletz, Heidrun Dorgeloh, Stefan Conrad and **Zeljko Bekcic**
Heinrich-Heine-University Düsseldorf, Germany
{lea.kawaletz, dorgeloh, stefan.conrad,zeljko.bekcic}@hhu.de

Abstract

Corpora of argumentative discourse are commonly analyzed in terms of argumentative units, consisting of claims and premises. Both argument detection and classification are complex discourse processing tasks. Our paper introduces a semantic classification of arguments that can help to facilitate argument detection. We report on our experiences with corpus annotations using a function-based classification of arguments and a procedure for operationalizing the scheme by using semantic templates.

1 Introduction

The corpus-based analysis of argumentative texts is a widely used discourse processing task needed both for an in-depth understanding of this basic discourse type, and in the field of argument mining. We here present an annotation scheme that has been developed as part of a project for gaining detailed insight into the linguistic features of arguments. These features can be used for machine learning as well as for the task of argument detection in the study of discourse and discourse processing.

In contrast to other approaches in the field, our method aims at the identification and classification of arguments, and not at the analysis of an overall argumentation structure (cf., for example, Peldszus et al. 2016). We argue that the annotation scheme will facilitate the annotation process in many applications of argument detection, enabling both researchers and annotators to zoom into linguistic characteristics that pertain to a specific class of arguments rather than to the notion of 'argument' as a whole. The approach therefore reduces some of the vagueness of the category of 'argument' and adds to the transparency of annotators' decisions.

Arguments are used for different purposes, aiming to persuade an addressee to believe, evaluate, or do something (see e.g. Eggs 2008; Stede and Schneider 2019). We use this functional versatility of arguments as a starting point for our annotation scheme. More precisely, we propose a systematic testing procedure during which annotators use a set of linguistic templates on a given text passage to determine whether it is an argument or not, and, if so, which argumentative function it has. We are currently developing and evaluating this approach with a corpus of COVID-19-related news opinion texts from *The New York Times*.

This paper is structured as follows. In section 2, we introduce the general idea of a function-based argument classification and briefly describe our corpus. In section 3, we present and evaluate our initial, rather ad hoc annotation efforts. In section 4, we introduce our function-based annotation scheme and report on our progress in terms of workflow and inter-annotator agreements. In section 5, we summarize our insights and provide an outlook.

2 Background

2.1 Arguments and argument categories

Theories of discourse generally claim that arguments do not have a particular linguistic form, but appear in all sorts of linguistic structures (e.g., Smith 2003; Virtanen 2010; Dorgeloh and Wanner 2010). Accordingly, the annotation of arguments in corpora is still a challenge because "a substantial amount of knowledge needed for the correct recognition of the argumentation, its composing elements and their relationships is not explicitly present in the text" (Moens 2018, 1; see also Lawrence and Reed 2020). Resulting from this difficulty, argument detection schemes so far often avoid cross-topic transfer (e.g., Nguyen and Litman 2015; Liebeck et al. 2016), but schemes for more heterogeneous corpora also exist (e.g., Stab et al. 2018; Cabrio and Villata 2018; Ein-Dor et al. 2020). Such work from argument mining typically relies on recurrent patterns identified by the NLP model used, but does not imply a systematic, truly topic-independent classification of arguments.

Proceedings of the SIGdial 2022 Conference, pages 62–67
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Argumentative discourse is characterized by presenting a central, disputed issue, the *major claim*, which the author argues for or against (Stab and Gurevych 2017). That is, they aim to persuade an addressee to believe and/or evaluate and/or do something, and they provide a number of arguments to this end (van Eemeren and Grootendorst 2004; Stede and Schneider 2019). This variability of what an argument is ultimately intended to do is often commented on in existing approaches, for example as an argument being either the expression of (positive or negative) stance, or of a policy or action to be taken (e.g., Hidey et al. 2017; Ein-Dor et al. 2020).

We suggest that this functional complexity of argumentation is exactly what is needed for the aim of developing a topic-independent classification scheme that can be applied to arguments as a whole. In the annotation scheme we developed, we distinguish between *epistemic*, *ethical* and *deontic* arguments, as first proposed by Eggs (2008; see Stede and Schneider 2019 for a summary in English). The three types are illustrated in Table 1.

Table 1: Argument categories

polarity	epistemic	ethical	deontic
positive	x is true	x is good	do x
negative	x is false	x is bad	don't do x

In addition to an argument being understood by its function, the most common definition is that it has two components, the *claim* and the *premise*. The claim is typically described as a controversial statement which provides the topic of the argument, and its premise is then a statement which provides evidence or expresses reasoning that either supports or attacks the claim (Stab et al. 2018). The link between a claim and its premise can thus be conceptualized as a directed argumentative relation, with a premise as the source and a (major) claim as its target (Stab and Gurevych 2014b). Each argument classified by our annotation scheme needs to have these two components expressed in the text.

2.2 Corpus compilation

Our corpus is currently being developed at Heinrich-Heine-University Düsseldorf ('HHU') as part of a collaborative project of both linguists and computer scientists working on argumentative discourse. So far, it consists of 25 COVID-19-related news opinion texts from *The New York Times* (29,466 words), and it will be consecutively expanded as the annotations progress. The corpus is designed to provide us with an inventory of arguments, divided into components and categorized by function and polarity. This inventory will first be used for linguistic analysis and, at a later stage, for an experiment with human subjects on argument-specific discourse relations, as well as for machine learning experiments.

3 The initial annotation process

Our first set of annotations ('set 1') was created before the introduction of our annotation scheme. Four annotators were instructed to apply a basic, simplified notion of 'argument,' consisting of a claim that is either supported or attacked. Practical issues of claim detection and annotation (e.g. size of the discourse unit, treatment of quotes within the texts) were discussed at regular meetings, leading the group from an initial, very thorough exemplary discussion of three texts (subset 1-1, 3,653 words) to an annotation of another ten texts in one hit (subset 1-2, 11,646 words). Annotations were created in the INCEpTION tool (Klie et al. 2018), hosted on a HHU server.

As the annotation task is not only a coding but also a unitizing task, we measure the inter-annotator agreement using Krippendorff's unitizing alpha (Krippendorff et al., 2016). This measure works with an arbitrary number of annotators (where not all have to annotate all texts) and determines the degree of observed disagreement in relation to the expected disagreement (assuming random annotations). Values range from -1 to 1, with values around zero representing random annotations, positive values representing more agreement among the annotators, and negative values representing more disagreement than expected by chance. The results for both subsets are displayed in Table 2.[1] We counted whether the annotators identified a given text passage as a *premise,* as a *claim,* or not as an argument component at all.

While subset 1-1 showed promising inter-annotator agreement scores, subset 1-2 comes with disappointing scores. The good values for subset 1-1 are likely the result of the initial, intensive discussion between and with the annotators, producing biased annotations. Comparing this to the weaker values for subset 1-2, it seems obvious that the an-

[1]The ID numbering starts at 10 because the very first annotations did not turn out to be suitable for our purposes, which is why the first nine texts were excluded from the corpus.

Table 2: Inter-annotator agreement ('iaa') of set 1 for annotating *premise* vs. *claim* vs. nothing, by text (Krippendorff's unitizing alpha, Krippendorff et al. 2016)

subset	id	iaa	# of annotators
1-1	10	0.2713	3
1-1	11	0.4078	3
1-1	12	0.2646	3
1-2	13	0.1932	3
1-2	14	-0.0268	3
1-2	15	0.3851	3
1-2	16	0.3002	3
1-2	17	0.0123	3
1-2	18	0.1705	3
1-2	19	0.0941	3
1-2	20	0.3853	3
1-2	21	0.1891	3
1-2	22	0.0681	3

notators need more precise guidelines than what was provided in this second annotation round. This is supported by the fact that introducing a systematic annotation scheme has also been shown to improve inter-annotator agreement in previous projects involving argument annotation (see Stab and Gurevych 2014a). Therefore, our logical next step was to introduce such a scheme, as described in the next section.

4 Introducing an annotation scheme

Our updated annotation process is divided into three major steps (see the similar approaches in e.g. Stab and Gurevych 2014a; Peldszus et al. 2016):

1. Identify the major claim: The annotator reads the full text in order to understand the overall argumentation, and annotates or formulates the major claim.
2. Identify claims and premises: The annotator identifies claims and premises according to a set of criteria, and labels them by semantic category.
3. Review and submit: The annotator goes through the whole text again to finalize their annotation, and submits their annotated text.

We here focus on step two, the identification of claims and premises. Specifically, we describe the approach we apply to identify arguments by systematically categorizing them semantically. For further information on steps one and three see our annotation guidelines (Kawaletz et al. in prep).

In order for a pair of text passages to be included in our database as an argument, it must meet the following criteria:

1. x is a controversial statement (the claim)
2. x is supported or attacked by y (the premise)
3. x supports, attacks or repeats the major claim
4. x is an epistemic, ethical or deontic claim

The first two criteria represent the standard definition of claim and premise (see above), while the third one guarantees that our resulting database has a homogeneous subject matter (in order to facilitate future experiments involving cross-topic transfer). The final criterion, which distinguishes our approach from other, existing ones, is the obligatory assignment of the claim to one of three semantic categories.

In order to test a pair of text passages for these criteria, annotators insert them into linguistic templates (see Kawaletz et al. in prep for details). For the final, semantic criterion, these templates take the form 'x, [___] y' as presented in Table 3. These templates make use of the connectors *and* (for support relations) and *but* (for attack relations), of sentential negation (e.g. *not true* negating *true*), lexical negation (e.g. *false* negating *true*), lexical cues (e.g. *approve/disapprove* for ethical claims), and indication of stress by means of italics to increase grammatical acceptability. All templates may be adapted by the annotator to fit a given syntactic context.

The application of these templates is exemplified in (1). There, we see a claim (bold print) and a premise (underlined) from our corpus, inserted in the template which tests for a supported, positive, deontic claim (represented in Table 3 by *and do this because*). By inserting the two text passages into this template, both the argumentative function and the relation between claim and premise are made explicit.

(1) a. **[M]asking should be mandated and enforced.**

b. *And this should be done because* [i]t's not just about your individual risk tolerance, but about keeping everyone safe.

By systematically applying such templates, our annotation process is now based on principled linguistic judgments rather than on ad hoc decisions. At the point of writing this paper, we have applied our annotation scheme to 12 texts (14,167 words), with promising results: Annotators have reported that applying the provided patterns and being obliged to think about a given text passage in functional terms facilitates argument identification from the start. Thus, by specifying the in-

Table 3: Templates testing for claim categories

claim category	positive claim	negative claim
epistemic		
support	and this is true because and this is the case because	and this is false because and this is not the case because
attack	but this is not true because but this is not the case because	but this is not false because but this *is* the case because
ethical		
support	and this is good because and I find this good because and I approve because and what is good about this is	and this is bad because and I find this bad because and I disapprove because and what is bad about this is
attack	but this is bad because but what is bad about this is	but this is good because but what is good about this is
deontic		
support	and do this because	and don't do this because
attack	but don't do this because	but *do* do this because

ternal, semantic structure of the category *claim* thoroughly, its separation from premises as well as non-argument units becomes clearer. Furthermore, discussions about the status of text passages as argumentative discourse units go more smoothly.

These impressions are backed up by a clear trend toward increasing inter-annotator agreements, as illustrated in Figure 1. In set 2, annotators reached an agreement of up to a rounded 0.6 (as compared to 0.4 for set 1), with no negative values. However, this difference does not come out as significant, as is shown by an unpaired t-test comparing set 1 (M = 0.208831, SD = 0.143289) and set 2 (M = 0.315300, SD = 0.190852); t(23) = 1.5857, p = 0.1265. The fact that we have not been able to support our intuition statistically is likely due to the small sample size and is currently being tested on more texts as the project progresses.

Introducing the argument categories has not only had beneficial effects, however. The annotators have also reported that actually deciding on one functional label is often difficult, due to ambiguities in the text. Interestingly, this sentiment is not reflected in the inter-annotator agreements for set 2: As shown in Table 4, for any given text the difference between the more basic decision (*premise* vs. *claim* vs. nothing) and the more complex decision on a specific claim label (*premise* vs. *epistemic claim* vs. *ethical claim* vs. *deontic claim* vs. nothing) is negligible. A paired t-test reveals that there is indeed no significant difference between the two (p/c/Ø: M = 0.315300, SD = 0.190852; ep/et/d/Ø: M = 0.316033, SD = 0.188318; t(11) = 0.2020, p = 0.8436).

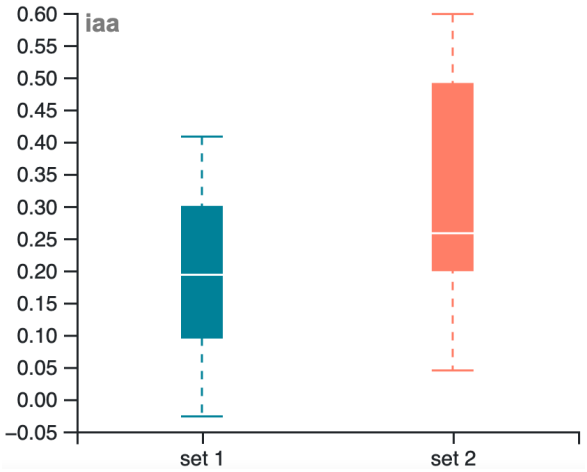


Figure 1: Inter-annotator agreement ('iaa') of sets 1 and 2 for annotating *premise* vs. *claim* vs. nothing (Krippendorff's unitizing alpha, Krippendorff et al. 2016)

Table 4: Inter-annotator agreement ('iaa') of set 2 by text (Krippendorff's unitizing alpha, Krippendorff et al. 2016), comparing *premise* vs. *claim* vs. nothing ('p/c/Ø') and *premise* vs. *epistemic claim* vs. *ethical claim* vs. *deontic claim* vs. nothing ('p/ep/et/d/Ø').

subset	id	iaa (p/c/Ø)	iaa (p/ep/et/d/Ø)	# of annotators
2-1	23	0.1811	0.181	4
2-1	24	0.2809	0.2657	4
2-2	25	0.0951	0.113	3
2-2	26	0.2516	0.2798	3
2-2	27	0.5906	0.5953	3
2-2	28	0.2496	0.2391	3
2-2	29	0.0446	0.0428	3
2-2	30	0.2638	0.2623	3
2-2	31	0.5531	0.5525	3
2-2	32	0.2046	0.2027	3
2-2	33	0.5982	0.5829	3
2-2	34	0.4704	0.4753	3

Apart from further improvements in inter-annotator agreement, we presume that applying a semantic classification of arguments is likely to reduce false positives in manual annotation as well. The text passage in (2), for example, was wrongly classified as an argument by one of three annotators during initial annotation. Applying the templates from Table 3, however, shows that the passage does not fit in either of the twelve categories, as exemplified in (2′) with the pattern *and this is true because* for the category *positive, epistemic, supported claim*.

(2) a. The U.S. Supreme Court threatens to get into the action, too.

b. In May, four conservative justices [...] dissented from an order in South Bay United Pentecostal Church v. Newsom allowing California's COVID-19-related restrictions to remain in place for gatherings at places of worship.

(2′) a. The U.S. Supreme Court threatens to get into the action, too.

b. # *And this is true because*, [i]n May, four conservative justices [...] dissented from an order in South Bay United Pentecostal Church v. Newsom allowing California's COVID-19-related restrictions to remain in place for gatherings at places of worship.

In this example, (2a) is a controversial statement and thus a valid candidate for a claim, but (2b) does not support (nor attack) it. Rather, it specifies more exactly what happened, as can be shown by applying another one of our templates, namely 'X. *What happened is that* y.':

(2″) a. The U.S. Supreme Court threatens to get into the action, too.

b. *What happened is that*, [i]n May, four conservative justices [...] dissented from an order in South Bay United Pentecostal Church v. Newsom allowing California's COVID-19-related restrictions to remain in place for gatherings at places of worship.

As these examples, contrasting with (1) above, illustrate, the point of the semantic classification and of the corresponding paraphrases is to enable annotators in the early stage of argument detection to make informed, well-founded decisions. Previous work with semantic types left the initial argument detection to experts and applied a semantic classification separately (see Hidey et al. 2017), while our approach aims at an improved identification of arguments, which then become available for thorough linguistic investigation.

5 Conclusion and outlook

In this paper, we have sketched an annotation scheme which builds on a function-based classification of arguments. By systematically applying an array of linguistic templates to pairs of text passages, the annotation process is streamlined and facilitated. A first trend for improved inter-annotator agreements, however, has yet to be statistically confirmed. In the long run, we expect significant improvements in annotator recall as well as a less labor-intensive creation of a gold standard (i.e., the curation of the annotated texts by an expert linguist annotator).

In order to further improve our results in terms of inter-annotator agreement and annotator recall, we are currently refining our work flow: For the third set of annotations, we have restricted our corpus to editorials, a more homogeneous subgenre of newspaper opinion pieces, and we are limiting text length to between 40 and 70 sentences in order to avoid too much variation in how the texts deal with argumentation in general. In addition, all annotators are actively involved in the text selection process, pre-assessing and potentially rejecting each text according to a growing catalogue of criteria (e.g. too anecdotal, too many direct quotes).

In the future, apart from the methodological benefits of applying a semantically-grounded annotation scheme, ultimately we will also be able to investigate the semantic types per se. Possible research questions are, for example, which linguistic features annotators and/or machines use to categorize arguments, and how our classification scheme relates to others (e.g. Hidey et al. 2017 on interpretation, evaluation, and agreement/disagreement).

References

Elena Cabrio and Serena Villata. 2018. Five years of argument mining: A data-driven analysis. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5427–5433. IJCAI Organization.

Heidrun Dorgeloh and Anja Wanner, editors. 2010. *Syntactic Variation and Genre*. Number 70 in Topics in

English Linguistics. De Gruyter Mouton, Berlin/New York.

Ekkehard Eggs. 2008. 39. Vertextungsmuster Argumentation: Logische Grundlagen. In Klaus Brinker, Gerd Antos, Wolfgang Heinemann, and Sven F. Sager, editors, *Text- und Gesprächslinguistik 1. Halbband*, volume 16/1 of *Handbücher zur Sprach- und Kommunikationswissenschaft*, pages 397–414. De Gruyter Mouton, Berlin/Boston.

Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. 2020. Corpus wide argument mining: A working solution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7683–7691.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. 2017. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark. Association for Computational Linguistics.

Lea Kawaletz, Heidrun Dorgeloh, Stefan Conrad, and Zeljko Bekcic. in prep. Annotation guidelines for the project 'Probing patterns of argumentative discourse'. Unpublished Manuscript.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The INCEpTION platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics.

Klaus Krippendorff, Yann Mathet, Stéphane Bouvry, and Antoine Widlöcher. 2016. On the reliability of unitizing textual continua: Further developments. *Quality & Quantity: International Journal of Methodology*, 50(6):2347–2364.

John Lawrence and Chris Reed. 2020. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Matthias Liebeck, Katharina Esau, and Stefan Conrad. 2016. What to do with an airport? mining arguments in the German online participation project tempelhofer feld. In *Proceedings of the 3rd Workshop on Argument Mining (54th Annual Meeting of the ACL), ArgMining@ACL, Berlin*, Berlin, Germany. Association for Computational Linguistics.

Marie-Francine Moens. 2018. Argumentation mining: How can a machine acquire common sense and world knowledge? *Argument & Computation*, 9(1):1–14.

Huy Nguyen and Diane Litman. 2015. Extracting argument and domain words for identifying argument components in texts. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 22–28, Denver, CO. Association for Computational Linguistics.

Andreas Peldszus, Saskia Warzecha, and Manfred Stede. 2016. Annotation guidelines for argumentation structure. English translation of chapter "Argumentationsstruktur" in Manfred Stede (ed.): Handbuch Textannotation – Potsdamer Kommentarkorpus 2.0. Universitätsverlag Potsdam, 2016.

Carlota S. Smith. 2003. *Modes of Discourse: The Local Structure of Texts*. Cambridge University Press, Cambridge.

Christian Stab and Iryna Gurevych. 2014a. Annotating argument components and relations in persuasive essays. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 1501–1510, Dublin. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014b. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56, Doha, Qatar. Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Christian Stab, Tristan Miller, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources using attention-based neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Manfred Stede and Jodi Schneider. 2019. *Argumentation Mining*. Number 40 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

Frans H. van Eemeren and Rob Grootendorst. 2004. *A systematic theory of argumentation: the pragma-dialectical approach*. Cambridge University Press, New York.

Tuija Virtanen. 2010. Variation across texts and discourses: Theoretical and methodological perspectives on text type and genre. In Heidrun Dorgeloh and Anja Wanner, editors, *Syntactic Variation and Genre*, volume 70, pages 53–84. De Gruyter Mouton, Berlin/New York.

Multi-Task Learning for Depression Detection in Dialogs

Chuyuan Li[1], Chloé Braud[2], Maxime Amblard[1]
[1] Universite de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2] IRIT, Université de Toulouse, CNRS, ANITI, Toulouse, France
[1] {firstname.name}@loria.fr, [2] chloe.braud@irit.fr

Abstract

Depression is a serious mental illness that impacts the way people communicate, especially through their emotions, and, allegedly, the way they interact with others. This work examines depression signals in dialogs, a less studied setting that suffers from data sparsity. We hypothesize that depression and emotion can inform each other, and we propose to explore the influence of dialog structure through topic and dialog act prediction. We investigate a Multi-Task Learning (MTL) approach, where all tasks mentioned above are learned jointly with dialog-tailored hierarchical modeling. We experiment on the DAIC and DailyDialog corpora – both contain dialogs in English – and show important improvements over state-of-the-art on depression detection (at best 70.6% F_1), which demonstrates the correlation of depression with emotion and dialog organization and the power of MTL to leverage information from different sources.

1 Introduction

Depression is a serious mental disorder that affects around 5% of adults worldwide.[1] It comes with multiple causes and symptoms, leading to major disability, but is often hard to diagnose, with about half the cases not detected by primary care physicians (Cepoiu et al., 2008). Automated detection of depression, sometimes associated to other mental health disorders, has been the topic of several studies recently, with a particular focus on social media data and online forums (Coppersmith et al., 2015; Benton et al., 2017; Guntuku et al., 2017; Yates et al., 2017; Song et al., 2018; Akhtar et al., 2019; Ríssola et al., 2021). The ultimate goal of such system would be to complement expert assessments, but such empirical studies are also valuable to better understand how communication is affected by health disorders. In this paper, we propose to investigate depression detection within dialogs, a scenario less studied but more similar to the interviews with clinicians, which allegedly involves dialog features and also allows to examine how interaction is affected.

However, depression detection suffers from data sparsity. In fact, using social media data was a way to tackle this issue, including considering data generated by self-diagnosed users – a method that leads to potentially noisy data and comes with ethical issues (Chancellor et al., 2019). We rather examine a dataset of 189 clinical interviews, the DAIC-WOZ (Gratch et al., 2014), collected by experts to support the diagnosis of distress conditions. Participants are identified as depressive or not, and if so they receive a severity score. A line of work proposed to overcome data scarcity by leveraging varied modalities, e.g., using audio as in Al Hanai et al. (2018). Previous approaches were solely based on textual information relied on hierarchical contextual attention networks on word and sentence-level representations (Mallol-Ragolta et al., 2019), or Multi-Task Learning (MTL) but limited to combing identification and severity prediction (Qureshi et al., 2019; Dinkel et al., 2019), possibly with emotion (Qureshi et al., 2020).

Inspired by the latter approaches, we also propose relying on the MTL framework to help our model leverage information from different sources. We exploit three auxiliary tasks: emotion classification – naturally tied to mental health states –, and dialog act and topic classification, hoping the shallow information about the dialog structure could further enhance the performance. Our architecture is classic, based on hard-parameter sharing (Ruder, 2017), simpler than the shared-private architecture in (Qureshi et al., 2020) but has shown effective. In order to take into account dialog organization, we advocate for a dialog-tailored hierarchical architecture with some tasks performed at the speech turn level and others at the document level.

[1] https://www.who.int/news-room/fact-sheets/detail/depression

Proceedings of the SIGdial 2022 Conference, pages 68–75
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Our contributions are: (i) An empirical study on depression detection in dialogs, leveraging the power of multi-task learning to deal with data sparsity; (ii) An extension of previous work in examining the effects of depression on dialog structure via shallow markers, i.e., dialog acts and topics, as a first step; (iii) State-of-the-art results on depression detection in DAIC test set with 70.6% in F_1 at best.

2 Related work

Within multi-task learning (MTL), a model has to learn shared representations to generalize the target task better. It improves the performance over single-task learning (STL) by leveraging commonalities or correlations between tasks. Recent years have witnessed a series of successful applications in various NLP tasks, as in Collobert and Weston (2008); Søgaard and Goldberg (2016); Ruder (2017); Ruder et al. (2019), which demonstrate the effectiveness of MTL in learning information from different but related sources. It also tackles the data sparsity issue and reduces the risk of overfitting (Mishra et al., 2017; Benton et al., 2017; Bingel and Søgaard, 2017).

Joshi et al. (2019) demonstrated the benefit of MTL for specific pairs of close health prediction tasks on tweets. Benton et al. (2017) used MTL on social media data and achieved important improvements in predicting several mental health signals, including suicide risks, depression, and anxiety, together with gender prediction. With a focus on depression detection, the shared task AVEC in 2016 (Valstar et al., 2016) has brought out a series of multi-modal studies using vocal and visual features on the DAIC-WOZ dataset (Gratch et al., 2014). Some of which also explored text-level features: Williamson et al. (2016) used Gaussian Staircase Model with semantic content features and reported a SOTA score on the validation set. Al Hanai et al. (2018) and Haque et al. (2018) learned sentence embeddings with an LSTM network. However, their results on textual features are lower than SOTA by a large margin. Dinkel et al. (2019) compared different word and sentence embeddings and various pooling strategies. Their best model is mean pooling with ELMo embeddings. Qureshi et al. (2019, 2020) proposed MTL approaches in adding emotion intensity and depression severity (i.e., a regression problem) prediction to the main classification task. They, however, found that the emotion-unaware model obtained the best result. They used a monologue corpus for the emotion task, a domain bias that possibly harms the performance. On the contrary, we hypothesize that emotional information would benefit depression detection. Mallol-Ragolta et al. (2019) used a hierarchical contextual attention network with static word embeddings within a single-task setting and then combined representations at the word and sentence levels. They reported at best 63% in F_1. Recently, Xezonaki et al. (2020) presented even better results, 70% in F_1, by augmenting the attention network with a conditioning mechanism based on effective external lexicons and incorporating the summary associated with each interview. We instead rely on MTL in this work, where incorporating external sources is more direct.

None of the previous studies investigated potential links between depression and dialog structure. We note that Cerisara et al. (2018) explored MTL with sentiment[2] and dialog act prediction on Mastodon (a Twitter-like dataset), where both annotations are available, and found a positive correlation. To the best of our knowledge, we are the first to tackle depression detection in dialog transcriptions with the MTL approach and explore joint learning techniques with tasks related to the dialog structure.

3 Model Architecture

One condition generally assumed for success within MTL, at least in NLP, is that the primary and auxiliary tasks should be related (Ruder, 2017). The emotion-related task is thus a natural choice since it is linked to mental states. We hypothesize that depressive disorder can also affect how people interact with others during conversations. We thus take a first step toward linking dialog structure and depression by examining shallow signals: dialog acts and topics. In addition, since the information comes at different levels, we propose hierarchical modeling, from speech turns to documents.

Baseline Model: Our basic model is a two-level recurrent network, similar to the one in Cerisara et al. (2018). The input words are mapped to vectors using word embeddings from scratch. The first level (*turn*-level) takes the embeddings into a bi-

[2]Sentiment and emotion are closely related with different function and/or granularity, cf. Munezero et al. (2014). Cerisara et al. (2018) use three labels for sentiment: *positive*, *negative*, *neutral*. In this paper, we use seven emotional labels: *anger*, *disgust*, *fear*, *happiness*, *sadness*, *surprise*, *neutral*.

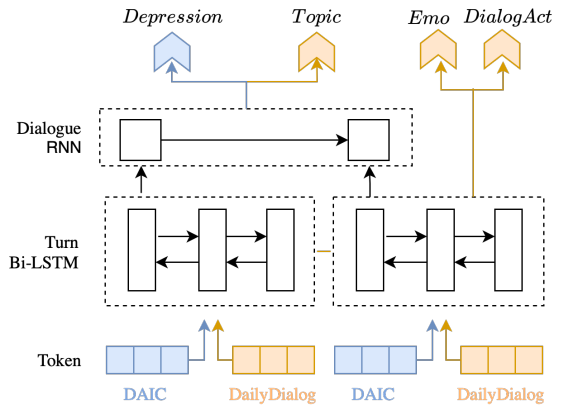


Figure 1: Multi-task fully shared hierarchical structure. Light blue is for DAIC dataset and depression task; orange is for DailyDialog and three auxiliary tasks.

LSTM network to obtain one vector for each turn. The second level (*dialog*-level) takes a sequence of turns into an RNN network, and the output is finally passed into a linear layer for depression prediction.

MTL Model: The MTL architecture is composed of shared hidden layers and task-specific output layers (see Fig. 1) and corresponds to the hard parameter sharing approach (Caruana, 1993, 1997; Ruder, 2017). Since some auxiliary tasks are annotated at the speech-turn level (i.e., emotion, dialog act) while others document level (i.e., depression, topic), our architecture is hierarchical and arranges task-specific output layers (MLP) at two levels. Sentence level emotion and dialog act information can be learned in the *turn*-level LSTM network and transferred upwards to help depression and topic prediction. On the other hand, higher-level information can be backpropagated to update the network at the lower level. The loss is simply the sum of the losses for each task. Regarding the MTL setting, we set equal weight for each task as the standard choice.

4 Datasets

DAIC-WOZ: This dataset is a subset of the DAIC corpus (Gratch et al., 2014).[3] It contains 189 sessions (one session is one dialog with avg. 250 speech turns) of two-party interviews between participants and Ellie – an animated virtual interviewer controlled by two humans. Table 1 gives the partition of train (107), development (35), and test (47) sets. Originally, patients are associated with a score related to the Patient Health Questionnaire (PHQ-9): a patient is considered depressive if PHQ-9 ≥ 10 (Kroenke and Spitzer, 2002).

	Train	Dev	Test
Depressed	77	23	33
Non Depressed	30	12	14
Total	107	35	47

Table 1: Number of sessions (dialogs) in DAIC-WOZ.

DailyDialog: This dataset (Li et al., 2017) contains 13,118 two-party dialogs (with averaged 7.9 speech turns per dialog) for English learners,[4] covering various topics from ordinary life to finance. Three expert-annotated information are provided: 7 emotions (Ekman, 1999), 4 coarse-grain dialog acts, and 10 topics. We select this corpus due to its large size, two-level annotations and high quality. The train set contains $> 87k$ turns for emotions and dialog acts and $> 11k$ dialogs for topics. Detailed statistics are given in Appendix A.

5 Experimental setup

Baselines: We compare our MTL results with: (1) Majority class where the model predicts all positive; (2) Baseline single-task model (see Sec. 3); (3) State-of-the-art results on test set reported by Mallol-Ragolta et al. (2019) and Xezonaki et al. (2020). We do not compare to (Williamson et al., 2016; Haque et al., 2018; Al Hanai et al., 2018; Dinkel et al., 2019; Qureshi et al., 2020) who only report on the development set.

Evaluation Metrics: For depression classification we follow Dinkel et al. (2019) and report accuracy, macro-F_1, precision, and recall. For emotion analysis, we follow Cerisara et al. (2018) and report macro-F_1.

Implementation Details: We implement our model with AllenNLP library (Gardner et al., 2018). We use the original separation of train, validation, and test sets for both corpora.

The model is trained for a maximum of 100 epochs with early stopping. For STL as well as for MTL scenario, we optimize on macro-F_1 metric for depression classification. We use cross-entropy loss. The batch size is 4 for DailyDialog and 1 for DAIC (within the limit of GPU VRAM). We

[3] https://dcapswoz.ict.usc.edu

[4] http://yanran.li/dailydialog

use the tokenizer from spaCy Library (Honnibal et al., 2020) and construct the word embeddings by default with a dimension of 128. The *turn* level has one hidden layer and 128 output neurons. We tune *document* RNN layers in $\{1, 2, 3\}$ and hidden size in $\{128, 256, 512\}$. Model parameters are optimized using Adam (Kingma and Ba, 2014) with $1e-3$ learning rate. Dropout rate is set to 0.1 for both *turn* and *document* encoders. The source code is available at https://github.com/chuyuanli/MTL4Depr.

6 Results and Discussion

6.1 Depression Detection Results on DAIC

Results using MTL hierarchical structure are shown in Table 2, which are compared to majority vote and SOTA models (at the top). Our baseline model is a single-task naive hierarchical model which obtains similar results (F_1 44) as the baseline model (NHN) in Mallol-Ragolta et al. (2019) (F_1 45).

Using the multi-task architecture, we get improvements when adding each task separately. We see more than a +11.5% increase in F_1 when adding emotion ('+Emo') or topic ('+Top') classification task and, at best, +16.9% with dialog acts ('+Diag'). This demonstrates the relevance of each task to the primary problem of depression detection, especially the interest of dialog acts. When adding topics, we observe a small drop in accuracy compared to STL while the F_1 is better, meaning that the prediction for minority class (non-depressive) improves. Interestingly, in terms of accuracy, the tasks at different levels (depression '+Emo' and depression '+Diag') seem to help more. We deduce that they help build a better local representation (speech turns) before the global representation.

When jointly learning all four tasks – combining depression detection with three auxiliary tasks ('+Emo+Diag+Top') –, all metrics improve. We obtain our best system with an improvement of +26.7% in F_1 compared to STL baseline, outperforming the state-of-the-art with a +7.6% increase compared to the best system in Mallol-Ragolta et al. (2019) and about +0.5% compared to Xezonaki et al. (2020). Depressed people tend to express specific emotions; it is thus natural to think that emotion is beneficial for the main task. These results indicate that both emotion and dialog structure help as they provide complementary information, paving the way for new research directions with more fine-grained modeling of dialog structure for tasks in conversational scenarios.

	F_1	Prec.	Rec.	Acc.
BSL Majority vote	41.3	35.1	50.0	70.2
State-of-the-art				
NHN[5] (Mallol-Ragolta et al., 2019)	45	-	50	-
HCAN[6] (Mallol-Ragolta et al., 2019)	63	-	66	-
HAN+L[7] (Xezonaki et al., 2020)	70	-	70	-
Ours				
STL Depression	43.9	44.5	47.5	63.8
MTL +Emo	55.5	56.2	61.6	70.2
MTL +Top	55.6	55.9	56.8	59.6
MTL +Diag	60.8	60.6	61.4	66.0
MTL +Emo+Diag+Top	**70.6***	**70.1**	**71.5***	**74.5**

Table 2: Depression detection results on DAIC. STL: single-task using DAIC only; MTL: multi-task using DAIC and adding classification for Emotion (+Emo), Topic (+Top), Dialog Act (+Diag) from DailyDialog. *Significantly better than SOTA performance with p-value < 0.05.

6.2 Analysis

Performance on Auxiliary Tasks: To better understand our model, we look at the performance of emotion, dialog act, and topic auxiliary tasks. Directly comparing the results of our MTL approach ('+Emo+Diag+Top') with a STL architecture for each task, however, seems unfair. The optimized objective and structural complexity are different: the former is optimized on the depression detection task on two levels, while the latter is tuned on the target auxiliary task with either speech turn (emotion and dialog act) or full dialog (topic). Unsurprisingly, the results show that the MTL system underperforms the basic STL structure for dialog acts and topics, with at best 67.8 in F1 (MTL) *vs.* 68.8 (STL) for dialog acts, and 52.0 (MTL) *vs.* 52.4 (STL) for topic classification.

For emotion, on the other hand, our best MTL system obtains 40.0 in F_1 compared to 38.3 for the STL baseline, showing the mutual benefit of both tasks. Even though the score is lower than the SOTA for emotion classification (51.0 F_1 in Qin et al. (2021))[8], we believe that refining our model for this task could lead to further improvements in depression detection. In addition, we observe that our MTL approach is particularly beneficial for negative and rare emotion classes, with *anger*,

[5]Naive hierarchical network (baseline).

[6]Hierarchical contextual attention network.

[7]Hierarchical attention network with LIWC lexicon.

[8]Precision: in Qin et al. (2021) authors report results on sentiment classification. It is yet unclear how they convert emotion annotation (7 labels) to sentiment (3 labels).

High-level DA	#	%	Sub-cat.	#	%
Question	7,907	53%	Emo	1,054	13%
			Non-emo	6,853	87%
Backchannel	3,231	22%	-	-	-
Comment	3,074	20%	-	-	-
Opening	611	4%	-	-	-
Other	171	1%	-	-	-

Table 3: High-level dialog act distribution of Ellie in DAIC-WOZ. # and % represent the number and percentage of Ellie's utterances, respectively.

disgust and *sadness* gaining resp. 5%, 6% and 1% in F_1. Finally, we conduct a manual inspection of the types of utterances (mostly questions) from Ellie, and classify them into high-level dialog acts: *Backchannel, Comment, Opening, Other, Question*.[9] We find that around 13% of the utterances are emotion-related, for instance "things which make you mad / you feel guilty about, last time feel really happy", etc., and that mentions of topics related to happiness or regret appear in almost all the interviews. Dialog act distribution is shown in Table 3. We release our annotation to the community for future studies.

Effectiveness of Hierarchical Structure: To examine the effectiveness of hierarchical structure, we conduct ablation studies on the full multi-learning setting ('+Emo+Diag+Top'). For dialog RNN level, we use topic information; for turn level, we test either emotion or dialog act. The results are shown in Table 4. Unsurprisingly, both ablated models ('+Emo+Top' and '+Diag+Top') underperform the full model, with F_1 scores decreasing $\approx 6\%$ each. Without dialog act, all metrics drop, showing the importance of this information for dialog structure. Without emotion, recall drops dramatically while accuracy and precision increase, indicating that the model '+Diag+Top' predicts more positive classes but fails in negative ones, which could result in too many false positives in real-life scenarios. On the other hand, when comparing hierarchical models ('+Emo+Top', '+Diag+Top', '+Emo+Diag+Top') with single-level models ('+Emo', '+Top', '+Diag'), we see considerable improvements in all metrics, and this holds for all auxiliary tasks. We can thus confirm the advantage of hierarchical structure for model performance.

[9] *Backchannel* refers to phatic expressions such as *yeah, hum mm*. Here we use different dialog acts from those in DailyDialog.

		F_1	Prec.	Rec.	Acc.
MTL	+Emo+Diag+Top	**70.6**	70.1	**71.5**	74.5
MTL	+Emo+Top	64.4	64.4	64.4	70.2
MTL	+Diag+Top	63.7	**78.1**	62.8	**76.6**

Table 4: Ablation study on hierarchical structure.

7 Conclusion

In this paper, we demonstrate the correlation between depression and emotion and show the relevance of features linked to dialog structures via shallow markers: dialog acts and topics. In the near future, we intend to investigate more refined modeling of dialog structures, possibly relying on discourse parsing (Shi and Huang, 2019). We would also like to explore depression severity classification as an extension to binary classification, possibly through a cascading structure: first, detect depression and then classify the severity. We intend to refine our work and report on cross-validation splits of the data to test the stability of the model, an issue even more crucial when dealing with sparse data with possibly representativeness problem. A further step will be to investigate the generalization of our model to other mental health disorders.

Acknowledgement

The authors thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the PIA project "Lorraine Université d'Excellence", ANR-15-IDEX-04-LUE, as well as the CPER LCHN (Contrat de Plan État-Région - Langues, Connaissances et Humanités Numériques). It was also partially supported by the ANR (ANR-19-PI3A-0004) through the AI Interdisciplinary Institute, ANITI, as a part of France's "Investing for the Future — PIA3" program, and through the project AnDiAMO (ANR-21-CE23-0020). Experiments presented were carried out in secured clusters on the Grid'5000 testbed. We would like to thank the Grid'5000 community (https://www.grid5000.fr).

Ethical Considerations

The goal of such systems is not to replace human healthcare providers. All these systems may be used only in support to human decision. The principle of leaving the decision to the machine would imply major risks for decision making in the health field, a mistake that in high-stakes healthcare set-

tings could prove detrimental or even dangerous.

Another issue is the representativeness of the data. Currently, it is very complex to access patients in order to have more examples. The institutional complexity leads researchers to systematically use the same data set, creating a bias between the representation of the pathology, in particular for mental ones whose expression can take very varied forms. This also implies defining a variation in relation to a normative use of language that comes with a strong risk in this type of approach.

Moreover, we carefully select the dialog corpora used in this paper to control for potential biases and personal information leakage. We only work with interview transcription, with no audio or visual information. For the text part, all the participant's name have been marked out with pseudo-ID.

References

Shad Akhtar, Deepanway Ghosal, Asif Ekbal, Pushpak Bhattacharyya, and Sadao Kurohashi. 2019. All-in-one: Emotion, sentiment and intensity prediction using a multi-task ensemble framework. *IEEE transactions on affective computing*.

Tuka Al Hanai, Mohammad M Ghassemi, and James R Glass. 2018. Detecting depression with audio/text sequence modeling of interviews. In *Interspeech*, pages 1716–1720.

Adrian Benton, Margaret Mitchell, and Dirk Hovy. 2017. Multitask learning for mental health conditions with limited social media data. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 152–162, Valencia, Spain. Association for Computational Linguistics.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain. Association for Computational Linguistics.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Machine learning: Proceedings of the tenth international conference*, pages 41–48.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Monica Cepoiu, Jane McCusker, Martin G Cole, Maida Sewitch, Eric Belzile, and Antonio Ciampi. 2008. Recognition of depression by non-psychiatric physicians—a systematic literature review and meta-analysis. *Journal of general internal medicine*, 23(1):25–36.

Christophe Cerisara, Somayeh Jafaritazehjani, Adedayo Oluokun, and Hoa T. Le. 2018. Multi-task dialog act and sentiment recognition on mastodon. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 745–754, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Stevie Chancellor, Michael L Birnbaum, Eric D Caine, Vincent MB Silenzio, and Munmun De Choudhury. 2019. A taxonomy of ethical tensions in inferring mental health states from social media. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 79–88.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Glen Coppersmith, Mark Dredze, Craig Harman, Kristy Hollingshead, and Margaret Mitchell. 2015. Clpsych 2015 shared task: Depression and ptsd on twitter. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality*, pages 31–39.

Heinrich Dinkel, Mengyue Wu, and Kai Yu. 2019. Text-based depression detection on sparse data. *arXiv preprint arXiv:1904.05154*.

Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Jonathan Gratch, Ron Artstein, Gale Lucas, Giota Stratou, Stefan Scherer, Angela Nazarian, Rachel Wood, Jill Boberg, David DeVault, Stacy Marsella, David Traum, Skip Rizzo, and Louis-Philippe Morency. 2014. The distress analysis interview corpus of human and computer interviews. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3123–3128, Reykjavik, Iceland. European Language Resources Association (ELRA).

Sharath Chandra Guntuku, David B Yaden, Margaret L Kern, Lyle H Ungar, and Johannes C Eichstaedt. 2017. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49.

Albert Haque, Michelle Guo, Adam S Miner, and Li Fei-Fei. 2018. Measuring depression symptom severity from spoken language and 3d facial expressions. *arXiv preprint arXiv:1811.08592*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Aditya Joshi, Sarvnaz Karimi, Ross Sparks, Cecile Paris, and C Raina MacIntyre. 2019. Does multi-task learning always help?: An evaluation on health informatics. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 151–158, Sydney, Australia. Australasian Language Technology Association.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kurt Kroenke and Robert L Spitzer. 2002. The phq-9: a new depression diagnostic and severity measure.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Adria Mallol-Ragolta, Ziping Zhao, Lukas Stappen, Nicholas Cummins, and Björn W Schuller. 2019. A hierarchical attention network-based approach for depression detection from transcribed clinical interviews. *Proc. Interspeech 2019*, pages 221–225.

Abhijit Mishra, Kuntal Dey, and Pushpak Bhattacharyya. 2017. Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 377–387, Vancouver, Canada. Association for Computational Linguistics.

Myriam Munezero, Calkin Suero Montero, Erkki Sutinen, and John Pajunen. 2014. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE transactions on affective computing*, 5(2):101–111.

Libo Qin, Zhouyang Li, Wanxiang Che, Minheng Ni, and Ting Liu. 2021. Co-gat: A co-interactive graph attention network for joint dialog act recognition and sentiment classification. In *AAAI*.

Syed Arbaaz Qureshi, Gaël Dias, Mohammed Hasanuzzaman, and Sriparna Saha. 2020. Improving depression level estimation by concurrently learning emotion intensity. *IEEE Computational Intelligence Magazine*, 15(3):47–59.

Syed Arbaaz Qureshi, Sriparna Saha, Mohammed Hasanuzzaman, and Gaël Dias. 2019. Multitask representation learning for multimodal estimation of depression level. *IEEE Intelligent Systems*, 34(5):45–52.

Esteban A Ríssola, David E Losada, and Fabio Crestani. 2021. A survey of computational methods for online mental state assessment on social media. *ACM Transactions on Computing for Healthcare*, 2(2):1–31.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv e-prints*, pages arXiv–1706.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7007–7014.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics.

Hoyun Song, Jinseon You, Jin-Woo Chung, and Jong C. Park. 2018. Feature attention network: Interpretable depression detection from social media. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Michel Valstar, Jonathan Gratch, Björn Schuller, Fabien Ringeval, Denis Lalanne, Mercedes Torres Torres, Stefan Scherer, Giota Stratou, Roddy Cowie, and Maja Pantic. 2016. Avec 2016: Depression, mood, and emotion recognition workshop and challenge. In *Proceedings of the 6th international workshop on audio/visual emotion challenge*, pages 3–10.

James R Williamson, Elizabeth Godoy, Miriam Cha, Adrianne Schwarzentruber, Pooya Khorrami, Youngjune Gwon, Hsiang-Tsung Kung, Charlie Dagli, and Thomas F Quatieri. 2016. Detecting depression using vocal, facial and semantic communication cues. In *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*, pages 11–18.

Danai Xezonaki, Georgios Paraskevopoulos, Alexandros Potamianos, and Shrikanth Narayanan. 2020. Affective conditioning on hierarchical attention networks applied to depression detection from transcribed clinical interviews. In *INTERSPEECH*, pages 4556–4560.

Andrew Yates, Arman Cohan, and Nazli Goharian. 2017. Depression and self-harm risk assessment in online forums. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2968–2978, Copenhagen, Denmark. Association for Computational Linguistics.

A Auxiliary Tasks Class Distribution in DailyDialog

Table 5, Table 6, and Table 7 show the number and percentage of emotion, dialog act, topic for each subset, resp.

Emotion	Train		Dev		Test	
	#	%	#	%	#	%
0-no emotion	72,143	82.8	7,108	88.1	6,321	81.7
1-anger	827	0.9	77	1.0	118	1.5
2-disgust	303	0.3	3	0.04	47	0.6
3-fear	146	0.2	11	0.1	17	0.2
4-happiness	11,182	12.8	684	8.5	1019	13.2
5-sadness	969	1.1	79	1.0	102	1.3
6-surprise	1,600	1.8	107	1.3	116	1.5
Utt. Total	87,170	100.0	8,069	100.0	7,740	100.0

Table 5: Emotion distribution in train, dev. and test sets.

Dialog Act	Train		Dev		Test	
	#	%	#	%	#	%
1-inform	39,873	45.7	3,125	38.7	3,534	45.7
2-question	24,974	28.6	2,244	27.8	2,210	28.6
3-directive	12,242	16.3	1,775	22.0	1,278	16.5
4-commissive	8,081	9.23	925	11.5	718	9.3
Utt. Total	87,170	100.0	8,069	100.0	7,740	100.0

Table 6: Dialog act distribution in train, dev. and test sets.

Topic	Train		Dev		Test	
	#	%	#	%	#	%
1-ordinary life	2,975	26.8	418	41.8	252	25.2
2-school life	453	4.1	0	0	34	3.4
3-culture & education	50	0	0	0.0	5	0.5
4-attitude & emotion	616	5.5	1	0.0	50	0.5
5-relationship	3,879	34.9	129	12.9	384	38.4
6-tourism	860	7.7	124	12.4	79	7.9
7-health	205	1.8	41	4.1	21	2.1
8-work	1,574	14.2	215	21.5	135	1.4
9-politics	105	0.9	13	1.3	13	1.3
10-finance	399	3.6	59	5.9	27	2.7
Total	11,118	100.0	1,000	100.0	1,000	100.0

Table 7: Topic distribution in train, dev. and test sets.

To laugh or not to laugh? The use of laughter to mark discourse structure

Bogdan Ludusan
Phonetics Workgroup, CITEC &
Faculty of Linguistics and Literary Studies
Bielefeld University, Germany
bogdan.ludusan@uni-bielefeld.de

Barbara Schuppler
Signal Processing and
Speech Communication Laboratory
Graz University of Technology, Austria
b.schuppler@tugraz.at

Abstract

A number of cues, both linguistic and non-linguistic, have been found to mark discourse structure in conversation. This paper investigates the role of laughter, one of the most encountered non-verbal vocalizations in human communication, in the signalling of turn boundaries. We employ a corpus of informal dyadic conversations to determine the likelihood of laughter at the end of speaker turns and to establish the potential role of laughter in discourse organization. Our results show that, on average, about 10% of the turns are marked by laughter, but also that the marking is subject to individual variation, as well as effects of other factors, such as the type of relationship between speakers. More importantly, we find that turn ends are twice more likely than transition relevance places to be marked by laughter, suggesting that, indeed, laughter plays a role in marking discourse structure.

1 Introduction

Despite the spontaneous nature of human communication, turn-taking between conversational partners occurs rather smoothly (Sacks et al., 1978), with interlocutors negotiating control of the floor through the marking of so-called transition relevance places (points in the conversation where a speaker change may occur) by means of various cues. A significant amount of work has been dedicated on investigating the acoustic characteristics involved in speaker-turn marking (e.g., Wichmann and Caspers, 2001; Gravano and Hirschberg, 2009; Niebuhr et al., 2013; Zellers, 2017). Yet, discourse structure has been shown to be signalled by a combination of different features (Duncan, 1972), both linguistic (e.g., lexical, syntactic, semantic) and non-linguistic. The latter type includes body movements and gestures, such as posture shifts (Cassell et al., 2001) and gaze (Jokinen et al., 2013), but also non-verbal vocalizations, in the form of breathing sounds (Włodarczak and Heldner, 2016).

We examine here one of the most commonly encountered non-verbal vocalizations in spontaneous interaction, laughter. It plays various roles in human communication (Trouvain and Truong, 2017), including social and communicative (Glenn and Holt, 2013) as well as linguistic roles (Mazzocconi et al., 2020). Evidence from conversational analysis suggests a possible role of laughter in discourse structure, as a cue marking the edges of speaker-turns (Gavioli, 1995; Ikeda and Bysouth, 2013; Madden et al., 2002). Most of this evidence is of qualitative nature, but there are also quantitative findings that offer additional support for this hypothesis. Norris and Drummond (1998) found that about 30% of total produced laughter occurred with the beginning and end of discourse structures, in materials based on tasks eliciting laughter. In a distributional analysis of laughter in task-based dyadic interactions, Ludusan et al. (2020) reported that turns for which laughter occurred at turn-initial or turn-final represented up to 50% of all turns containing laughter, in the three studied languages (French, German and Mandarin Chinese). Turns marked by laughter at their edges made up between 13% and 20% of total turns in the same materials (Ludusan and Wagner, 2022). Also the fact that laughter entrainment effects have been found at the turn-level in conversation (Ludusan and Wagner, 2022), represents further indication of the potential role of laughter in marking turns.

The aforementioned studies, however, presented only descriptive statistics of laughter events co-occurring with turn edges, without showing a relationship between laughter and discourse structure. Thus, we aim to establish in this study the possible role of laughter in marking turn boundaries, by comparing laughter at speaker turn versus at transition relevance places and by determining whether turn-holds or turn-changes are more likely to be marked by laughter. Moreover, as some of these studies used materials from tasks that elicited

Proceedings of the SIGdial 2022 Conference, pages 76–82
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

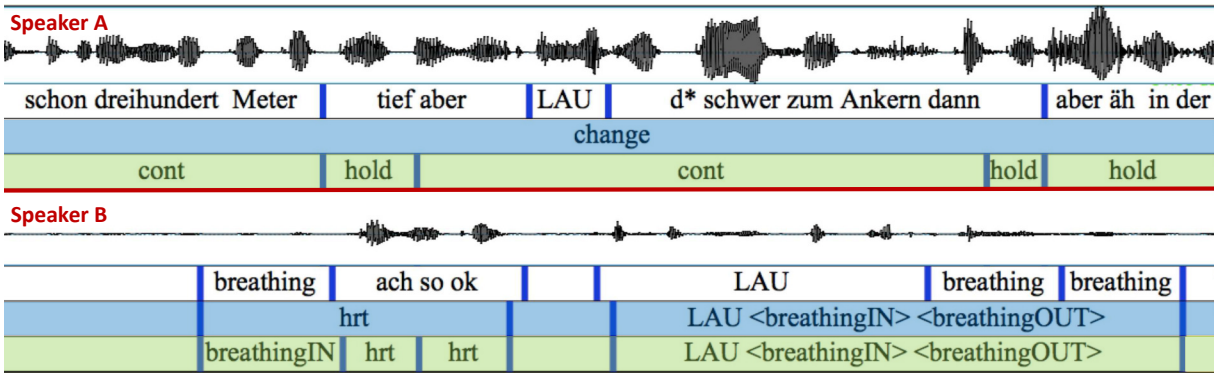


Figure 1: Conversation fragment from the GRASS corpus illustrating the discourse structure annotation. For each speaker (A and B), it shows the waveform of the recording, its orthographic transcription, the turn-level annotations (in blue), and the level of potential transition relevance places (in green). The laughter produced by the speakers is marked with *LAU*.

laughter and since laughter patterns in everyday conversations might differ from those produced in such tasks, we employ here informal conversations between friends/family members. We also evaluate the role of message-external factors, namely relation type between interlocutors and the gender composition of the dyad, as previous work has shown that they may play a role in the overall production of laughter (Smoski and Bachorowski, 2003).

2 Materials

The Graz Corpus of Read and Spontaneous Speech (GRASS) contains about 30 hours of Austrian German read and conversational speech, collected from 38 Austrian speakers (19 females, 19 males) (Schuppler et al., 2014). The conversational speech component contains speech from 19 pairs of speakers who had known each other for at least several years, and who were either friends, family-members, colleagues or couples, with a similar number of mixed-gender and same-gender dyads. These speaker pairs were recorded for one hour each, without interruption, in order to encourage a fluent, casual conversation. There were no restrictions in terms of topic or speaking behaviour, leading to the use of casual, partly dialectal pronunciation, frequent occurrence of overlapping speech, as well as laughter (laughs and speech-laughs) (Schuppler et al., 2017). This resulted in a wide variety of conversation topics, such as discussions about family or about public figures, travelling, relationship problems, or work-related issues.

The conversational speech component of GRASS is currently being manually annotated for discourse structure. As manual annotations are highly time consuming, in combination with limited resources, the manual annotation of the entire GRASS corpus is not possible. In order to capture as many different speakers and as many different communicative stages as possible, from each one-hour conversation, 5 minutes were annotated either from its beginning, its middle, or its end. So far, 14 dyads (5 f-m, 4 f-f, 5 m-m) were annotated, resulting in a total of 70 minutes of recordings available for this study.

Two independent discourse structure levels were annotated (cf., conversation example shown in Figure 1): one for turn management (based on inter-pausal units), further called *turn-level* (the blue tier in Figure 1), and one for potential transition relevance places (further called *TRP-level*), which were defined in terms of points of potential syntactic completion (the green tier in Figure 1). The turn-level labels were based on the four categories proposed in Zellers (2017): *hold* (the same speaker continues talking), *change* (a new speaker takes the floor), *question* (the speaker transfers the turn to another speaker), and *Hearer Response Tokens* (HRT, backchannel-like tokens, Sikveland, 2012). Three additional turn labels captured incomplete structures before pauses: *incomplete-hold* (the speaker makes a pause at a point of "maximum grammatical control", Schegloff, 1998: 241, and then continues speaking), *trail-off* (a syntactically incomplete speaker change, cf. Walker, 2012), and *self-interruption* (in the case of turn competition, one speaker interrupts themselves to cede the turn to the other speaker). The annotation at the TRP-level is more fine-grained, having the categories proposed by Zellers (2017) and six additional la-

bels. For further details on the different labels used for annotating the TRP-level, we refer the reader to Schuppler and Kelterer (2021). All annotations were created while listening to the recordings and were not based on the orthographic transcription alone. Thus, for example, the token "ja" (yes) may be assigned the label *HRT* in one instance, where it was produced with the function of a backchannel (i.e., no interruption of the turn of the interlocutor), or the label *change* in another instance, where it was produced with a question-like intonation followed by a turn of the interlocutor.

In order to guarantee a high annotation quality, the same process was applied to both discourse structure levels: First, the conversations were annotated by one trained annotator, self-corrected at a later point in time and then corrected by another, second annotator. In order to estimate the inter-rater agreement for the two discourse structure annotation levels, we evaluated a set of 878 word tokens from 3 different conversations. The Cohen's kappa on whether a TRP was placed at a word boundary or not was $\kappa = 0.96$. The agreement between the two turn-level labels *change* and *hold* (the only two categories we discriminated between in this study) was $\kappa = 0.83$. Thus, both levels of discourse annotations used for this study showed a very high inter-annotator agreement.

3 Methods

Based on the annotations of GRASS, we determined the units (both at the turn- and at the TRP-level) which were marked by laughter at their end. For this, the speaker having the floor or their interlocutor should have produced laughter either at the end of the unit, overlapping with the end of the unit, or immediately following (within one second) the unit. If the interlocutor produced the laughter, they should not have produced any other speech between the end of the unit marked by laughter and the start of the laughter instance. For the labelling process, other non-verbal vocalizations, such as in- or out-breaths and coughs, were not considered as being speech. All units were labelled for the existence of laughter in the analysis, except for the HRT tokens, which do not represent an actual conversational turns. Although not included in the analysis, HRT were taken into account for the labelling of turn-units: If a speaker turn-end overlapped or was followed by an HRT of the conversational partner containing laughter, the turn was labelled as being marked with laughter. Statistics about the total number of units in our data, the ones analysed here (non-HRT) and the units marked with laughter can be found in Table 1.

Level	Total	Analysed	Laughter
Turn	1874	1313	125
TRP	3772	3071	64

Table 1: The number of units considered in this study. For each analysis level (turn/TRP), the total number of units, the number of analysed units (non-HRT), and the units marked by laughter are shown.

We then counted, for each speaker and each level, the number of units signalled by laughter and the number of units not signalled by laughter. These counts, representing together the odds of units having laughter (number of successes and failures), were used as dependent variable in a mixed effects logistic model, to determine whether a significant difference exists between the marking of two levels. The unit-level (turn/TRP) was employed as predictor in the model and the speaker was introduced as a random intercept. Three logistic models were then fitted on the data consisting of the turn-level counts, in order to determine the effect of several message-external factors on the signalling of turns with laughter. We considered the dyad identity (ranging between 1 and 14), its gender composition (f-f, f-m or m-m) or the relation between the conversational partners (colleagues, couples, family or friends), as the independent variables in those models. Finally, we checked whether turn-marking with laughter occurs more often for turn-change or for turn-hold. For this, we deemed all turns labelled as incomplete-hold and hold to represent a turn-hold and the remaining labels to represent a turn-change. We then tested the probability of having a turn-change marked by laughter, out of the total number of turns marked by laughter, by means of a binomial test. The R (R Core Team, 2019) software was used for all statistical analyses, with the mixed effects model being fitted by means of the lmerTest package (Kuznetsova et al., 2017), based on the lme4 package (Bates et al., 2015) functionalities.

4 Results

First, we examined the likelihood of laughter in marking turns. Figure 2 illustrates the proportion of speaker turns followed by laughter, out of the

total number of turns produced by each speaker. Speakers were grouped based on the dyad they were part of and each speaker is represented by a point. On average, across dyads, 10.6% of all turns are marked by laughter (represented by a solid horizontal line), but there is significant variation across speakers (from a minimum of 0% for speaker B in dyad 3 to a maximum of 43.8% for speaker A in dyad 11). We checked whether the marking of turns by the various dyads differs significantly from mean value, by means of a logistic regression model with the dyad ID as predictor and employing a sum to zero contrast. Only three dyads (3, 11 and 13) showed significant differences from the overall mean.

Then, with regards to the effect of message-external factors on the laughter-marking of turns, we examined the purposely built logistic models, having either the relation between speakers or the gender composition of the dyads as independent variable. Logistic models estimate the effect of the predictors on the log odds ratio of success vs. failure (here, the probability of a turn to be marked vs. not be marked by laughter). Higher odds indicate a higher probability of turns being marked by laughter. For the relation status, the highest odds were seen for the dyads made up of couples (the intercept of the model, $\beta = -1.998$), followed by family ($\beta = -0.301$, $p = .270$), friends ($\beta = -0.460$, $p = .051$), and the lowest odds for colleagues ($\beta = -0.725$, $p = 0.008$). Regarding the gender composition, the highest odds were observed for the female-female dyads (intercept, $\beta = -2.187$), with similar odds for mixed gender dyads ($\beta = -0.025$, $p = .913$) and lower odds for all-male dyads ($\beta = -0.551$, $p = .028$). The difference between mixed-gender and all-male dyads was also found significant ($p = .018$).

Next, we estimated whether there is an effect of the discourse level where laughter is used for marking the structure (turn/TRP). Employing the mixed effects model described in the Methods section, we obtained a significant effect of the level ($p = 1.3e^{-6}$), with the odds of a laughter-marked structure increasing by 107% (95% confidence interval: [0.54, 1.78]) at the turn-level compared to at the TRP-level. While the intercept of the model showed that the probability of a TRP to be signalled by laughter is about 4%, it increases more than twice in the case of turn boundaries.

Finaly, we looked in more detail at which types

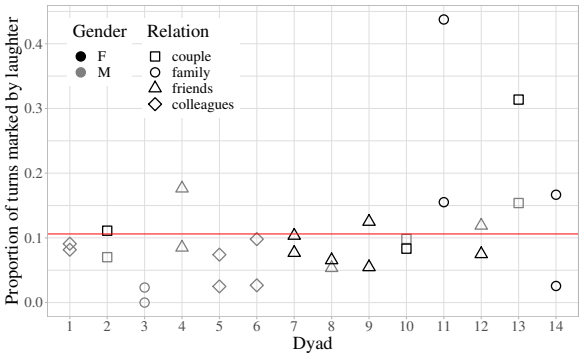


Figure 2: The proportion of turns marked by laughter, out of the total number of turns produced by each speaker. The results are illustrated on a per-dyad basis, with each dyad being represented by two data points, one for each dyad member. Each speaker is coded by a colour, representing their gender, and a shape, encoding their relation with the interlocutor. The horizontal line represents the average proportion across all dyads.

of turns were more likely marked by laughter. The conducted binomial test showed a significant preference for turn-changes ($p = .007$), with a probability of 0.62 (95% confidence interval [0.53, 0.71]).[1]

5 Discussion

Based on our data from casual conversations between family members or friends, we have found that turn boundaries tend to be signalled by laughter, on average in 10% of the cases. This represents a lower value than those reported by Ludusan and Wagner (2022), in which between 13% and 20% of turns were marked by laughter, across the three studied languages. Moreover, in the latter case, backchannels were counted as turns, thus a higher proportion of turns might be marked by laughter if one were not to consider backchannels, as in our case. These differences may well reflect the different data elicitation methods. The data employed by Ludusan and Wagner (2022) consisted of recordings in which a significant amount of laughter was expected, due to the nature of the considered task (coming up with an idea for a film script based on an embarrassing moment). This emphasizes the role of the type of data employed in the investigation: In a context consisting of casual conversations between individuals that are close to each other, a lower proportions of turns are signalled by laughter. The observed laughter-marking behaviour seems to

[1]The fact that turns signalled by partner laughter were included in the analysis did not bias these results, as there was a higher proportion of turn-holds (0.40) than turn-changes (0.18) marked by partner laughter, in our data.

be consistent in our data, with 11 out of the total 14 dyads showing no significant difference from the mean.

For both investigated message-external factors, the relation type between the conversational partners and the gender mix of the dyad, we observed significant effects on the laughter-marking of turns. Couples exhibited higher odds of turns marked with laughter than family members, friends and colleagues, although only the difference between couples and colleagues was found to be significant. Previous work looking at the effect of interlocutors' relation on laughter production (e.g. Smoski and Bachorowski, 2003; Jansen et al., 2021) considered two cases: familiar/unfamiliar, and the results were mixed, either showing a significant effect (Smoski and Bachorowski, 2003), or a lack of it (Jansen et al., 2021). Looking at the marking of turn edges by laughter, Ludusan and Wagner (2022) found no effect of familiarity (defined as the number of years the speaker knew each other). However, we employed here a definition based on the relationship between speakers, which may be more appropriate. With respect to the gender mix, we saw no difference between all-female and mixed-gender dyads, but significantly lower odds for all-male dyads compared to the other two groups. Our results partially align with work reporting more laughter in mixed-gender dyads composed of friends (Smoski and Bachorowski, 2003) (although a different behaviour may be seen for mixed-gender dyads composed of strangers Grammer and Eibl-Eibesfeldt, 1990; Smoski and Bachorowski, 2003). The observed differences may stem from the types of laughter considered in each study (laughter at turn boundaries here, all laughter instances in previous studies).

How does the marking of turns by means of laughter compare to the signalling of turns by other cues? Niebuhr et al. (2013) observed differences in speech reduction phenomena between turn-final and turn-internal positions of up to more than four times, while Cassell et al. (2001) found that posture shifts at turn boundaries were five times more likely than turn-internal. We have seen here that laughter turns are twice more likely to be signalled by laughter, than transition relevance places. While laughter may seem, therefore, a weaker cue to the marking of turns, one must take into account that we compared here turn-final laughter with laughter produced only at TRPs (not any turn-internal position). When comparing turn-final with phrase-final positions, also Niebuhr et al. (2013) showed that the difference in likelihood between these two levels is lower than between turn-final and any turn-internal location.

Among the types of considered turn-units, we observed a higher probability of turn-changes than turn-holds being marked with laughter. This finding indicates that laughter is one of the cues that speakers employ to signal the end of their turn or the taking of the floor from their interlocutor. While the current study did not examine the characteristics of the various turn-final laughter instances, it might be that giving/taking the turn may use different types of laughter (laughs vs. speech-laughs, snorts vs. grunts, etc) or laughs with different acoustic properties (voiced vs. unvoiced, etc.). Further investigations in this direction would be necessary to better understand the role of laughter in turn-taking. Moreover, studies on larger datasets as well as on other languages are welcome, in order to test the generalizability of these findings.

6 Conclusions

We investigated the role of laughter in the marking of speaker turns in a corpus of informal conversations between family members and friends. Besides establishing the frequency of occurrence of laughter at turn-ends, in a dataset not composed of task-based interactions, we also showed that laughter is twice more likely to occur at the end of turn-units than at TRPs. Next, we found that the probability of laughter-marked turn-changes was higher than for turn-holds, suggesting a possible role of laughter as a cue signalling turn-change. Finally, our study revealed that this laughter function is modulated by message-external factors, such as the nature of the relationship between speakers and the dyad gender composition. These results represent one step further in understanding the various functions that non-verbal phenomena and laughter, in particular, play in human communication.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) project number 461442180 to BL. BS was supported by grant P-32700-N from the Austrian Science Fund (FWF). We would like to thank Anneliese Kelterer for her support with the discourse structure annotations of GRASS, as well as for the computation of the inter-rater agreement.

References

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48.

Justine Cassell, Yukiko Nakano, Timothy Bickmore, Candace Sidner, and Charles Rich. 2001. Non-verbal cues for discourse structure. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 114–123.

Starkey Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.

Laura Gavioli. 1995. Turn-initial versus turn-final laughter: Two techniques for initiating remedy in English/Italian bookshop service encounters. *Discourse Processes*, 19(3):369–384.

Phillip Glenn and Elizabeth Holt. 2013. *Studies of laughter in interaction*. London: Bloomsbury Academic.

Karl Grammer and Irenaus Eibl-Eibesfeldt. 1990. The ritualisation of laughter. In Walter A. Koch, editor, *Natürlichkeit der Sprache und der Kultur*, pages 192–214. Brockmeyer Bochum, Germany.

Agustín Gravano and Julia Hirschberg. 2009. Turn-yielding cues in task-oriented dialogue. In *Proceedings of the SIGDIAL 2009 Conference*, pages 253–261.

Keiko Ikeda and Don Bysouth. 2013. Laughter and turn-taking: Warranting next speakership in multiparty interactions. In Phillip Glenn and Elizabeth Holt, editors, *Studies of laughter in interaction*, pages 39–64. London: Bloomsbury Academic.

Michel-Pierre Jansen, Khiet Truong, and Dirk Heylen. 2021. How familiarity influences the frequency, temporal dynamics and acoustics of laughter. In *Proceedings of the 9th International Conference on Affective Computing and Intelligent Interaction*, pages 1–8.

Kristiina Jokinen, Hirohisa Furukawa, Masafumi Nishida, and Seiichi Yamamoto. 2013. Gaze and turn-taking behavior in casual conversational interactions. *ACM Transactions on Interactive Intelligent Systems*, 3(2):1–30.

Alexandra Kuznetsova, Per Brockhoff, and Rune Christensen. 2017. lmertest package: tests in linear mixed effects models. *Journal of Statistical Software*, 82(13):1–26.

Bogdan Ludusan and Petra Wagner. 2022. Laughter entrainment in dyadic interactions: Temporal distribution and form. *Speech Communication*, 136:42–52.

Bogdan Ludusan, Maik Wesemann, and Petra Wagner. 2020. A distributional analysis of laughter across turns and utterances. In *Proceedings of the Workshop on Laughter and Other Non-verbal Vocalisations*, pages 28–31.

Mary Madden, Mary Oelschlaeger, and Jack Damico. 2002. The conversational value of laughter for a person with aphasia. *Aphasiology*, 16(12):1199–1212.

Chiara Mazzocconi, Ye Tian, and Jonathan Ginzburg. 2020. What's your laughter doing there? A taxonomy of the pragmatic functions of laughter. *IEEE Transactions on Affective Computing*, pages 1–19.

Oliver Niebuhr, Karin Görs, and Evelin Graupe. 2013. Speech reduction, intensity, and F0 shape are cues to turn-taking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 261–269.

Melissa Norris and Sakina Drummond. 1998. Communicative functions of laughter in aphasia. *Journal of Neurolinguistics*, 11(4):391–402.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Harvey Sacks, Emanuel Schegloff, and Gail Jefferson. 1978. A simplest systematics for the organization of turn taking for conversation. In *Studies in the Organization of Conversational Interaction*, pages 7–55. Elsevier.

Emanuel Schegloff. 1998. Reflections on studying prosody in talk-in-interaction. *Language and Speech*, 41(3-4):235–263.

Barbara Schuppler, Martin Hagmüller, Juan Andres Morales-Cordovilla, and Hannes Pessentheiner. 2014. GRASS: the Graz corpus of Read And Spontaneous Speech. In *Proceedings of LREC*, pages 1465–1470.

Barbara Schuppler, Martin Hagmüller, and Alexander Zahrer. 2017. A corpus of read and conversational Austrian German. *Speech Communication*, 94:62–74.

Barbara Schuppler and Anneliese Kelterer. 2021. Developing an annotation system for communicative functions for a cross-layer ASR system. In *Proceedings of the Integrating Perspectives on Discourse Annotation Workshop*, pages 1–5.

Rein Ove Sikveland. 2012. Negotiating towards a next turn: Phonetic resources for 'doing the same'. *Language and Speech*, 55(1):77–98.

Moria Smoski and Jo-Anne Bachorowski. 2003. Antiphonal laughter between friends and strangers. *Cognition and Emotion*, 17(2):327–340.

Jürgen Trouvain and Khiet Truong. 2017. Laughter. In *The Routledge Handbook of Language and Humor*, pages 340–355. Routledge.

Gareth Walker. 2012. Coordination and interpretation of vocal and visible resources: 'trail-off' conjunctions. *Language and Speech*, 55(1):141–163.

Anne Wichmann and Johanneke Caspers. 2001. Melodic cues to turn-taking in English: evidence from perception. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, pages 1–6.

Marcin Włodarczak and Mattias Heldner. 2016. Respiratory turn-taking cues. In *Proceedings of INTERSPEECH 2016*, pages 1275–1279.

Margaret Zellers. 2017. Prosodic variation and segmental reduction and their roles in cuing turn transition in Swedish. *Language and Speech*, 60(3):454–478.

QualityAdapt: an Automatic Dialogue Quality Estimation Framework

John Mendonça[1,2,*], **Alon Lavie**[3] and **Isabel Trancoso**[1,2]
[1]INESC-ID, Lisbon
[2]Instituto Superior Técnico, University of Lisbon
[3]Unbabel, Pittsburgh
{john.mendonca, isabel.trancoso}@tecnico.ulisboa.pt
alon.lavie@unbabel.com

Abstract

Despite considerable advances in open-domain neural dialogue systems, their evaluation remains a bottleneck. Several automated metrics have been proposed to evaluate these systems, however, they mostly focus on a single notion of quality, or, when they do combine several sub-metrics, they are computationally expensive. This paper attempts to solve the latter: **QualityAdapt** leverages the Adapter framework for the task of Dialogue Quality Estimation. Using well defined semi-supervised tasks, we train Adapters for different subqualities and score generated responses with AdapterFusion. This compositionality provides an easy to adapt metric to the task at hand that incorporates multiple subqualities. It also reduces computational costs as individual predictions of all subqualities are obtained in a single forward pass. This approach achieves comparable results to state-of-the-art metrics on several datasets, whilst keeping the previously mentioned advantages.

1 Introduction

Open-domain neural dialogue systems have increasingly drawn attention in Natural Language Generation (NLG). These systems, colloquially known as Chatbots, take advantage of large-scale training of complex models, making them increasingly more humanlike (Zhang et al., 2020; Adiwardana et al., 2020a; Roller et al., 2021). A crucial step in the development of a dialogue system is its evaluation. The community has identified multiple characteristics of what constitutes a high-quality dialogue. These include comprehensible, fluent, empathetic, relevant and interesting, among others. The precise definition is often challenging to define and is application dependent.

The current trend is to train models to evaluate responses under various aspects. These learning-based metrics either (1) map overall quality to a single defined aspect such as Sensibleness (is the response adequate given the context) or (2) leverage several individual models to cover a wider range of quality aspects (subqualities). Both have their drawbacks: in the first approach, the use of a single notion of quality limits the overall understanding of model performance and consequently its applicability to other domains; in the second approach, the need to individually train several models is both time and resource consuming, possibly duplicating model parameters that could be shared, such as feature representations.

This paper proposes QualityAdapt[1], an automatic dialogue quality estimation framework that leverages the Adapter paradigm (Houlsby et al., 2019a) to train individual Adapters on different dialogue subqualities. Then, AdapterFusion (Pfeiffer et al., 2021) combines the knowledge of the individual Adapters for the downstream task of overall quality estimation. This allows for a system that is both extensible (by including different subqualities) and less resource-intensive (by sharing most of the pretrained model parameters). Experimental results show that QualityAdapt achieves comparable correlations with human judgements when compared to other state-of-the-art metrics.

2 Background

2.1 Automatic Quality Estimation Metrics

Word-overlap metrics, such as BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005), are a popular choice to evaluate dialogues as they are used to evaluate machine translation and summarization models and are easy to employ. These metrics assume valid responses have significant word-overlap with the ground truth. However, this is not a valid assumption: there are many equally good responses for a single utterance. As

* Corresponding author

[1]Model parameters and codebase are available at: github.com/johndmendonca/qualityadapt.

Proceedings of the SIGdial 2022 Conference, pages 83–90
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

such, the correlation with human judgements is very low for these metrics (Liu et al., 2016), and they cannot be used to evaluate models in an online setting, where a gold-response is not available.

Earlier learned metrics such as ADEM (Lowe et al., 2017) and RUBER (Tao et al., 2018) explicitly predict human annotations by initialising pretrained RNN response generators. In both cases, a reference response is used to score the candidate response. As such, these metrics still suffer the same issues as word-overlap metrics.

More recently, open-domain automatic dialogue quality estimation has concentrated on reference-free methods. Most metrics focus on evaluating a single notion of quality such as Engagement (Ghazarian et al., 2020), Sensibleness (Dziri et al., 2019; Huang et al., 2020) or Human-likeness (Gao et al., 2020). Metrics such as USR (Mehri and Eskenazi, 2020b), USL-H (Phy et al., 2020) and Deep AM-FM (Zhang et al., 2021b) combine predictions of individual sub-metrics obtained from Language Models.

2.2 Adapters

Adapters in NLP (Houlsby et al., 2019b) have been introduced as an alternative to the full model fine-tuning strategy. They consist of a small set of additional trainable parameters added between layers of a pretrained network. These consist of feed-forward layers with normalizations, residual connections, and projection layers. The weights are trained during fine-tuning for a given task, while the pretrained parameters of the large model are kept frozen. This strategy allows for parameter sharing by training different task and language specific Adapters using the same model. Furthermore, previous work has shown that Adapters achieve comparable performance to full fine-tuning (Pfeiffer et al., 2020a, 2021), despite the primary focus being geared towards parameter efficiency.

AdapterFusion (Pfeiffer et al., 2021) proposes improving downstream task results by transferring task specific knowledge obtained from training Adapters on supporting tasks. The architecture takes inspiration from the attention mechanism (Vaswani et al., 2017), and consists of learnable weights Query, Key, and Value: the Query consists of the pretrained transformer weights; the Key and Value take as input the output of the respective Adapters. The dot product of the query with all the keys is passed into a softmax function, which learns to weight the Adapters with respect to the context. Therefore, the goal is to learn a parameterized mixer of the available trained Adapters.

3 QualityAdapt

QualityAdapt trains individual Adapters for each subquality and composes them using AdapterFusion for the task of overall quality estimation. In both the subquality and overall quality tasks, it returns a score that is obtained by combining a transformer encoder with a regression head on top. During inference, individual subquality predictions can be obtained in a single forward pass by parallelising their respective heads.

Encoder In our experiments, RoBERTa-large (Liu et al., 2019) is used to encode the context-response pair. In the tokenization step, we add for each utterance a token representative of the speaker. This added information lets the network identify the response's speaker, which in turn allows it to pay more attention to utterances from this speaker in the context if needed.

Compositionality Training AdapterFusion for the downstream task of overall quality estimation is a supervised task. As such, quality annotated data in terms of overall quality is required. However, the amount of annotations required for the Fusion training step is much smaller when compared to fully fine-tuning a Language Model with this data. As a proof of concept, we composed two Adapters in this paper: **U-Adapter**, for Understandability, and **S-Adapter** for Sensibleness.

U-Adapter An understandable response is one that can be understood without context. Such responses may contain minor typos that do not hinder the comprehension of the response. Mehri and Eskenazi (2020b) evaluates this sub-metric by calculating the likelihood of the response using a Masked Language Modelling (MLM) metric. In this paper, we follow the approach used by Phy et al. (2020) and initially proposed by Sinha et al. (2020). A model is trained to differentiate between positive samples and synthetic negative samples. Positive samples are perturbed by randomly applying one of the following: (i) no perturbation, (ii) punctuation removal, (iii) stop-word removal. Negative samples are generated by randomly applying one of the following rules: (i) word reorder (shuffling the ordering of the words); (ii) word-drop; and (iii) word-repeat (randomly repeating words).

S-Adapter A sensible response is one that takes into account its preceding context. The task of predicting sensibleness can be considered a binary Next Sentence Prediction (NSP) task, distinguishing a positive example (the subsequent utterance) from a semantically negative one (a random utterance from a response pool obtained from the dataset). Many dialogue quality estimation metrics leverage the NSP task when training their models for quality estimation (Zhao et al., 2020; Zhang et al., 2021a; Phy et al., 2020; Mehri and Eskenazi, 2020b).

4 Experiments

4.1 Datasets

Different data sources are used in the experiments:

Training – DailyDialog (Li et al., 2017) is used for the self-supervised training and evaluation of the S and U Adapters. Additionally, the Fusion module is trained using the annotated split by Zhao et al. (2020) (denoted as *DD-Z*).

Evalution – The evaluation of the subqualities is done on the data annotated by Phy et al. (2020) (denoted as *DD-P*). QualityAdapt's extensibility is also evaluated on different overall quality annotated datasets:

- TopicalChat (Gopalakrishnan et al., 2019) and PersonaChat (Zhang et al., 2018), which were annotated by Mehri and Eskenazi (2020b) and denoted in this work as *USR-TC* and *USR-PC*, respectively;
- *DSTC6* (Hori and Hori, 2017);
- *FED* (Mehri and Eskenazi, 2020a).

A more detailed overview of these datasets can be found in Appendix A.

4.2 Baselines

USR (Mehri and Eskenazi, 2020b) leverages several Language Models to measure dialogue properties. These include: *Fluency*, measured using masked language modelling (MLM) objectives; *Relevance*, using a dialog retrieval model and *Uses Knowledge*, measured using a fact-to-response selection model. Overall quality prediction is obtained using a Linear Regression model.

RoBERTa-eval (Zhao et al., 2020) proposes an evaluator that produces an encoding vector given a context and a response, and then calculates its score via an MLP with a sigmoid function. The model takes the pretrained transformer and primes it on an NSP task with in-domain data using Negative Sampling, which offsets the lack of annotated data. A final finetuning is done for quality prediction.

USL-H (Phy et al., 2020) combines three models trained with different objectives: Valid Utterance Prediction (BERT-VUP), Next Sentence Prediction (BERT-NSP), and BERT-MLM. The BERT-VUP model determines whether a response is valid and grammatically correct. The BERT-NSP model and BERT-MLM models are trained with self-supervised objectives to evaluate the sensibleness and the likelihood of a given response.

4.3 Subquality Estimation

		Pearson	Spearman
Understand.	BERT-MLM	-0.16	*0.01*
	BERT-VUP	0.26	*0.14*
	USR-MLM	*0.01*	*0.11*
	RoBERTa-large	**0.35**	*0.18*
	U-Adapter	0.32	**0.21**
Sensible	BERT-NSP	0.63	0.61
	USR-DR (x=c)	0.54	0.47
	RoBERTa-large	0.61	0.65
	S-Adapter	**0.68**	**0.67**

Table 1: Correlation for Understandability and Sensibleness subquality between human annotations and automatic metrics. Best results are denoted in **bold**, *italic* identifies $p > 0.01$.

The test set results on the DailyDialog dataset for the Understandability and Sensibleness subqualities are presented in Table 1. Here, we evaluate the correlation between the average human annotation and the model prediction. For fair comparison, we also include the results with a fully finetuned RoBERTa-large model. With respect to the estimation of Understandability, U-Adapter outperforms the models proposed by USR (USR-MLM perplexity) and USL-H (BERT-VUP). Similar results are observed on the Sensibleness task, where both RoBERTa and S-Adapter outperform both USL-H (BERT-NSP) and USR baselines. These results confirm Adapters are a valid substitute to fully finetuned models for the task of subquality estimation.

4.4 Overall Quality Estimation

In the overall quality prediction task, we compare the different metrics on all datasets. Results in Table 2 show that, on average, the S+U metric outperforms all other metrics on these datasets. As expected, all models obtain the best performance

	DD-Z		DD-P		USR-TC		USR-PC		DSTC6		FED		Avg	
	Pr.	Spr.	Pr.	Spr.	Pr.	Spr.	Pr.	Spr.	Pr.	Spr.	Pr.	Spr.	Pr.	Spr.
USR	0.38	0.39	0.51	0.48	**0.41**	**0.42**	0.44	0.42	0.18	0.17	0.11	0.12	0.34	0.33
USL-H	*0.25*	*0.26*	0.63	0.64	0.32	0.34	**0.50**	**0.52**	0.22	0.18	0.20	0.19	0.35	0.36
RoB-eval	0.64	0.66	0.73	0.74	0.22	0.22	0.34	0.33	0.28	0.29	**0.29**	**0.26**	0.42	0.41
S+U	**0.73**	**0.74**	0.76	**0.76**	0.29	0.29	0.36	0.36	**0.43**	**0.42**	0.27	0.23	**0.47**	**0.47**
-U Adapter	0.67	0.69	**0.80**	**0.76**	0.28	0.30	0.37	0.37	0.39	0.40	*0.17*	*0.13*	0.45	0.44
-Speaker	0.62	0.65	0.67	0.70	0.33	0.33	0.36	0.36	0.33	0.31	0.20	0.20	0.42	0.42
-Fusion	0.60	0.54	0.72	0.73	0.20	0.23	0.37	0.34	0.36	0.33	0.17	0.21	0.40	0.40
S+U+E	0.68	0.70	0.76	0.73	0.18	0.19	0.36	0.36	0.36	0.36	0.18	0.14	0.42	0.41

Table 2: Correlation for Overall Quality between human annotations and automatic metrics. Best results are denoted in **bold**, *italic* identifies $p > 0.01$. Baseline results are obtained using codebase provided by Yeh et al. (2021).

when evaluated on both DD test sets. Lowest results are obtained on the FED dataset, which contains responses from advanced chatbots, and are therefore more difficult to identify as being low-quality. This underlines the importance of including more subqualities for dialogue evaluation, as contemporary chatbots achieve human performance on typical subqualities such as sensibleness and understandability. This in turn makes them insufficient to discriminate between good and bad responses. However, finer-grained submetrics do not have an obvious mapping to semi-supervised data collection methods, and are therefore discarded due to the lack of sufficient annotated data to fully train models.

4.5 Ablation Studies

Single Adapter Finetuning In this experiment, we verify the effectiveness of having several Adapters trained on different objectives contributing to the performance of the downstream task. To evaluate this, the U-Adapter and the Fusion module is discarded and the S-Adapter is further finetuned with the quality annotated data (denoted in Table 2 as -U Adapter). On average, dropping the U-Adapter reduces relative performance by 5%.

Removing Speaker Tokens We compare the performance of S+U without the speaker tokenization (denoted in Table 2 as -Speaker). Results show the removal of these tokens reduces performance on all datasets except on USR-PC and USR-TC. This may indicate the topic shift between speakers is small and as such "who said what" is inconsequential to sensibleness.

Removing Adapter Fusion The contribution of AdapterFusion for the task of quality estimation is assessed by comparing S+U against a Linear Regression model that receives as input the predictions of the individual qualities obtained by the trained Adapters (denoted in Table 2 as -Fusion). The regression model is trained using the same annotated data split as AdapterFusion. Overall, the regression model yields worse results when compared against AdapterFusion. This underlines the power of composition using Fusion, leveraging the learned parameters of the trained Adapters instead of just their prediction.

4.6 Emotion Adapter

We posit the emotion conveyed by the agent during the conversation should positively correlate with overall quality annotations: responses that display happiness and excitement are expected to have a positive impact in the dialogue and therefore should favour higher quality annotations when compared to responses that portray neutral, or negative emotions. This was the basis for adding an Emotion Adapter to S+U, denoted S+U+E. The Adapter was trained on the DailyDialog corpus, using the same training parameters as the S and U Adapters, and a Weighted Cross Entropy Loss. A Macro-F1 of 45.00 is achieved on the test set. The inclusion of the emotion Adapter fails to outperform S+U. Our initial hypothesis is that this is due to generative models being conditioned to respond with positive emotions. We leave further investigation of these results for future work.

5 Prediction Compute

One of the motivations of the QualityAdapt framework is it's computational efficiency. We present average sample predictions per second on the test set using a single RTX 3070Ti 8BG GPU, together with size of the **metric's unique parameters** on Table 3. For the baseline methods, the transformer model is fully fine-tuned and therefore the full model in included; for the Adapters, only the Adapter, the fusion layer and corresponding heads are included in the calculation. We note that a full transformer model (RoBERTa-base/large) is

Metric	Samples/s	Model Params
USR	22.44	4.2 GB
USL-H	10.83	3.9 GB
RoBERTa-eval	79.11	3.2 GB
S (large)	59.67	17.1 MB
S+U (base)	107.29	168.8 MB
S+U (large)	59.11	319.1 MB
S+U+E (large)	59.24	332.1 MB

Table 3: Prediction loop compute on DD-Z (250 samples). For the QualityAdapt models, (base/large) denote the transformer model's size.

still required for inference in QualityAdapt. However, the sharing of its weights is simplified.

As expected, the forward pass on several transformer models decreases runtime performance when compared to a single forward pass, even when using larger models (USR and USL-H metrics are based on the RoBERTa and BERT-base models, respectively). When comparing between the different larger models, we can see that the inclusion of the Adapter model decreases run-time performance by 25%. However, both the fusion module and the inclusion of more Adapters does not significantly affect performance.

6 Conclusions

This paper presents QualityAdapt, a framework for automatic dialogue quality estimation. We show the composition of Sensibleness and Understandability Adapters for the downstream task of quality estimation outperforms, on average, the performance of robust baselines, including those that take advantage of subquality composition. However, QualityAdapt only requires a single forward pass on a Language Model to produce predictions for overall quality, thus reducing computational complexity.[2]

Current research in dialogue focuses mostly on monolingual chatbots, typically in English. Multilingual LMs such as XLM-RoBERTa (Conneau et al., 2020) can be used to extract utterance representations directly in the target language after fine-tuning. However, this approach would still be somewhat limited by the lack of multilingual annotated data. Pfeiffer et al. (2020b) proposes leveraging Adapters for transfer learning in low resource settings by training a stack consisting of the source-language Adapter with a task Adapter. Then, during inference, the source-language Adapter is replaced with the target-language one. We leave these experiments for future work.

[2]The parallel inference of individual Adapters and their fusion using AdapterHub is still WIP.

Acknowledgements

We would like to thank Patrícia Pereira for helping with emotion experiments, and the reviewers, for their helpful feedback and discussions. This work was supported by national funds through *Fundação para a Ciência e a Tecnologia* (FCT) with references PRT/BD/152198/2021 and UIDB/50021/2020, and by the P2020 program MAIA (LISBOA-01-0247-FEDER-045909).

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020a. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020b. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3806–3812, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. In

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 386–395, Online. Association for Computational Linguistics.

Sarik Ghazarian, Ralph Weischedel, Aram Galstyan, and Nanyun Peng. 2020. Predictive engagement: An efficient metric for automatic evaluation of open-domain dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7789–7796.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Chiori Hori and Takaaki Hori. 2017. End-to-end conversation modeling track in dstc6. *arXiv:1706.07440*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019a. Parameter-efficient transfer learning for nlp. In *Proceedings of the 36th International Conference on Machine Learning, PMLR*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019b. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Lishan Huang, Zheng Ye, Jinghui Qin, Liang Lin, and Xiaodan Liang. 2020. GRADE: Automatic graph-enhanced coherence metric for evaluating open-domain dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9230–9240, Online. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ryan Lowe, Michael Noseworthy, Iulian Vlad Serban, Nicolas Angelard-Gontier, Yoshua Bengio, and Joelle Pineau. 2017. Towards an automatic turing test: Learning to evaluate dialogue responses. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1116–1126.

Shikib Mehri and Maxine Eskenazi. 2020a. Unsupervised evaluation of interactive dialog with DialoGPT. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235, 1st virtual meeting. Association for Computational Linguistics.

Shikib Mehri and Maxine Eskenazi. 2020b. Usr: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020a. Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020b. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Vitou Phy, Yang Zhao, and Akiko Aizawa. 2020. Deconstruct to reconstruct a configurable evaluation metric for open-domain dialogue systems. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4164–4178, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L. Hamilton, and Joelle Pineau. 2020. Learning an unreferenced metric for online dialogue evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2430–2441, Online. Association for Computational Linguistics.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yi-Ting Yeh, Maxine Eskénazi, and Shikib Mehri. 2021. A comprehensive assessment of dialog evaluation metrics. *CoRR*, abs/2106.03706.

Chen Zhang, Yiming Chen, Luis Fernando D'Haro, Yan Zhang, Thomas Friedrichs, Grandee Lee, and Haizhou Li. 2021a. DynaEval: Unifying turn and dialogue level evaluation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5676–5689, Online. Association for Computational Linguistics.

Chen Zhang, Luis Fernando D'Haro, Rafael E Banchs, Thomas Friedrichs, and Haizhou Li. 2021b. Deep am-fm: Toolkit for automatic dialogue evaluation. In *Conversational Dialogue Systems for the Next Decade*, pages 53–69. Springer.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Tianyu Zhao, Divesh Lala, and Tatsuya Kawahara. 2020. Designing precise and robust dialogue response evaluators. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Online. Association for Computational Linguistics.

A Experiments

A.1 Datasets

DailyDialog (Li et al., 2017) is a high-quality human-human open-domain dialogue dataset focused on day-to-day conversations. The dataset consists of 13,118 dialogues and 103,632 utterances. Zhao et al. (2020) (DD-Z) annotates 900 context-response pairs in terms of *Appropriateness* from a pool of responses obtained by negative-sampling response randomly selected from a different dialogue and responses generated by generative models trained on the training split; Phy et al. (2020) (DD-P) collected five responses from two retrieval methods, two generative methods, and one human-generation for 50 contexts. These responses are then annotated in terms of *Understandability*, *Sensibleness*, *Specificity* and *Overall Quality*.

TopicalChat (Gopalakrishnan et al., 2019) is a knowledge-grounded human-human conversation dataset that consists of 11,319 dialogues and 248,014 utterances. **PersonaChat (Zhang et al., 2018)** is human-human persona-conditioned conversations that consists of 10,907 dialogues and 162,064 utterances. Mehri and Eskenazi (2020b) (USR-TC) performs human annotation on 60 dialog contexts, with 6 responses per context for TopicalChat (four system outputs, one newly-annotated human output, one original ground-truth response) and five for PersonaChat (USR-PC). Each response was annotated in terms of *Understandability*, *Naturalness*, *Sensibleness*, *Interesting*, *Uses Knowledge* and *Overall Quality*.

DSTC6 (Hori and Hori, 2017), the 6th Dialog System Technology Challenge, used dialog data collected from multiple Twitter accounts of customer service for its conversation modeling track. Each dialogue consisted of real tweets between a customer and an agent. 40,000 responses are obtained from the competing system, all of which are based on the LSTM Seq2Seq model, which are then annotated in terms of overall quality (DSTC-6).

FED (Mehri and Eskenazi, 2020a) is constructed by annotating 40 Human-Meena conversations, 44 Human-Mitsuku conversations and 40 Human-Human conversations obtained from Adi-

wardana et al. (2020b). The conversations are annotated with 18 subqualities, at the turn and dialogue levels. In this work we use the turn-level overall quality annotations for evaluation (FED).

A.2 Training setup and Hyperparamters

This work's codebase uses AdaterHub [3], which is based on HuggingFace Transformers [4]. We train all Adapters using Adam with a learning rate of 1e-4. Training is conducted for 10 epochs, with a batch size of 16, except for the Fusion training, which we set to 8. We experiment different seeds for the Fusion training, and present the best performing one. The best performing model on the evaluation set is selected for testing. Max sequence length was fixed to 128. The regression head consists of 2 layer MLP with a hidden size of 1024. We use the Hyperbolic tangent as the activation function. We use a single Quadro RTX 6000 24GB GPU for training.

[3] https://Adapterhub.ml/

[4] https://github.com/huggingface/transformers

Graph Neural Network Policies and Imitation Learning for Multi-Domain Task-Oriented Dialogues

Thibault Cordier[*,1,2], **Tanguy Urvoy**[2], **Fabrice Lefèvre**[1], **Lina M. Rojas-Barahona**[2]
[1]LIA - University of Avignon, Avignon, France
[2]Orange Labs, Lannion, France
thibault.cordier@alumni.univ-avignon.fr
fabrice.lefevre@univ-avignon.fr
{thibault.cordier, linamaria.rojasbarahona, tanguy.urvoy}@orange.com

Abstract

Task-oriented dialogue systems are designed to achieve specific goals while conversing with humans. In practice, they may have to handle simultaneously several domains and tasks. The dialogue manager must therefore be able to take into account domain changes and plan over different domains/tasks in order to deal with multi-domain dialogues. However, learning with reinforcement in such context becomes difficult because the state-action dimension is larger while the reward signal remains scarce. Our experimental results suggest that structured policies based on graph neural networks combined with different degrees of imitation learning can effectively handle multi-domain dialogues. The reported experiments underline the benefit of structured policies over standard policies.

Introduction

Task-oriented dialogue systems are designed to achieve specific goals while conversing with humans. They can help with various tasks in different domains, such as seeking and booking a restaurant or a hotel (Zhu et al., 2020). The conversation's goal is usually modelled as a slot-filling problem. The *dialogue manager* (DM) is the core component of these systems that chooses the dialogue actions according to the context. *Reinforcement learning* (RL) can be used to model the DM, in which case the policy is trained to maximize the probability of satisfying the goal (Gao et al., 2018).

We focus here on the multi-domain multi-task dialogue problem. In practice, real applications like personal assistants or chatbots must deal with multiple tasks: the user may first want to **find** a hotel (first task), then **book** it (second task). Moreover, the tasks may cover several domains: the user may want to find a hotel (first task, first domain), book it (second task, first domain), and then find a restaurant nearby (first task, second domain).

One way of handling this complexity is to rely on a *domain hierarchy* which decomposes the decision-making process; another way is to switch easily from one domain to another by scaling up the policy. Although *structured dialogue policies* can adapt quickly from a domain to another (Chen et al., 2020b), covering multiple domains remains a hard task because it increases the dimensions of the state and action spaces while the reward signal remains sparse. A common technique to circumvent this reward scarcity is to guide the learning by injecting some knowledge through a teacher policy[1].

Our main contribution is to study how structured policies like *graph neural networks* (GNN) combined with some degree of *imitation learning* (IL) can be effective to handle multi-domain dialogues. We provide large scale experiments in a dedicated framework (Zhu et al., 2020) in which we analyze the performance of different types of policies, from multi-domain policy to generic policy, with different levels of imitation learning.

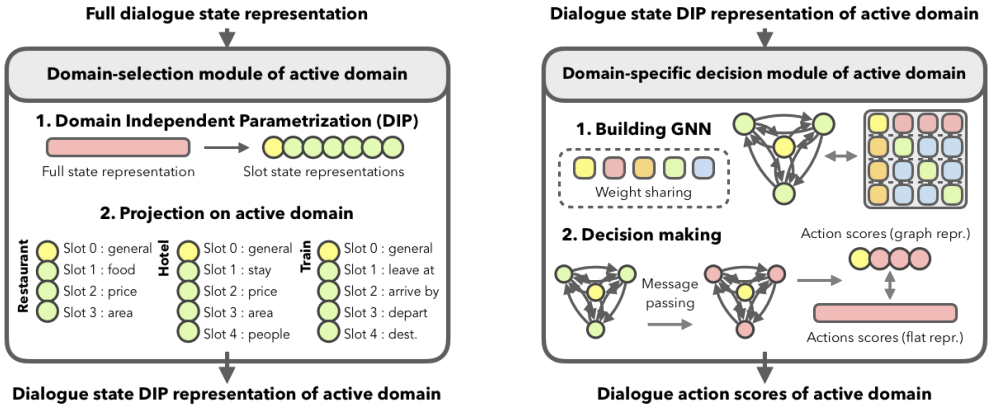
The remainder of this paper is structured as follows. We present the related work in Section 1. Section 2 presents our structured policies combined with imitation learning. The experiments and evaluation are described in Sections 3 and 4 respectively. Finally, we conclude in Section 5.

1 Related Work

Fundamental hierarchical reinforcement learning (Dayan and Hinton, 1993; Parr and Russell, 1998; Sutton et al., 1999; Dietterich, 2000) has inspired a previous string of works on dialogue management (Budzianowski et al., 2017; Casanueva et al., 2018a,b; Chen et al., 2020b). Recently, the use of structured hierarchy with GNN (Zhou et al., 2020; Wu et al., 2020) rather than a set of classical *feed-forward networks* (FNN) enables the learning of non-independent sub-policies (Chen et al., 2018,

[1] For deployment the teacher is expected to be a human expert, however, for experimentation purposes we used the handcrafted policy as a proxy (Casanueva et al., 2017).

Proceedings of the SIGdial 2022 Conference, pages 91–100
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022



(a) Domain-selection module.

(b) Domain-specific decision module.

Figure 1: GNN policy for multi-domain dialogues with hierarchical decision making and weight sharing.

2020a). These works adopted the *Domain Independent Parametrisation* (DIP) that standardizes the slots representation into a common feature space to eliminate the domain dependence. It allows policies to deal with different slots in the same way. It is therefore possible to build policies that handle a variable number of slots and that transfer to different domains on similar tasks (Wang et al., 2015).

Our contribution differs from Chen et al. (2020b) on three points: first we perform our experiments on ConvLab (Zhu et al., 2020) which is a dedicated multi-domain framework; second, the *dialogue state tracker* (DST) output is not discarded when activating the domain; third, we adapt the GNN structure to each domain by keeping the relevant nodes while sharing the edge's weights.

The reward sparsity can be bypassed by guiding the learning through the injection of some knowledge via a teacher policy. This approach, called *imitation learning* (IL) (Hussein et al., 2017), can be declined from pure *behaviour cloning* (BC) where the agent only learns to mimic its teacher to pure *reinforcement learning* (RL) where no hint is provided (Shah et al., 2016; Hester et al., 2018; Gordon-Hall et al., 2020; Cordier et al., 2020).

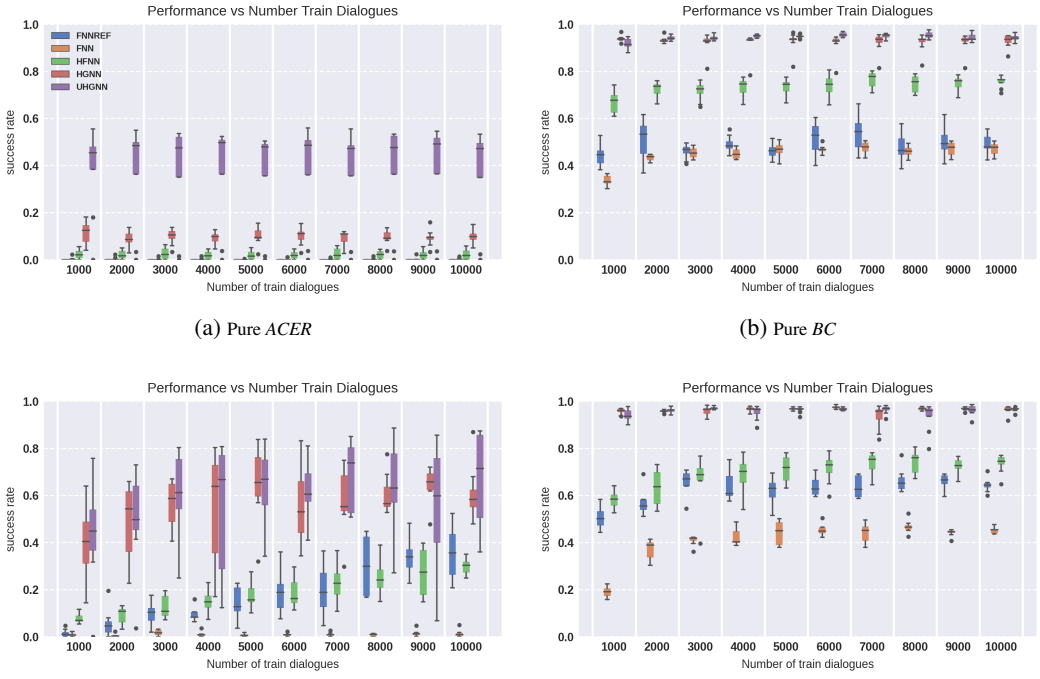
2 Extended GNN Policies with Imitation

We adopt the multi-task setting as presented in ConvLab, in which a single dialogue can have the following tasks: (i) **find**, in which the system requests information in order to query a database and make an offer; (ii) **book**, in which the system requests information in order to book the item. A single dialogue can also contain multiple domains such as *hotel*, *restaurant*, *attraction*, *train*, etc.

Our method, illustrated in Figure 1, is designed to adapt: (i) at the domain-level (*i.e.* be scalable to changes in the number of slots), and (ii) at the multi-domain-level (*i.e.* be scalable to changes of domain). For each dialogue turn, it works as follow: first, the DST module chooses which domain to activate. Then, the multi-domain belief state (and action space) is projected into the active domain (*i.e* only the DIP nodes corresponding to the active domain are kept) as shown in Figure 1a. Afterwards, we apply the GNN message passing as Chen et al. (2020b) but only among the domain specific DIP nodes in the decision making module (Figure 1b).

GNN Policies The GNN structure we consider is a fully connected graph in which the nodes are extracted from the DIP. We distinguish two types of nodes: the slot nodes representing the parametrisation of each slot (denoted as S-node) and the general node representing the parametrisation of the domain (as I-node for slot-Independent node). This yields three types of edges: I2S (for I-node to S-node), S2I and S2S. This abstract structure is a way of modelling the relations between slots as well as exploiting symmetries based on weight sharing (Figure 1b).

Imitation Learning In addition to the structured architecture, we use some level of IL to guide the agent's exploration. In our experiments, we used ConvLab's handcrafted policy as a *teacher* (or *oracle*)[1], but other policies could be used as well. *Behaviour cloning* (BC) is a pure supervised learning method that tries to mimic the teacher policy. Its loss function is the cross-entropy loss as in a classi-



(a) Pure *ACER*

(b) Pure *BC*

(c) *ACER* with *ILfOD*.

(d) *ACER* with *ILfOS*.

Figure 2: Distribution via boxplot of the performance of the proposed approaches on CONVLAB, with 10 different initializations and without pre-training. The coloured area represents the interquartile Q1-Q3 of the distribution, the middle line represents its median (Q2) and the points are outliers.

fication problem. *Imitation Learning From Oracle Demonstrations* (ILFOD) is a RL method which allows the agent to play oracle actions as demonstrations and to inject them in its *replay buffer*. In our experiments, we kept half of the agent's own actions in the buffer along with those generated by the oracle. *Imitation Learning From Oracle Supervision* (ILFOS) is the combination of supervised and reinforcement learning when the agent learns with a supervised loss, namely the margin loss (Hester et al., 2018).

3 Experiments

We performed an ablation study: (i) by progressively extending the baseline to our proposed GNNs and (ii) by guiding the exploration with IL. All the experiments were restarted 10 times with random initialisations and the results evaluated on 500 dialogues were averaged. Each learning trajectory was kept up to 10,000 dialogues with a step of 1,000 dialogues in order to analyse the variability and stability of the methods.

Models The baseline is **ACER** which is a sophisticated actor-critic method (Wang et al., 2016). After an ablation study, we progressively added some notion of hierarchy to FNNs to approximate the structure of GNNs. **FNN** is a feed-forward neural network with DIP parametrisation. Thus, the agent actions are single-actions. **FNN-REF** is a FNN with the native parametrisation (no DIP) with multiple-actions of CONVLAB[2]. **HFNN** is a hierarchical policy with domain-selection module and based on FNNs for each domain. **HGNN** is a hierarchical policy with domain-selection module and based on GNNs. **UHGNN** is a HGNN with a unique GNN for all domains.

Metrics We evaluate the performance of the policies for all tasks. For the find task, we use the precision, the recall and the F-score metrics: the **inform rates**. For the book task, we use the accuracy metric namely the **book rate**. The dialogue is marked as **successful** if and only if both inform's recall and book rate are 1. The dialogue is considered **completed** if it is successful from the user's point of view (*i.e* a dialogue can be completed without being successful if the information provided is not the one objectively expected by the simulator).

[2]The native parametrisation manually groups multi-actions based on MULTIWOZ (Budzianowski et al., 2018).

Configuration			Avg Turn	Inform (%)	Book	Complete		Success	
NLU	Policy	NLG	(succ/all)	Prec. / Rec. / F1	Rate (%)	Rate (%)		Rate (%)	
-	HDC	-	10.6/10.6	87.2 / 98.6 / 90.9	98.6	97.9	-	97.3	-
-	ACGOS (ours)	-	13.1/13.2	94.8 / 99.0 / 96.1	98.7	98.2	(+0.3)	97.0	(-0.3)
BERT	HDC	T	11.4/12.0	82.8 / 94.1 / 86.2	91.5	92.7	-	83.8	-
BERT	HDC†	T	11.6/12.3	79.7 / 92.6 / 83.5	91.1	90.5	(-2.2)	81.3	(-2.5)
BERT	MLE†	T	12.1/24.1	62.8 / 69.8 / 62.9	17.6	42.7	(-50.0)	35.9	(-47.9)
BERT	PG†	T	11.0/25.3	57.4 / 63.7 / 56.9	17.4	37.4	(-55.3)	31.7	(-52.1)
BERT	GDPL†	T	11.5/21.3	64.5 / 73.8 / 65.6	20.1	49.4	(-43.3)	38.4	(-45.4)
BERT	PPO†	T	13.1/17.8	69.4 / 85.8 / 74.1	86.6	75.5	(-17.2)	71.7	(-12.1)
BERT	ACGOS (ours)	T	14.0/14.8	88.8 / 92.6 / 89.5	86.6	89.1	(-3.6)	81.7	(-2.1)

Table 1: Dialogue system evaluation with simulated users. T means template-based NLG. Configurations without NLU and NLG modules pass directly the dialogue act. Configurations with ACGOS and HDC policies are evaluated on a single run with 1,000 dialogues. Configurations with † are taken from the GitHub of ConvLab. PPO in ConvLab used behaviour cloning as the pre-trained weights (see for more details).

4 Evaluation

We evaluate the dialogue manager and the dialogue system both with simulated users.

Dialogue Manager We performed an ablation study based on ACER as reported in Figure 2. First, all RL variants of ACER (Figure 2a) have difficulties to learn without supervision in contrast to BC variants (Figure 2b). In particular, we see that hierarchical decision making networks (HFNN in green), graph neural network (HGNN in red) and generic policy (UHGNN in purple) drastically improve the performance compared to FNNs. Similarly, using IL like ILFOD (Figure 2c) and ILFOS (Figure 2d) notably improves the performance. Therefore, learning generic GNNs allows collaborative gradient update and efficient learning on multi-domain dialogues. Conversely, we observe that hierarchical decision making with HFNNs does not systematically guarantee any improvement. These results suggest that GNNs are useful for learning dialogue policies on multi-domain which can be transferred during learning across domains on-the-fly to improve performance. Finally, regarding ILFOD variants (Figure 2c), we can observe that all architectures are affected by a large variability. This shows that multi-domain dialogue management is difficult despite the use of demonstrations and that learning with reward is not sufficient to robustly succeed.

Dialogue System We evaluate the policy learning algorithms in the entire dialogue pipeline, in particular our best DM policy ACER-ILFOS-UHGNN under a shorter name **ACGOS**. The results of our experimentation are presented in Table 1. We observe that the performance of our approach is closed to the handcrafted policy (the teacher) when directly passing the dialogue acts (97.3 *vs.* 97.0). It is also closed to the handcrafted policy when using BERT NLU (Devlin et al., 2018) and template-based NLG (83.8 *vs.* 81.7). It is much better compared to the baselines with a significant difference (*e.g.* with 81.7 for ACGOS *vs.* 71.7 for pre-trained PPO). These results highlight the benefit of structured policies against standard policies.

5 Conclusion

We studied structured policies like GNN combined with some imitation learning that effectively handle multi-domain dialogues. The results of our large-scale experiments on ConvLab confirm that an actor-critic based policy with a GNN structure can solve multi-domain multi-task dialogue problems. Finally, we evaluated our best policy (ACGOS) in a complete dialogue system with simulated users. It overcomes the baselines and it is comparable to the handcrafted policy.

A limitation of current policies in ConvLab, including ours, is that the robustness to noisy inputs is not specifically addressed as it had been done in PyDial (Ultes et al., 2017). It could be also interesting to study the impact of incorporating real human feed-backs and demonstrations instead of a handcrafted teacher.

The GNN structured policies combined with imitation learning avoid sparsity, while being data efficient, stable and adaptable. They are relevant for covering multi-domain task dialogue problems.

References

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Paweł Budzianowski, Stefan Ultes, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Inigo Casanueva, Lina Rojas-Barahona, and Milica Gašić. 2017. Subdomain modelling for dialogue management with hierarchical reinforcement learning. *arXiv preprint arXiv:1706.06210*.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*.

Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Stefan Ultes, Lina M Rojas Barahona, Bo-Hsiang Tseng, and Milica Gasic. 2018a. Feudal reinforcement learning for dialogue management in large domains. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 714–719.

Iñigo Casanueva, Paweł Budzianowski, Stefan Ultes, Florian Kreyssig, Bo-Hsiang Tseng, Yen-Chen Wu, and Milica Gasic. 2018b. Feudal dialogue management with jointly learned feature extractors. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 332–337.

Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Rojas-Barahona, Steve Young, and Milica Gašić. 2017. A Benchmarking Environment for Reinforcement Learning Based Task Oriented Dialogue Management. *arXiv:1711.11023 [cs, stat]*. ArXiv: 1711.11023.

Lu Chen, Bowen Tan, Sishan Long, and Kai Yu. 2018. Structured dialogue policy with graph neural networks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1257–1268.

Zhi Chen, Lu Chen, Xiaoyuan Liu, and Kai Yu. 2020a. Distributed structured actor-critic reinforcement learning for universal dialogue management. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2400–2411.

Zhi Chen, Xiaoyuan Liu, Lu Chen, and Kai Yu. 2020b. Structured hierarchical dialogue policy with graph neural networks. *arXiv preprint arXiv:2009.10355*.

Thibault Cordier, Tanguy Urvoy, Lina M Rojas-Barahona, and Fabrice Lefèvre. 2020. Diluted near-optimal expert demonstrations for guiding dialogue stochastic policy optimisation. In *Human in the loop dialogue systems Workshop at 34th Conference on Neural Information Processing Systems*.

Peter Dayan and Geoffrey E Hinton. 1993. Feudal reinforcement learning. In *Advances in neural information processing systems*, pages 271—-278.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Thomas G Dietterich. 2000. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374.

Gabriel Gordon-Hall, Philip Gorinski, and Shay B Cohen. 2020. Learning dialog policies from weak demonstrations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1394–1405.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. 2018. Deep q-learning from demonstrations. In *Thirty-second AAAI conference on artificial intelligence*.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. 2017. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35.

Ronald Parr and Stuart Russell. 1998. Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, pages 1043–1049.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pararth Shah, Dilek Hakkani-Tur, and Larry Heck. 2016. Interactive reinforcement learning for task-oriented dialogue management.

Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, 112(1-2):181–211.

Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien

Wen, and Milica Gasic. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78.

Zhuoran Wang, Tsung-Hsien Wen, Pei-Hao Su, and Yannis Stylianou. 2015. Learning domain-independent dialogue policies via ontology parameterisation. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 412–416.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. 2016. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. Convlab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149.

A Appendix

A.1 Domains

Domain	# constraint slots	# request slots
CONVLAB	find/book	search
Restaurant	4/3	5
Attraction	3/-	7
Hotel	7/3	5
Taxi	4/-	2
Train	5/1	5
Hospital	1/-	3
Police	-/-	3

Table 2: Domains Description of CONVLAB framework

Belief State The belief state representation is deterministic. As shown in Figure 3, there is no uncertainty (all values are either 0's or 1's).

State Space The input to the dialogue manager is the belief state which is a dictionary of all tractable information (slot-value pairs, history, dialogue actions of system and user, etc.). This is called the *master state space*. And, due to its large size, the representation is projected into the *summary state space* by a process called *value abstraction* (Wang et al., 2015). Finally, it must be vectorised in order to be interpretable by neural networks.

Action Space The dialogue manager's output is a probabilistic distribution over all possible actions. To reduce the complexity of the learning problem, *master actions*, which are valued dialogue acts such as INFORM(date = '2022-01-15'), are abstracted into *summary actions* like INFORM(date), the *value abstraction* module being in charge of restoring the relevant values in the context. On CONVLAB the policy may activate several actions simultaneously (called *multiple-actions*).

Domain Independent Parametrisation (or DIP) (Wang et al., 2015) standardises the slots representation into a common feature space to eliminate the domain dependence. In particular, the DIP state and action representations are not reduced to a flat vector but to a set of sub-vectors: one corresponding to the domain parametrisation (called *slot-independent representation*), the others to the slots parametrisation (called *slot-dependent representations*).

Component / Description
Beliefs
constraint slot beliefs: $\{b_{d,s}^{inf} \in \mathcal{V}_s, \forall s \in \mathcal{S}_d^{inf}, \forall d \in \mathcal{D}\}$ The goal constraints belief for each informable slot. This is either an assignment of a value from the ontology which the user has specified as a constraint, or has a special value — either *dontcare* which means the user has no preference, or *none* which means the user is yet to specify a valid goal for this slot. To be exact, for each domain, the constraint slot dictionary separates slots with respect to the task i.e we distinguish the *find* slot dictionary and the *book* slot dictionary.
request slot beliefs: $\{b_{d,s}^{req} \in \mathbb{B}, \forall s \in \mathcal{S}_d^{req}, \forall d \in \mathcal{D}\}$: A set of requested slots, i.e. those slots whose values have been requested by the user, and should be informed by the system.
Features
terminated: $f_1 \in \mathbb{B}$: A boolean showing that the user wants to end the call.
booked: $f_2 \in \mathcal{V}_{DB(d)}$: The name of the last venue offered by the system to the user with respect to the constraint slots with additional information like reference. To be exact, this feature is located in the *book* slot dictionary.
degree pointer: $f_3 \in \mathbb{B}^6$: The vector counting the number of entities *count* matching with constraint slots in acceptance list: [count==0, count==1, count==2, count==3, count==4, count>=5].
System Acts
system acts: $a^{sys} \in list(\mathcal{A}^{sys})$: The list of the last system actions.
User Acts
user acts: $a^{user} \in list(\mathcal{A}^{user})$: The list of the last user actions.

Table 3: Belief State Template in ConvLab framework

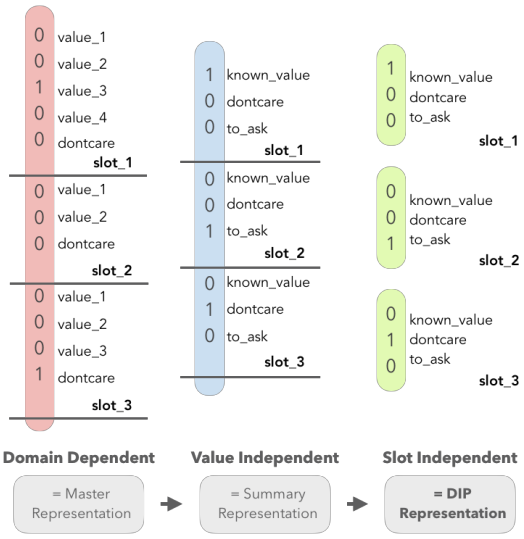


Figure 3: Transformation from initial state to DIP state representation (it works similarly for actions).

A.2 State and Action Representations

We propose to formally present the state representations used in our experiments. For details about our notations, see Table 3.

Flat state representation in ConvLab

$$\phi(x) = \left(\bigoplus_{s \in \mathcal{S}^{inf}} b_s^{inf} \right) \oplus a^{user} \oplus a^{sys} \oplus [f_1] \oplus f_2 \oplus f_3$$

where x is the initial state, $\phi(x)$ is the full state parametrisation, $\mathcal{S}^{inf}$ is the set of informable slots, b_s^{inf} is the one-encoding vector of the informable slot s, a^{user} and a^{sys} are the one-encoding vectors of previous user and system actions, f_1 is the boolean "terminated dialogue", f_2 is the boolean "booked offer" with respect to each domain, f_3 is the one-encoding vector of the matching entities count with respect to each domain and $\oplus$ is the vector concatenation operator.

DIP state representation

Slot independent parametrisation:

$$\phi_d(x) = a^{user}|_g \oplus a^{sys}|_g \oplus [f_1,\ f_2|_d,\ f_3|_d]$$

where x is the initial state, $\phi_d(x)$ is the active domain state parametrisation, $a^{user}|_g$ and $a^{sys}|_g$ are the one-encoding vectors of previous general user and system actions, f_1 is the boolean "terminated dialogue", $f_2|_d$ is the boolean "booked offer" with respect to the active domain, $f_3|_d$ is the one-encoding vector of the matching entities count with respect to the active domain and $\oplus$ is the vector concatenation operator.

Slot dependent parametrisation:

$$\forall s_i \in \mathcal{S}_d,\ \phi_{s_i}(x) = a^{user}|_{s_i} \oplus a^{sys}|_{s_i}$$
$$\oplus \left[\mathbb{1}(\exists\, v \in \mathcal{V}_{s_i}/\{\text{none}\},\ b^*_{s_i}[v] = 1)\right] \quad (2a)$$
$$\oplus \left[\mathbb{1}(s_i \in \mathcal{S}_d^{inf})\right] \quad (2b)$$
$$\oplus \left[\mathbb{1}(s_i \in \mathcal{S}_d^{req})\right] \quad (2c)$$

where x is the initial state, $\phi_{s_i}(x)$ is the slot parametrisation of the i^{th} slot, $\mathcal{S}_d$ is the set of slots of the active domain, $a^{user}|_{s_i}$ and $a^{sys}|_{s_i}$ are the one-encoding vectors of previous user and system actions of the i^{th} slot, (2a) is the indicator of known value, (2b) is the indicator of informable slot and (2c) is the indicator of requestable slot and $\oplus$ is the vector concatenation operator.

A.3 Implementation Details

Imitation learning The used oracle is the hand-crafted agent proposed by each framework. When we use ILFOD or ILFOS methods, 50% of the time the oracle trajectories is used. When we use ILFOS, we call also in 100% of the time the oracle which gives us the best expert action as supervision and a margin penalty $\mu = \log(2)$ (Hester et al., 2018).

Reinforcement learning Our policy algorithm is an off-policy learning that uses experience replay (all data are stored in buffers) without priority *i.e* without importance sampling. The exploitation-exploration procedure is achieved by Boltzmann sampling with a fixed temperature $\tau = 1$.

Metrics and Rewards **Inform recall** evaluates whether all the requested information has been informed when **inform precision** evaluates whether only the requested information has been informed. **Book rate** assesses whether the offered entity meets all the constraints specified in the user goal. The system is guided by the rewards as follows. If all domains are solved (a domain is solved if all related tasks are solved), it gains 40 points. If the current active domain is solved, it gains 5 points. Otherwise, it is penalised by 1 point.

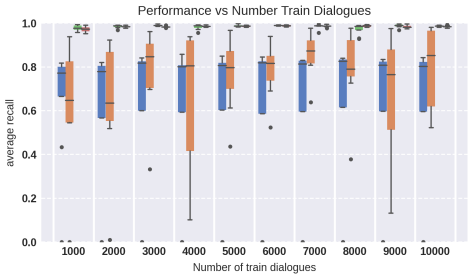
Model setup for neural network architectures Our FNN models have two hidden layers, both with 128 neurons. Our GNN models have one first hidden layer with 32 neurons for each node (two in all: S-NODE and I-NODE). Then the second hidden layer is composed of 32 neurons for each relation (three in all: S2S, S2I and I2S). The size of the tested networks are of the order of magnitude of $10\,000$ to more than $100\,000$ parameters.

For learning stage, we use a learning rate $\mathrm{lr} = 10^{-3}$, a dropout rate $\mathrm{dr} = 0.1$ and a batch size $\mathrm{bs} = 64$. Each loss function has a weight of $\lambda_Q = 0.5$, $\lambda_\pi = 1.$, $\lambda_{IL} = 1.$ and $\lambda_{ent} = 0.01$ respectively. The learning frequency is one iteration after each episode (finished dialogue) with only one gradient iteration.

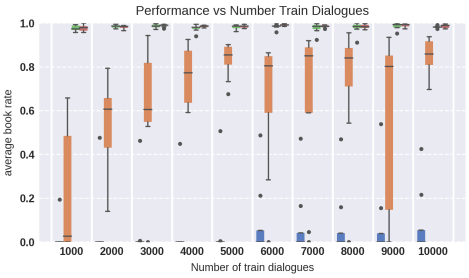
Used packages for the experiment We used the dialogue system frameworks named CONVLAB (Zhu et al., 2020). For the implementation of neural networks, we used PYTORCH (Paszke et al., 2019) in our dialogue systems. We also used another toolkit for reinforcement learning research named OPENAI GYM (Brockman et al., 2016).

A.4 Supplementary Results

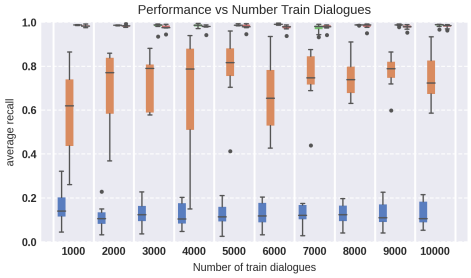
We propose to present supplementary results of our ablation study. We show the distribution (via boxplot) of different measures with 10 different initialisations and without pre-training. In particular, Figure 4 presents the distribution of inform recall, Figure 5 the distribution of book rate, Figure 6 the distribution of success rate and Figure 7 the distribution of cumulative rewards. We precise that the coloured area represents the interquartile Q1-Q3 of the distribution, the middle line represents its median (Q2) and the points are outliers.



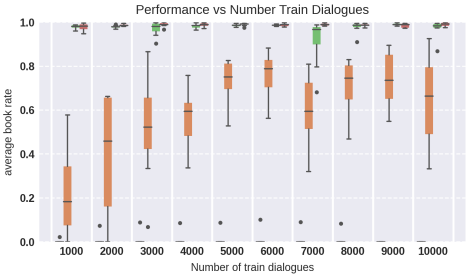
(a) Recall Average - UHGNN models



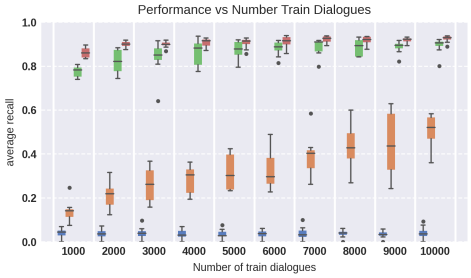
(a) Book Rate - UHGNN models



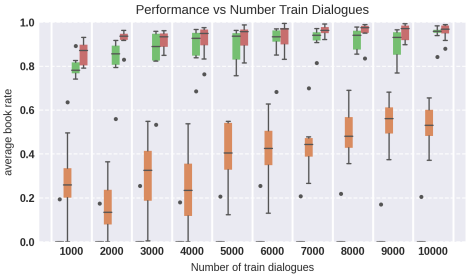
(b) Recall Average - HGNN models



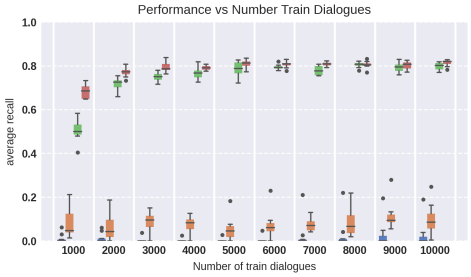
(b) Book Rate - HGNN models



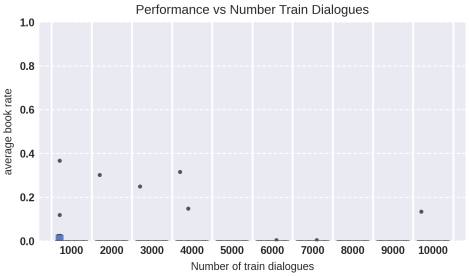
(c) Recall Average - HFNN models



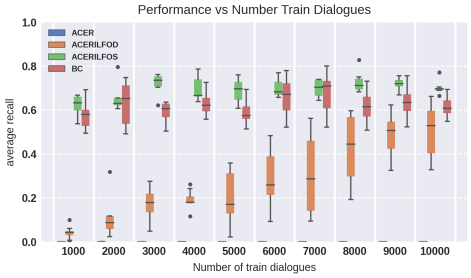
(c) Book Rate - HFNN models



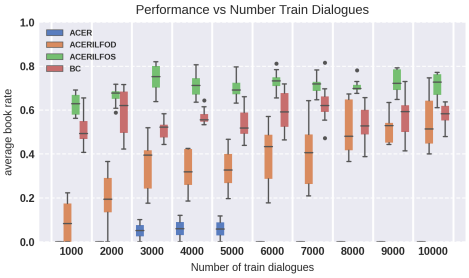
(d) Recall Average - FNN models with DIP parametrization



(d) Book Rate - FNN models with DIP parametrization



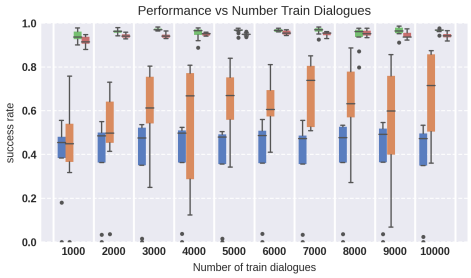
(e) Recall Average - FNN models with native parametrization



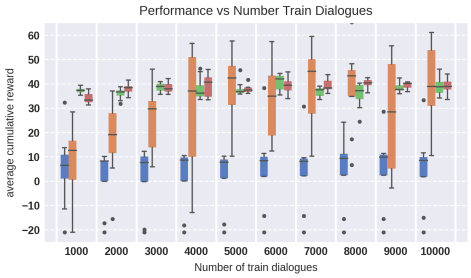
(e) Book Rate - FNN models with native parametrization

Figure 4: Summary of performance - Task *find*

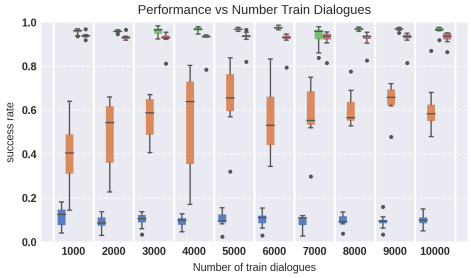
Figure 5: Summary of performance - Task *book*



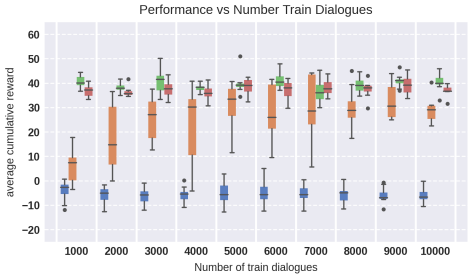
(a) Success Rate - UHGNN models



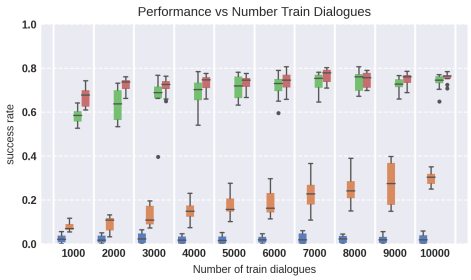
(a) Cumulative rewards - UHGNN models



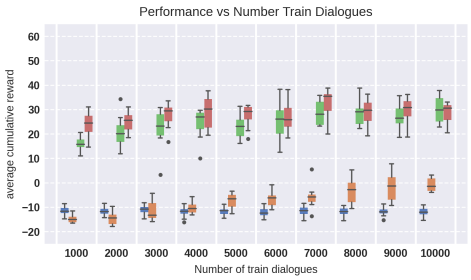
(b) Success Rate - HGNN models



(b) Cumulative rewards - HGNN models

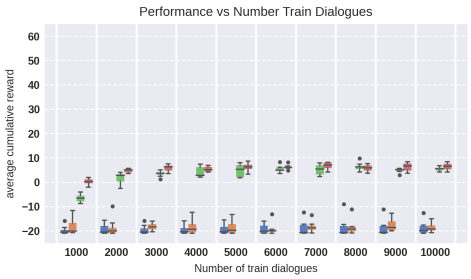


(c) Success Rate - HFNN models

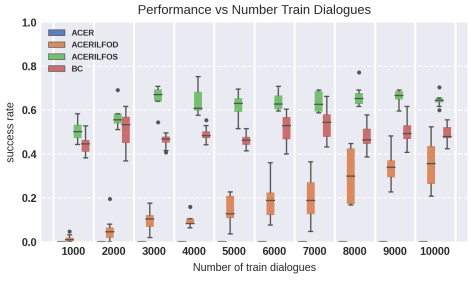


(c) Cumulative rewards - HFNN models

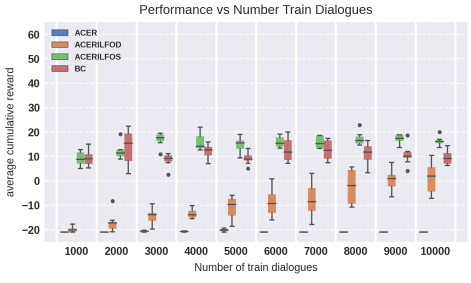
(d) Success Rate - FNN models with DIP parametrization



(d) Cumulative rewards - FNN models with DIP parametrization



(e) Success Rate - FNN models with native parametrization



(e) Cumulative rewards - FNN models with native parametrization

Figure 6: Summary of performance - Global task (Task *find* and/or Task *book*)

Figure 7: Summary of performance - Cumulative rewards

The DialPort tools

Jessica Huynh* and **Shikib Mehri*** and **Cathy Jiao*** and **Maxine Eskenazi**
Carnegie Mellon University
jhuynh, amehri, cljiao, max@cs.cmu.edu

Abstract

The DialPort project (http://dialport.org/), funded by the National Science Foundation (NSF), covers a group of tools and services that aim at fulfilling the needs of the dialog research community. Over the course of six years, several offerings have been created, including the DialPort Portal and DialCrowd. This paper describes these contributions, which will be demoed at SIGDIAL, including implementation, prior studies, corresponding discoveries, and the locations at which the tools will remain freely available to the community going forward.

1 Introduction

The DialPort project[1] has created tools and services that respond to needs voiced by many in the dialog research community during several workshops organized by the Principle Investigators (PIs). Its offerings are available at no cost to the community with the goal of helping researchers gather high quality data, and easily assess and compare their dialog systems. This paper and its corresponding demos showcase the DialPort Portal[2] and DialCrowd[3].

There is an increasing need for large amounts of natural dialog data that can be obtained at reasonable cost and in an interactive manner. Static datasets are ineffective for both evaluation and optimization. This has led to the creation of the DialPort Portal, which facilitates the collection of flexible and evolving data as well as interactive assessment with real users. Notably, the Portal was used to connect systems and collect data for the *Interactive Evaluation of Dialog* track (Mehri et al., 2021) at DSTC9 (Gunasekara et al., 2020).

Another community need centers around how to gather high quality data when using crowdsourcing platforms. DialCrowd has been constructed to facilitate crowdsourcing by guiding researchers to give clear, understandable explanations of the task to the workers who produce or annotate data. It also aids in calculating the correct level of worker payment. Finally, it includes several methods of data quality assessment.

The University of Southern California (USC) is a partner in DialPort. The team at USC works on a tools repository[4] and the REAL Challenge.

This paper gives background and describes in detail the parts of both the Portal and DialCrowd. It also provides information on how to access and use them. As the DialPort project draws to an end, the paper indicates the permanent sites where these tools will reside.

2 Background

2.1 Interactive Platforms for Dialog

As dialog models improve, it is imperative that they are evaluated in interactive settings with real users. Mehri and Eskenazi (2020) show that while pre-trained dialog systems excel at generating responses (Zhang et al., 2019; Bao et al., 2020), they underperform in back-and-forth interactions.

The Alexa Prize challenge (Ram et al., 2018; Khatri et al., 2018) allows university teams to build socialbots that are assessed in interactive settings with Alexa users. In contrast, the DialPort Portal is accessible to the broader research community. Furthermore, the Alexa Prize challenge primarily relies on speech input from the user, which may result in speech recognition errors. Though the DialPort Portal can accept speech input, its web interface can also be used with text-only input.

*Equal contribution
[1] http://dialport.org/
[2] https://dialport.org/portal
[3] http://dialport.org/dialcrowd.html
[4] https://dialport.ict.usc.edu/

Proceedings of the SIGdial 2022 Conference, pages 101–106
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

2.2 Crowdsourcing

With the amount of dialog data available or able to be collected with systems such as DialPort, it is important to have easy and accessible tools to create detailed annotations of this data for different metrics. One method of obtaining annotations is crowdsourcing with platforms such as Amazon Mechanical Turk (AMT). However, it is sometimes difficult to obtain conclusive results, and a survey of current natural language processing HITs has shown the weaknesses of these HITs (Huynh et al., 2021). Instructions (Chandler et al., 2013), examples (Doroudi et al., 2016), and payment are some of the aspects that need to be attended to in order for HITs to acquire higher quality data.

3 DialPort Portal

The DialPort Portal was initially conceived with the objective of listing many dialog systems from a variety of sites. This type of platform, with demonstrations, links, and references to various systems, is valuable to both researchers and real users. The concept of the Portal evolved, and the different systems were linked such that a user could interact with all of the connected systems, transitioning seamlessly between systems, with the dialog state (consisting of slots such as city or date) shared across systems (Zhao et al., 2016; Lee et al., 2017). As dialog systems continued to improve, especially with the advent of engaging response generation models (Zhang et al., 2019; Bao et al., 2020), the Portal recruited real users through Facebook advertising with the objective of providing researchers with a platform to collect interactive dialogs with real users (Mehri et al., 2021).

3.1 Portal Version 1

The original version of the Portal grouped several dialog systems from different sites (Cambridge, USC, CMU) and managed seamless switching amongst (Zhao et al., 2016). For example, a user could ask for the weather in Pittsburgh and get the CMU weather system, then ask the CMU system for the weather in Cambridge, then ask for a restaurant and automatically switch to the Cambridge restaurant system, then ask to play a game and get the USC system.

This instance of the Portal serves as a platform to interact with different systems over the course of one dialog (Zhao et al., 2016). To accomplish this, the Portal needed to address several challenges (1) how to share information across systems (e.g., remembering the city the user wanted the weather for when interacting with the CMU system, and sharing that with the Cambridge system when the user wants a restaurant recommendation), (2) how to gracefully continue a dialog when a system is down, and (3) how to give two systems addressing the same task (e.g., restaurants) equal time with the users. Respectively, these problems were addressed by (1) maintaining a shared dialog state across systems, (2) backing off to an equivalent system or changing the topic, and (3) a pseudo-random system selection policy. In order to make the system easy to use, an API was developed to facilitate connecting new systems to the Portal. This version has pedagogical value as it can easily be demonstrated for dialog classes.

3.2 Portal Version 2

With the advent of the API, the possibilities of use of the Portal greatly expanded. The Portal was used for the DSTC9 Challenge (Mehri et al., 2021), as a tool that enabled researchers to both compare their systems on one common platform (with real users) and to gather considerable amounts of data. The Portal was made available to DSTC9 participants. The idea was to connect systems and have them tested by real (unpaid) users. The CMU DialPort team advertised the Portal on Facebook and interested individuals tried it out (with text only). Upon visiting the Portal, real users are randomly matched with a dialog system, without knowledge of the specific system they are interacting with. While some people left the site after only one or two turns with a system, many actually continued to communicate with a system for a substantial conversation, and were thus considered to be real users. Real users consist of users who find some personal interest (getting information, companionship, curiosity) in continuing a dialog. There were 11 participants in the interactive part of the Challenge (Mehri et al., 2021). With an advertising budget of $2500, we collect more than 4000 dialogs on the DialPort portal (2960 dialogs with at least 4 turns or 8 utterances); thus the cost was less than $1.00 per usable dialog. The DialPort portal, through funding from the National Science Foundation, has been able to provide interactive evaluation as a service free of charge to any dialog researchers. The Appendix contains a sample dialog from the winning system of the DSTC9 track (Bao et al., 2020).

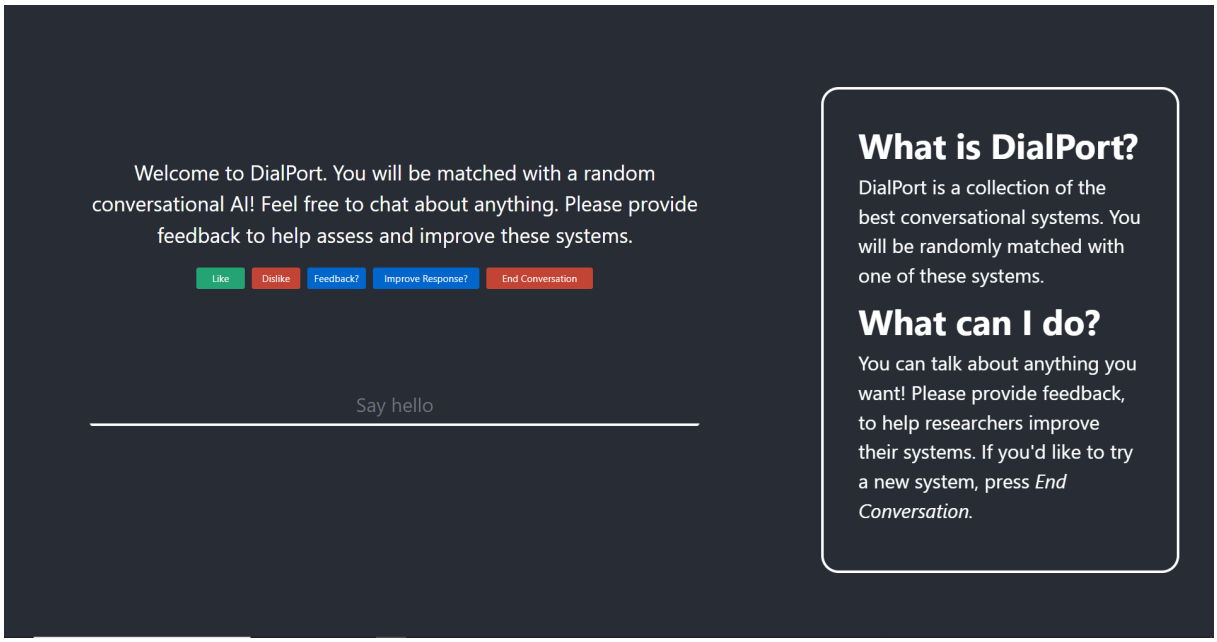


Figure 1: DialPort Portal. This screenshot of the Portal displays (1) the dialog history, shown in the center of the screen, (2) an input field for the user to type their responses, and (3) a set of feedback buttons below the dialog history ("Like", "Dislike", "Feedback?", "Improve Response?" and "End Conversation"). The interface clear and emphasizes the three important actions that a user should perform while using the Portal: (1) reading the dialog history, (2) responding to the dialog system, and (3) providing feedback.

DSTC9 demonstrated that the Portal could easily be used to both compare systems and to gather data with real users. Besides challenges, another potential use of the Portal would be for students to connect systems that they build for a class project to see how well they do in real user interaction.

At the end of the DialPort project in the coming year, the Portal will move from the Dialog Research Center at CMU to LDC at UPenn.

3.3 DialPort Dashboard

After collecting data from real users on the DialPort interface, a subsequent task is to perform analysis on the gathered data. We provide the DialPort dashboard which allows researchers to (i) *analyze* dialogs collected on their system, (ii) *interact* with the dashboard to filter and organize dialogs based on various criteria, and (iii) *compare* their system to other systems connected to the DialPort Portal. Currently, the Dashboard contains over 7000 dialogs from 28 systems. The Dashboard is connected to the DialPort Portal via API calls, allowing dialogs to be quickly displayed on the Dashboard after being collected from the Portal. The Dashboard code will soon be released, allowing for use of the Dashboard in offline mode.

The Dashboard UI contains panels, tables, and charts. At both the system and dialog level, attributes such as the number of utterances, likes, dislikes, comments, corrections are displayed (see figure 3). In addition, the two evaluation metrics of FED (Mehri and Eskenazi, 2020) and human ratings are shown. Since the Dashboard is designed to be easily extended, additional metrics can be added in the future. Users can interact with the dashboard by filtering and ranking dialogs based on attributes and metrics. For example, the provided toolbar can be used to find all conversations with a given user's system with more than n turns or rank conversations from most-to-least number of likes. Users can also filter words and phrases in dialogs by their number of occurrences from the perspective of both the system or human participant, and thus view common phrases or words mentioned on either side of the conversation. Finally, each system contains a progress monitor graph which displays the number of dialogs being collected over time, allowing users to actively observe data collection in the DialPort Portal.

4 DialCrowd

To address the many issues that present themselves when using crowdsourcing to collect high quality data, DialCrowd was created. DialCrowd (Lee

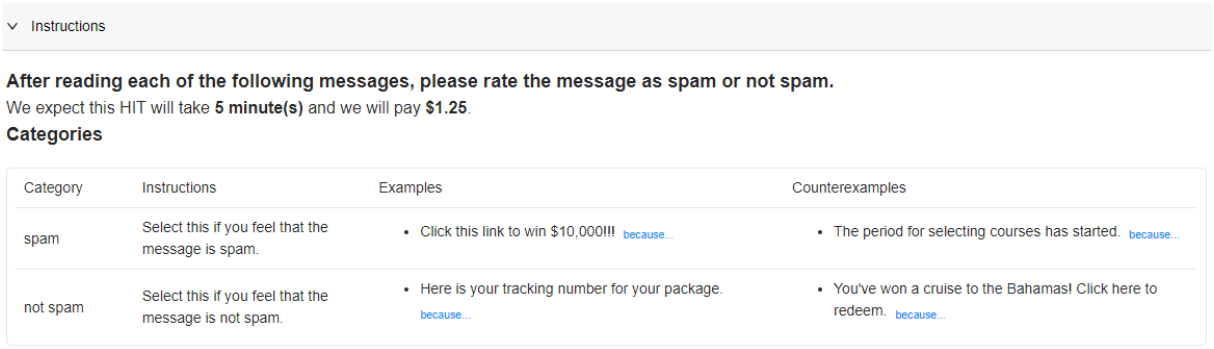


Figure 2: DialCrowd Examples and Counterexamples with Explanations

et al., 2018) is a dialog assessment toolkit which aids researchers with human intelligence task (HIT) creation. Requesters follow templates on the DialCrowd site, which generate a HIT that can be linked for a worker on any crowdsourcing site.

The second version of this tool (Huynh et al., 2022) focuses on collecting high-quality data with tools such as:

- Links to create better instructions
- Prompts to provide examples and counterexamples with explanations seen in Figure 2
- Functionality for adding golden data and duplicate data in each HIT
- Payment suggestions
- A feedback area
- Overall statistics from the HIT (time, patterns in the responses, inter-annotator agreement)

This allows for requesters to create a well-structured HIT which allows workers to provide better quality annotations. Consequently, it makes it easier to filter responses from potential bots. Additional tools include the capability to include a mandatory consent form at the start of the HIT, and detailed style changes for the HIT. Further description of the system along with corresponding images can be found in (Huynh et al., 2022).

One DialCrowd template, intent classification, has been merged into the new home for DialCrowd, ParlAI [5], and is now available for use.

5 The DialPort demo

The demos of the DialPort Portal and Dashboard and of DialCrowd at SIGDIAL will include:

- how to connect a system
- what interaction with each tool looks like
- advantages there are in using the tools, with examples (for example, what resulting data looks like)

6 Conclusion and Future Directions

The tools presented in this demo help dialog researchers in data gathering and assessment. As the community uses them, more types of applications will arise. The tools have been created in a way that enable additions as the field and the needs evolve.

7 Acknowledgements

This work is funded by National Science Foundation Grant Nos. CNS-1512973, DGE1745016, and DGE2140739. The opinions expressed in this paper do not necessarily reflect those of the National Science Foundation. The authors would like to thank Tiancheng Zhao, Kyusong Lee, and Ting-Rui Chiang for their contributions to these tools.

References

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2020. Plato-2: Towards building an open-domain chatbot via curriculum learning. *arXiv preprint arXiv:2006.16779*.

Jesse Chandler, Gabriele Paolacci, and Pam Mueller. 2013. Risks and rewards of crowdsourcing marketplaces. In *Handbook of human computation*, pages 377–392. Springer.

Shayan Doroudi, Ece Kamar, Emma Brunskill, and Eric Horvitz. 2016. Toward a learning science for complex crowdsourcing tasks. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 2623–2634.

[5] https://github.com/facebookresearch/ParlAI/tree/main/parlai/crowdsourcing/tasks/dialcrowd

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, et al. 2020. Overview of the ninth dialog system technology challenge: Dstc9. *arXiv preprint arXiv:2011.06486*.

Jessica Huynh, Jeffrey Bigham, and Maxine Eskenazi. 2021. A survey of nlp-related crowdsourcing hits: what works and what does not. *arXiv preprint arXiv:2111.05241*.

Jessica Huynh, Ting-Rui Chiang, Jeffrey Bigham, and Maxine Eskenazi. 2022. Dialcrowd 2.0: A quality-focused dialog system crowdsourcing toolkit. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1256–1263, Marseille, France. European Language Resources Association.

Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, et al. 2018. Advancing the state of the art in open domain dialog systems through the alexa prize. *arXiv preprint arXiv:1812.10757*.

Kyusong Lee, Tiancheng Zhao, Alan W Black, and Maxine Eskenazi. 2018. Dialcrowd: A toolkit for easy dialog system assessment. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 245–248.

Kyusong Lee, Tiancheng Zhao, Yulun Du, Edward Cai, Allen Lu, Eli Pincus, David Traum, Stefan Ultes, Lina M Rojas Barahona, Milica Gasic, et al. 2017. Dialport, gone live: an update after a year of development. In *Proceedings of the 18th annual SIGdial meeting on discourse and dialogue*, pages 170–173.

Shikib Mehri and Maxine Eskenazi. 2020. Unsupervised evaluation of interactive dialog with dialogpt. *arXiv preprint arXiv:2006.12719*.

Shikib Mehri, Yulan Feng, Carla Gordon, Seyed Hossein Alavi, David Traum, and Maxine Eskenazi. 2021. Interactive evaluation of dialog track at dstc9.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2016. Dialport: Connecting the spoken dialog research community to real user data. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 83–90. IEEE.

A Appendix

User: Hello
System: hello! hows your day going so far?
User: What are you up to?
System: just finished making coffee! what about you? what do you do?
User: i'm playing video games
System: oh, nice! what kind? and i'm just relaxing here and listening to some classical music.
User: i'm playing league of legends
System: nice! what rank are you? i haven't played in forever.
User: who is your favorite classical musician?
System: probably beethoven. i like classical music.
User: i'm plat 2
System: nice! i'd imagine you are better than that!
User: can you play an instrument?
System: i can play the piano. do you play any instruments?
User: thanks
System: i appreciate it! so what is it like being plat 2?
User: i can play the clarinet, but not very well haha
System: oh nice! i played clarinet in high school.
User: the games are difficult but rewarding
System: yes they are! i played the trumpet in college.
. . .

Table 1: An example of a conversation snippet from the DSTC9 top performing system. The system is capable of maintaining two simultaneous threads of dialog about League of Legends and music, and switch between them in a coherent and engaging manner.

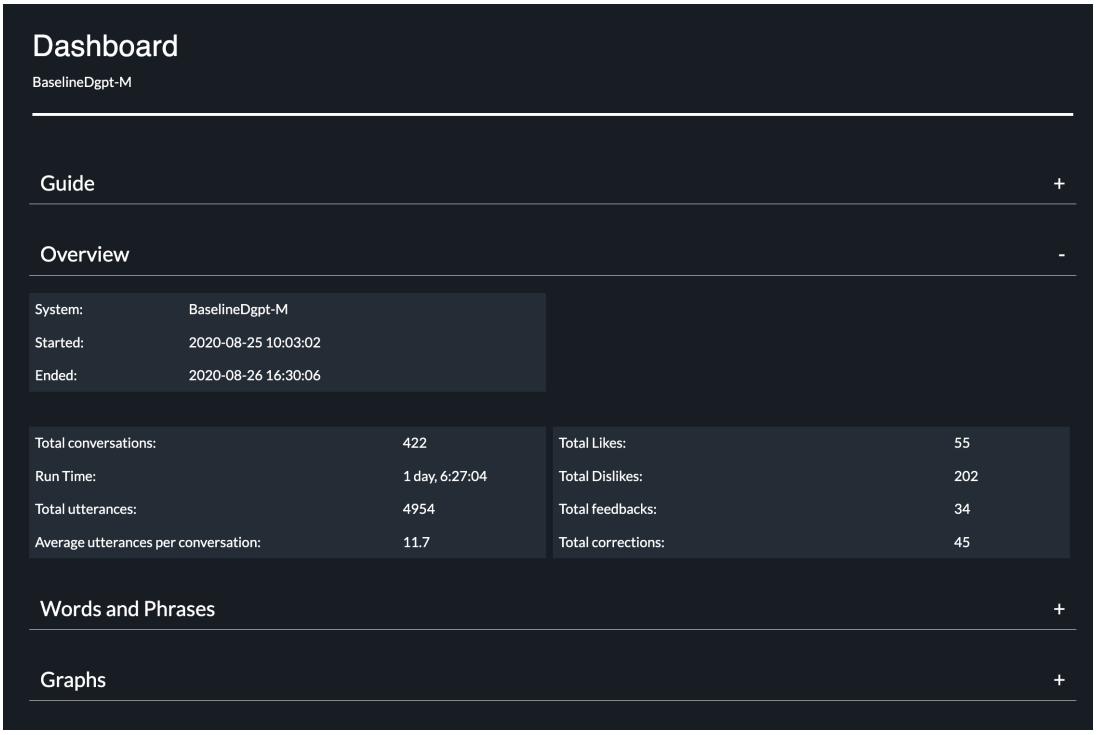


Figure 3: The home page for a system on the DialPort dashboard. General information about the conversations collected from the system are displayed. Sections such as "Words and Phrases" and "Graphs" can be expanded or collapsed to view additional information about the system.

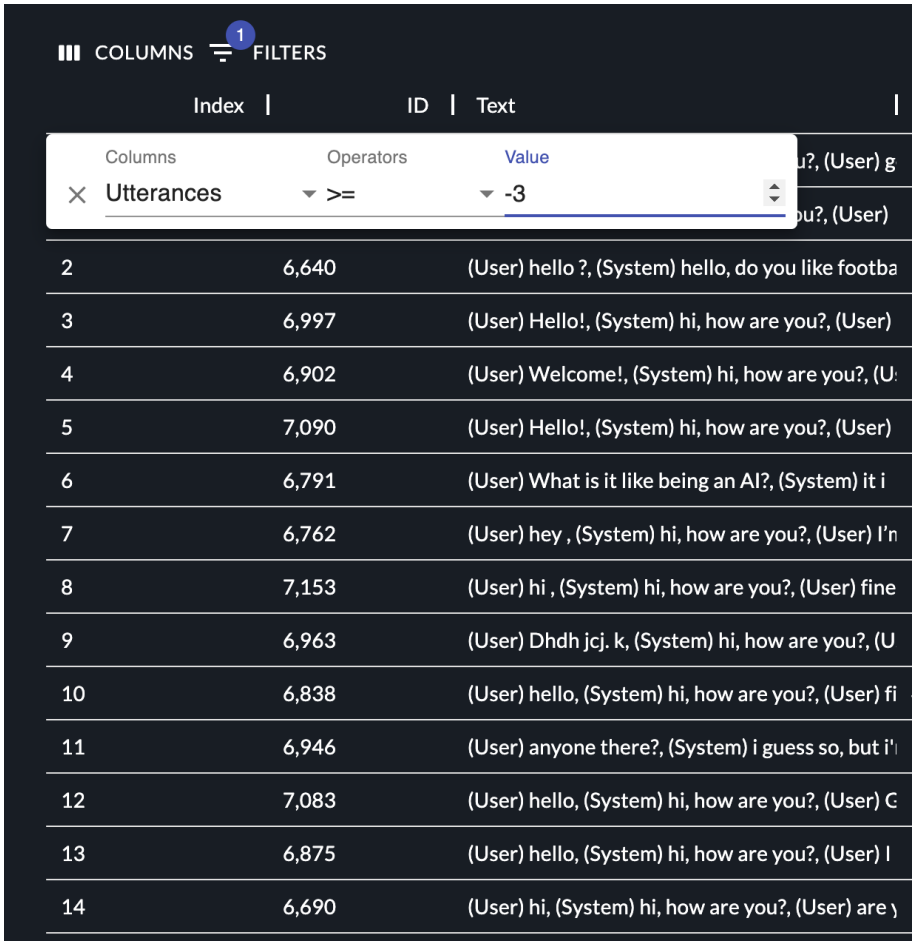


Figure 4: Using the DialPort dashboard to find all conversations in a system with more than 3 utterances

Simultaneous Job Interview System Using Multiple Semi-autonomous Agents

Haruki Kawai, Yusuke Muraki, Kenta Yamamoto,
Divesh Lala, Koji Inoue, and **Tatsuya Kawahara**
Graduate School of Informatics, Kyoto University, Japan
[kawai,muraki,yamamoto,lala,inoue,kawahara]
@sap.ist.i.kyoto-u.ac.jp

Abstract

In recent years, spoken dialogue systems have been used in job interviews where an applicant talks to a system that asks pre-defined questions, called on-demand and self-paced job interviews. We propose a simultaneous job interview system, where one interviewer can conduct one-on-one interviews with multiple applicants simultaneously by cooperating with multiple autonomous interview dialogue systems. However, it is challenging for interviewers to monitor and understand all parallel interviews done by the autonomous system simultaneously. To address this issue, we implement two automatic dialogue understanding functions: (1) response evaluation of each applicant's responses and (2) keyword extraction for a summary of the responses. In this system, interviewers can intervene in a dialogue session when needed and smoothly ask a proper question that elaborates the interview. We have conducted a pilot experiment where an interviewer conducted simultaneous job interviews with three candidates.

1 Introduction

Owing to the widespread use of online job interviews during the COVID-19 situation, spoken dialogue systems supporting job interviews to make them more efficient are being investigated. In conventional face-to-face job interviews, interviewers conducted interviews with many applicants one by one, which was time-consuming. Therefore, on-demand interviews have been widely adopted as an alternative to face-to-face interviews, such as *Hirevue* [1] and *Modern Hire* [2]. In this style, job applicants answer predefined typical questions and then submit video recordings of interviews. However, there is a lack of the much needed interaction between interviewers and applicants since applicants only respond to predefined questions. Therefore, to elicit sufficient information from applicants for their selection becomes difficult.

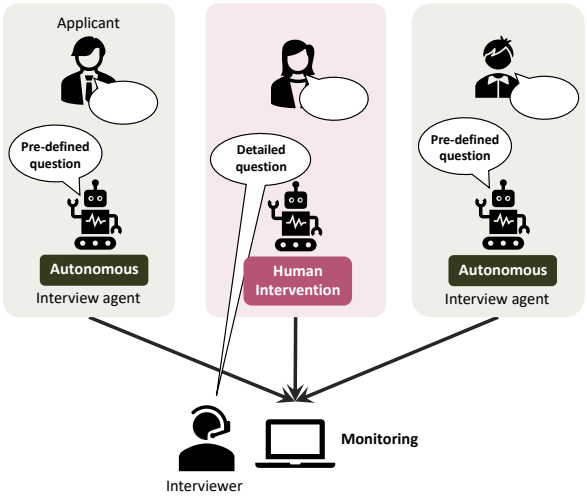


Figure 1: Concept of simultaneous job interview system

In this study, we propose a new framework for a spoken dialogue system that makes job interviews more interactive and efficient than that of on-demand interviews. The proposed framework is a cooperation between system and humans, namely semi-autonomous agents. With this framework, job interviewers can conduct multiple job interviews simultaneously. Specifically, a human job interviewer (operator) cooperates with multiple autonomous job interview agents to conduct one-on-one interviews with multiple applicants simultaneously (Figure 1). For most of the session, an autonomous agent conducts a job interview with each job applicant, and the human interviewer (operator) monitors them. The interviewer can intervene in any of the dialogues when necessary and then asks specific follow-up questions that cannot be generated by the autonomous agent. These follow-up questions are necessary to make job interviews more interactive and substantial. In this paper, we describe the framework of the proposed system and report a pilot experiment.

[1] https://www.hirevue.com/
[2] https://modernhire.com/

Proceedings of the SIGdial 2022 Conference, pages 107–110
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

2 Simultaneous job interview system

First, we introduce the one-on-one autonomous job interview dialogue system which is a basic component of the proposed framework. This system only asks predefined questions one by one such as motivation, strengths, and weaknesses. Similar to the existing on-demand job interview systems, no follow-up questions are asked after the responses. Although several works exist on follow-up question generation in the job interview domain (Su et al., 2019; Inoue et al., 2020), the questions automatically generated by the system are not necessarily appropriate or what the interviewer actually wants to know.

Next, we describe the proposed simultaneous job interview system. In this system, each applicant is interviewed by the above autonomous agent, and the human interviewer observes these multiple interview sessions. If the human interviewer wants to directly interact with any applicant, the interviewer can switch from the autonomous agent and then interact with the applicant. For example, the interviewer can ask specific follow-up questions that cannot be generated by the autonomous agent. Then, after the interviewer ends the intervention, the autonomous agent continues the session.

In this system, the interviewer is required to comprehend each applicant's answer and then ask proper follow-up questions and also decide on the timing of intervention. However, due to the cognitive ability of humans, it is not possible to understand the contents of multiple dialogues simultaneously. Even if each log of automatic speech recognition is generated and shown to the interviewer, it is difficult to follow all of them. It is necessary to summarize the information of each session. Therefore, we introduce response evaluation and keyword extraction that enable the interviewer to follow the dialogues done by multiple agents, as follows.

2.1 Response evaluation

We implemented a model that automatically evaluates the quality of each applicant's response. First, we conducted an annotation of response quality using a job interview dialogue corpus containing 86 mock job interview sessions (Inoue et al., 2020). The following three metrics were evaluated on the 3-point scale, from 0 (low) to 2 (high), for each response from the corpus.

Table 1: Number of annotated samples for response evaluation (0: insufficient, 1: middle, 2: sufficient)

Evaluation item	0	1	2
Appropriateness	18	26	464
Concreteness	164	190	154
Conciseness	112	311	85

- Appropriateness (Does the response fulfill what was asked?)
- Concreteness (Is the response concrete? Does the response contain any evidence and specific episodes?)
- Conciseness (Is the response brief?)

The numbers of annotated samples for each score and item are summarized in Table 1.

For each evaluation item, we made a binary classifier with BERT where the input is the concatenation of the system's question and applicant's response. The pre-trained BERT model [3] was fine-tuned with the three class labels of each item. The five-fold cross-validation was conducted and the macro F1-scores were 64.2%, 71.6%, and 76.0% for appropriateness, concreteness, and conciseness, respectively. A sample input is shown below, and the response evaluation models correctly assign each score of 2.

> (What is your strength?)
> "I have a degree in education, so I know a lot about how to help children learn while having fun. I also studied specialized content in my master's program, which I believe will be useful in creating teaching materials. I am also well versed in special needs education, and I think my strength lies in my ability to work with a wide variety of children."

The sum of the three scores is presented to the operator and used for evaluation of applicants. The operator can choose to intervene the applicant who is given high scores. On the other hand, when the system is used for interview practice, the operator might intervene against applicants with low scores.

2.2 Keyword extraction

Keyword extraction was implemented using the same response data. We annotated keywords using

[3] https://github.com/cl-tohoku/bert-japanese

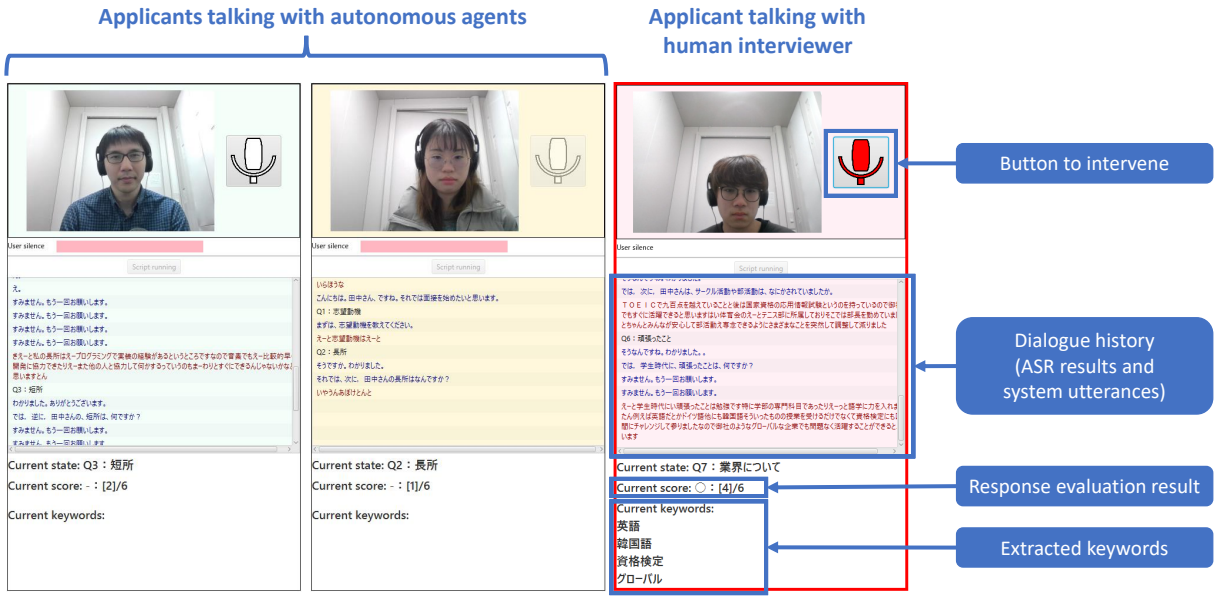


Figure 2: Interface for interviewer

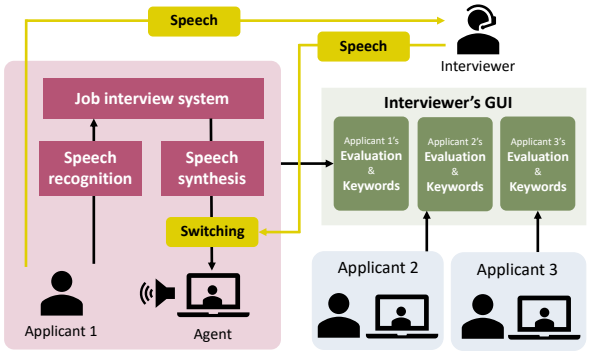


Figure 3: System configuration

the criterion of "words (or compound nouns) that represent the applicant's ability and experience". A character-based BiLSTM-CRF (Akbik et al., 2018) was used as a keyword extraction model. The benchmark result showed that the F1-score was 61.9%. For example, keywords extracted from the same input response as in Section 2.1, were "degree in education", "how to help", "specialized content", and "a wide variety of children". These keywords are presented to the interviewer as summary of the responses as a help for follow-up questions.

3 System implementation

Figure 3 depicts the configuration of the proposed system. The autonomous job interview system runs for each applicant. The input speech is segmented by a pause and fed into an automatic speech recognition with the sub-word-based attention mechanism. The recognition results are concatenated within the same turn and then used for the response evaluation and keyword extraction. The interface of the job interviewer agent is realized by MMDAgent (Lee et al., 2013). The system utterances are played with a text-to-speech engine.

Figure 2 shows the GUI for an interviewer where they can monitor multiple dialogues. This interface consists of mainly three items: (1) the dialogue history of each applicant, (2) the results of response evaluation, and (3) the results of keyword extraction. The human interviewer can select any applicant they want to intervene by clicking a button in the GUI. Once the interviewer selects an applicant, they can interact with each other directly, meanwhile, autonomous agents talk with the other applicants simultaneously.

4 Pilot experiment

We conducted a pilot experiment to confirm if the proposed system can handle multiple job interviews. In this experiment, a within-subject comparison was made between the fully autonomous system without human intervention (baseline) and the proposed system with three applicants. The subjects were 30 undergraduate and graduate students as applicants in the setting of "a student who participates in a first-round interview of some company." Note that the company was selected by each participant freely and independently. They were divided into groups of three persons in the condition

Table 2: Evaluation result in pilot experiment (5-point scale from 1:low to 5:high)

Evaluation items		Baseline		Proposed		p-value
		Mean	STD	Mean	STD	
(Q1)	The dialogue was smooth	4.14	1.53	4.32	0.82	.153
(Q2)	The system's responses were natural	4.07	1.18	4.14	1.02	.301
(Q3)	You participated in the interview seriously	4.36	0.91	4.43	0.77	.245
(Q4)	You were nervous during the interview	3.29	1.34	3.61	1.14	.030*
(Q5)	You talked well about yourself	3.64	1.13	3.93	1.03	.066+
(Q6)	You felt the interviewer listened your answers	3.29	1.40	4.14	0.35	<.001**
(Q7)	The interviewer understood you	3.11	1.43	3.64	0.83	.005**

(+ $p < .10$, * $p < .05$, ** $p < .01$)

of the proposed system. The evaluation items are listed in Table 2 where each was rated on a 5-point scale from 1 to 5. This experiment was conducted in Japanese.

Table 2 summarizes the evaluation results. The one-tailed paired *t*-test was conducted for each evaluation item, and the proposed system received significantly higher scores on the three items "You were nervous during the interview", "You felt the interviewer listened to your answers", and "The interviewer understood you". A significant trend was also observed for the item "You talked well about yourself". Although no significant trend was observed, the proposed method was rated higher than the baseline method for the other three items. Therefore, the proposed system improved the quality of interaction through the intervention of the interviewer, and can conduct efficient multiple job interviews with three applicants simultaneously.

We present some comments given by the subjects after the experiment. Following were the comments regarding the proposed system.

> "I thought it was efficient to let the machine ask the typical questions that have to be asked during the interview and let a human engage in interaction more advanced."
>
> "I get very nervous when the questions are asked back. It is good to have a realistic sense."

The baseline fully autonomous system received the following comments.

> "I did not feel like I was being listened to."
>
> "I did not really feel like I was being interviewed because I was always told "I see" after each answer. I did not feel like I was being interviewed very much."

5 Conclusions

We propose a simultaneous job interview system that allows human interviewers to interact with multiple applicants in real-time based on response evaluation and keyword extraction. For the interface of interviewers, the response evaluation and keyword extraction were implemented for making efficient intervention. In the pilot experiment, we showed the effectiveness of the proposed system and confirmed the proposed architecture would potentially be accepted as a new framework for future job interviews.

Acknowledgement

This work was supported by JST, Moonshot R&D Grant Number JPMJPS2011.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING*, pages 1638–1649.

Koji Inoue, Kohei Hara, Divesh Lala, Kenta Yamamoto, Shizuka Nakamura, Katsuya Takanashi, and Tatsuya Kawahara. 2020. Job interviewer android with elaborate follow-up question generation. In *ICMI*, pages 324–332.

Akinobu Lee, Keiichiro Oura, and Keiichi Tokuda. 2013. MMDAgent–A fully open-source toolkit for voice interaction systems. In *ICASSP*, pages 8382–8385.

Ming-Hsiang Su, Chung-Hsien Wu, and Yi Chang. 2019. Follow-up question generation using neural tensor network-based domain ontology population in an interview coaching system. In *Interspeech*, pages 4185–4189.

Dialog Acts for Task-Driven Embodied Agents

Spandana Gella[*], **Aishwarya Padmakumar**[*], **Patrick Lange, Dilek Hakkani-Tur**
Amazon Alexa AI
{sgella,padmakua,patlange,hakkanit}@amazon.com

Abstract

Embodied agents need to be able to interact in natural language – understanding task descriptions and asking appropriate follow up questions to obtain necessary information to be effective at successfully accomplishing tasks for a wide range of users. In this work, we propose a set of dialog acts for modelling such dialogs and annotate the *TEACh* dataset that includes over 3,000 situated, task oriented conversations (consisting of 39.5k utterances in total) with dialog acts. *TEACh-DA* is one of the first large scale dataset of dialog act annotations for embodied task completion. Furthermore, we demonstrate the use of this annotated dataset in training models for tagging the dialog acts of a given utterance, predicting the dialog act of the next response given a dialog history, and use the dialog acts to guide agent's non-dialog behaviour. In particular, our experiments on the *TEACh* Execution from Dialog History task where the model predicts the sequence of low level actions to be executed in the environment for embodied task completion, demonstrate that dialog acts can improve end task success rate by up to 2 points compared to the system without dialog acts.

1 Introduction

Natural language communication has the potential to significantly improve the accessibility of embodied agents. Ideally, a user should be able to converse with an embodied agent as if they were conversing with another person and the agent should be able to understand tasks specified at varying levels of abstraction and request for help as needed, identifying any additional information that needs to be obtained in follow up questions. Human-human dialogs that demonstrate such behavior are critical to the development of effective human-agent communication. Annotation of such dialogs with dialog acts is beneficial to better understand common conversational situations an agent will need to handle (Gervits et al., 2021). Dialog acts can also be used in building task oriented dialog systems to plan how an agent should react to the current situation (Williams et al., 2014).

In this paper, we design a dialog act annotation schema for embodied task completion based on the dialogs of the *TEACh* dialog corpus (Padmakumar et al., 2021). *TEACh* is a dataset of over 3,000 situated text conversations between human annotators role playing a user (*Commander*) and a robot (*Follower*) collaborating to complete household tasks such as making coffee and preparing breakfast in a simulated environment. The tasks are hierarchical, resulting in agents needing to understand task instructions provided at varying levels of abstraction across dialogs. The human annotators had a completely unconstrained chat interface for communication, so the dialogs reflect natural conversational behavior between humans, not moderated by predefined dialog acts or turn taking. Additionally, the *Follower* had to execute actions in the environment that caused physical state changes which were examined to determine whether a task was successfully completed. We believe that these annotations will enable the study of more realistic dialog behaviour in situated environments, unconstrained by turn taking.

Summarizing our contributions:

- We propose a new schema of dialog acts for task-driven embodied agents. This consists of 18 dialog acts capturing the most common communicative functions used in the *TEACh* dataset.
- We annotate the *TEACh* dataset according to the proposed schema to create the *TEACh-DA* dataset.
- We investigate the use of the proposed dialog acts in an extensive suite of tasks related to language understanding and action prediction for task-driven embodied agents.

[*]These two authors contributed equally.

Proceedings of the SIGdial 2022 Conference, pages 111–123
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

We establish baseline models for classifying the dialog act of a given utterance in our dataset and predicting the next dialog act given an utterance and conversation history. Additionally, we explore whether dialog acts can aid in plan prediction - predicting the sequence of object manipulations the agent needs to make to complete the task, and Execution from Dialog History (EDH) - where the agent predicts low level actions that are executed in the virtual environment and directly evaluated on whether required state changes were achieved.

2 Related Work

Dialog act annotations are common in language-only task-oriented dialog datasets, and are commonly used to plan the next agent action in dialog management or next user action in user simulation (Williams et al., 2014; Budzianowski et al., 2018; Schuster et al., 2019; Hemphill et al., 1990; Feng et al., 2020; Byrne et al., 2019). Many frameworks have been proposed to perform such annotations. Some examples are DAMSL (Dialog Act Markup in Several Layers) and ISO (International Organization for Standardization) standard (Core and Allen, 1997; Young, 2007; Bunt et al., 2009; Mezza et al., 2018). Such standardization of dialog acts across applications has been shown to be beneficial for improving the performance of dialog act prediction models (Mezza et al., 2018; Paul et al., 2019).

Most task-oriented dialog (TOD) applications and dialog act coding standards assume that the tasks to be performed can be fully specified in terms of slots whose values are entities (Young, 2007). However, we find that if we need to adopt a slot-value scheme for multimodal task-oriented dialog datasets such as *TEACh*, much of the information that needs to be conveyed is not purely in the form of entities. For example, If an utterance providing a location of an object: "the cup is in the drawer to the left of the sink" is to be coded at the dialog act level simply as an `INFORM` act, it could for example have a slot value called `OBJECT_LOCATION` but the value of this would need to refer to most of the utterance, i.e. "the drawer to the left of the sink". Hence, we define more fine-grained categories, such as `InfoObjectLocAndOD` (information on object location and other details) in *TEACh-DA*. These categories are designed in a way so that they could be re-purposed into broader dialog act category and intent/slot in the future by merging categories, if needed. As in a TOD, inform would be the DA tag, intent could be `inform_object_location` or `object_location` could be slot category. Thus, we combine the use of many standardized dialog acts such as `Greetings`, `Acknowledge`, `Affirm / Deny` with domain-specific finer grained dialog acts replacing the typical `Inform` and `Request` dialog acts.

Additionally, since the *TEACh* dataset is not constrained by turn taking or a pre-defined dialog flow, sometimes a single utterance may perform multiple communicative functions. To address this, similar to Core and Allen 1997, we allow multiple dialog acts per utterance and require annotators to mark utterance spans corresponding to each dialog act.

There exist other multimodal task-oriented dialog datasets that include annotations of dialog acts such as Situated and Interactive Multimodal Conversations (SIMMC 2.0) (Kottur et al., 2021) and Multimodal Dialogues (MMD) (Saha et al., 2018). These are multimodal datasets in the shopping domain that allows users to view products visually, and engage in dialog with an agent where the agent can take actions to refine the products available for the user to view. However, in contrast to the *TEACh* dataset considered in our work, the dialogs are created by first simulating probable dialog flows and then having annotators paraphrase utterances. As such, in these datasets, utterances clearly map to predefined dialog acts and follow patterns expected by the designers. These may not fully cover the range of possible conversational flows that can happen between humans in an unconstrained multimodal context, as can be observed in *TEACh*. The Human Robot Dialogue Learning (HuRDL) corpus includes annotations of human-human multimodal dialogs, with a focus on classifying different types of clarification questions to be used by a dialog agent (Gervits et al., 2021) but it is limited in size - consisting of only 22 dialogs, in contrast to the over 3,000 dialogs in *TEACh*. Another related dataset is MindCraft (Bara et al., 2021) where annotators are periodically asked to answer questions in the middle of the collection of dialog sessions to elicit their belief states. However, belief states do not map directly to utterances and do not directly capture communicative intents, differentiating them from dialog acts.

Prior works propose models for predicting dialog acts given the current utterance and context (Kalch-

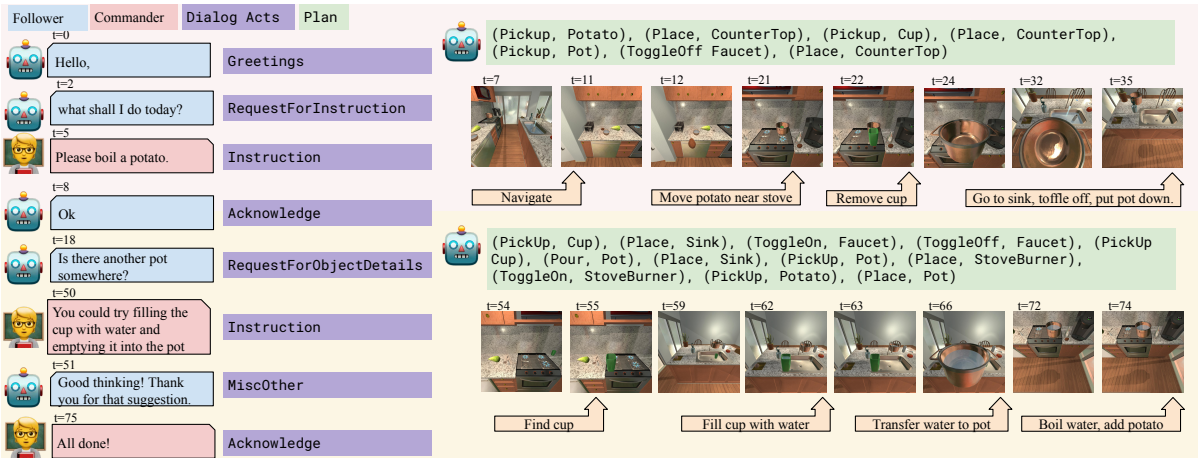


Figure 1: Illustration of example session for the task *Boil Potato* with corresponding dialog acts for each utterance and plans with corresponding actions in the game session.

brenner and Blunsom, 2013; Lee and Dernoncourt, 2016; Ribeiro et al., 2019), dialog acts of previous utterances or both (Paul et al., 2019). We perform similar experiments on our dataset to tag the dialog acts of given utterances and also to predict the dialog acts of future utterances. Due to the limited set of situated dialog datasets annotated with dialog acts, there has been relatively limited work on exploring the benefit of dialog acts on predicting an agent's future behavior in the environment. However, there are works that explore when to engage in a dialog as opposed to acting in the environment (Gervits et al., 2020; Chi et al., 2020; Shrivastava et al., 2021). While we do not directly model this problem, we experiment with the *TEACh* Execution from Dialog History task, where the end of our predicted action sequence would signal the need for another dialog utterance.

3 *TEACh-DA* dataset

The *TEACh* dataset (Padmakumar et al., 2021) consists of situated dialogs between human annotators role playing a user (*Commander*) and robot (*Follower*) collaborating to complete household tasks. In each dialog session, there is a high level task that the *Follower* is expected to accomplish, for example MAKE COFFEE or PREPARE BREAKFAST. Details of the task are known to the *Commander* but not the *Follower*. The *Follower* needs to engage in a dialog with the user to identify the task to be completed, customize the task (for example identify what dishes need to be prepared for breakfast) or obtain additional information such as locations of relevant objects, or more detailed steps needed to accomplish a task, and translate these to actions that can be executed in a simulated environment to complete the task.

In this work, we annotate the *TEACh* dataset with dialog acts (we refer to this new, annotated dataset as *TEACh-DA*) to better understand how language is used in task-oriented situated dialogs. We also explore the usefulness of these dialog acts to develop better agents that can converse in natural language and act in a situated environment for task completion.The *TEACh-DA* dataset consists of 39.5k utterances from 3,000 dialogs, 60% of which are from the *Commander* and the rest from the *Follower*.

We find that other dialog act frameworks for multimodal datasets (Gervits et al., 2021; Kottur et al., 2021; Saha et al., 2018) tend to be domain specific and do not cover all utterance types that would be beneficial for embodied task completion. Hence, we propose a new set of dialog acts for embodied task completion based on the communicative functions we observe in the *TEACh* dataset. Whenever possible, for utterances that are not very specific to the *TEACh* task, we have borrowed dialog acts from prior work. These include dialog acts related to generic chit chat such as `Greetings`, `Affirm`, `Deny` and `Acknowledge` (Paul et al., 2019).

In total, we defined 18 dialog acts that covered all utterances in *TEACh*. Our careful analysis of utterances in *TEACh* data lead to 5 broader categories of dialog acts as shown in Table 1.

- Generic: Acts that fall under conventional dialog such as opening and closing of dialog,
- Instruction Related: Which represent the utterances related to actions that should be per-

Dialog Act	Category	Example	Count	*Commander*(%)	*Follower*(%)
`Instruction`	Instruction	fill the mug with coffee	11019	99.4	0.6
`ReqForInstruction`	Instruction	what should I do today?	4043	0.7	99.3
`RequestOtherInfo`	Instruction	How many slices of tomato?	675	0.75	99.25
`RequestMore`	Instruction	Is there anything else to do	503	0.2	99.80
`InfoObjectLocAndOD`	Object/Location	knife is behind the sink	6946	99.4	0.6
`ReqForObjLocAndOD`	Object/Location	where is the mug?	2010	0.3	99.70
`InformationOther`	Object/Location	Mug is already clean	1148	88.76	11.24
`AlternateQuestions`	Object/Location	yellow or blue mug?	123	27.65	72.35
`Acknowledge`	Generic	perfect	7421	21.38	78.62
`Greetings`	Generic	hello	2565	44.01	55.9
`Confirm`	Generic	Should I clean the cup?	726	25.75	74.25
`MiscOther`	Generic	ta-da	607	52.22	47.78
`Affirm`	Generic	Yes	460	78.26	21.74
`Deny`	Generic	No	161	72.92	26.08
`FeedbackPositive`	Feedback	great job	2745	97.12	2.88
`FeedbackNegative`	Feedback	that is not correct	46	95.65	4.35
`OtherInterfaceComment`	Interface	Which button opens drawer	486	60.09	39.91
`NotifyFailure`	Interface	not able to do it	408	3.68	96.32

Table 1: Dialog act labels, total number of utterances and frequencies per speaker type in overall corpus.

formed in the environment to accomplish the household task.

- Object/Location related: Represents requests and information seeking utterances related to objects that need to be handled or manipulated for the specific *TEACh* task. Many of these are on the specifics of object location (where to find it, where to place it) and queries on disambiguation related to objects or their locations.
- Interface Related: Utterances related to *TEACh* data annotation itself (`NotifyFailure` and `OtherInterfaceComment`)
- Feedback related: Utterances used to provide feedback (both positive and negative) on navigation, object manipulation and in general task execution.

We hired expert annotators who are fluent in English to annotate utterances from the *TEACh* dataset with our dialog acts. Annotators were shown the complete dialog and asked to annotate each utterance with the most appropriate dialog act. When an utterance had multiple dialog acts applicable, annotators were asked to divide the utterance into spans and annotate each span with a single dialog act label. We observed that 7% of the utterances were segmented to have multiple dialog acts. To measure the quality of the annotations, on a small subset of 235 utterances (17 dialogs), we collected annotations from two annotators. On this subset, we observed a Cohen's kappa score of 0.87. We include an example *TEACh* session in Figure 1 for the task *Boil Potato* containing dialog act acctions for each utterance.

Similar to many task-oriented dialogs, we observe a strong correlation between the speaker role (*Commander* or *Follower*) and the dialog act of an utterance. For example, the majority of the inform utterances are from *Commander* i.e., where *Commander* gives instructions or informs object locations or other details on the task, whereas majority of the request utterances (instructions, object locations etc.) are from *Follower*. In Table 1, we present the set of dialog acts, definitions and their frequency distributed across *Commander* and *Follower* utterances. We observe that some communicative functions such as clarification of ambiguity are relatively infrequent in this dataset. We group together such rare functions into a single dialog act `MiscOther`.

4 Experiments

In this section, we explore how dialog acts can be used for various modeling tasks including predicting the agent's future behavior in the environment. We explore the following tasks (i) dialog act classification: predicting the dialog act of an utterance; (ii) future turn dialog act prediction given dialog history; (iii) given *TEACh* dialog history, predicting a plan for the task and (iv) given dialog history and the past actions in environment, predicting the entire sequence of low-level actions to be executed in the *TEACh* environment to complete the task (Execution from Dialog History (EDH) benchmark from Padmakumar et al. 2021). Note that *TEACh*

Utterance (Utt)	the bowl is in the microwave
Utt + ST	<<Commander>> the bowl is in the microwave
Utt + DH	how can i help <<TURN>> please serve 1 slice of tomato in a bowl <<TURN>> where can i find a bowl <<TURN>> the bowl is in the microwave
Utt + DH + DA-E	how can i help <<ReqForInstruction>> <<TURN>> please serve 1 slice of tomato in a bowl <<Instruction>> where can i find a bowl <<ReqForObjLocAndOD>> <<TURN>> the bowl is in the microwave <<InfoObjectLocAndOD>>
Utt + ST + DH + DA-E	<<Follower>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>> <<TURN>> <<Follower>> where can i find a bowl <<ReqForObjLocAndOD>> <<TURN>> <<Commander>> the bowl is in the microwave <<InfoObjectLocAndOD>>

Figure 2: Sample input to dialog act prediction or next turn dialog act prediction models showing incorporation of speaker and dialog history

	Valid seen	Valid unseen	Test seen	Test unseen
Utterance	85.59	83.74	85.88	83.59
+Speaker Tags (ST)	87.98	85.91	87.55	85.73
+ Dialog History (DH)	86.7	84.66	86.48	84.25
+ DH + DA-E	**88.6**	**86.32**	88.35	**86.09**
+ DH + ST + DA-E	88.35	86.15	**88.54**	85.89
Follower utterances only				
Utterance	83.12	79.58	84.86	83.85
+Speaker Tags (ST)	86.84	82.26	88.33	**87.71**
+Dialog History (DH)	86.52	84.13	86.67	84.53
+ DH +DA-E	**88.62**	**85.87**	88.82	86.56
+ DH + ST + DA-E	88.32	85.79	**89.22**	86.3
Commander utterances only				
Utterance	87.16	86.71	86.5	83.42
+ Speaker Tags (ST)	**88.70**	**88.52**	**87.08**	84.42
+ Dialog History (DH)	87.11	81.03	85.79	83.49
+ DH + DA-E	88.55	87.90	86.69	**84.84**
+ DH + ST + DA-E	88.42	87.4	86.15	84.79

Table 2: Dialog Act prediction accuracy scores for whole *TEACh-DA* dataset. We also report accuracy scores for *Follower* and *Commander* utterances separately.

has two validation and two test splits each - seen and unseen. These refer to visual differences between the environments in which gameplay sessions occurred. With the exception of the EDH experiment, since we only focus on language, we do not expect significant differences between the seen and unseen splits.

4.1 Dialog Act Classification

Dialog Act classification is the task of identifying the general intent of the user utterance in a dialog. While dialog act classification has been well explored in both task-oriented dialogs and open-domain dialogs, it is still an under explored problem in human-robot dialogs (Gervits et al., 2020). We study the *TEACh* dataset to predict the dialog act for a given utterance. We experimented with fine-tuning a large pre-trained language model *RoBERTa-base* for the classification of dialog acts[1]. We expect the speaker role (*Follower* or *Commander*) and the dialog context to be important for predicting the intent of an utterance. To test this, we predict dialog acts with different input formats (shown in Figure 2) ablating the value of speaker and context information (DH: all the previous utterances in the dialog, ST: speaker tags, DA-E: ground-truth dialog act tags of all the previous utterances in the dialog). We present our results in Table 2. Similar to prior studies on dialog act classification for task-oriented dialogs, we observe that both the speaker tags and dialog history help in predicting the correct dialog act for a given utterance, and the best performance is observed when both of them are used.

In *TEACh*, the distribution of dialog acts varies with the speaker role (*Commander* vs. *Follower*) as shown in Table 1. To understand the accuracy of the models on utterances of each speaker role, we also present results separated by speaker role in Table 2. We observed that both speaker tags and dialog history with previous turn dialog acts helped identifying dialog acts for *Follower* utterances. For *Commander* utterances both speaker tags and dialog history gave marginal improvements.

[1]We also experimented with *BERT-base* and *TOD-BERT* but observed *RoBERTa-base* performed consistently better

	Valid seen	Valid unseen	Test seen	Test unseen
DH	42.62	42.44	43.55	41.07
DH + ST	56.23	54.68	54.69	53.27
DH + DA-E	56.05	55.58	56.49	53.45
DH + ST + DA-E	**56.72**	**56.14**	**56.28**	**54.99**
Follower utterances only				
DH	30.73	28.64	31.41	29.06
DH +ST	51.67	49.3	54.11	52.34
DH + DA-E	50.19	50.28	54.72	52.24
DH + ST + DA-E	**52.17**	**50.35**	**54.72**	**53.44**
Commander utterances only				
DH	49.27	51.08	50.07	48.08
DH + ST	58.78	58.05	55.01	53.82
DH + DA-E	59.33	58.9	57.4	54.16
DH + ST + DA-E	**59.26**	**59.77**	**57.11**	**55.89**

Table 3: Predict next utterance Dialog Act given dialog history. We also report results when next utterance is *Commander* and *Follower* separately. Speaker Tags: Additional to current utterance speaker tag we also provide next utterance speaker information.

4.2 Next Dialog Act Prediction

In end-to-end dialog models, predicting the desired dialog act for the next turn is useful for response generation (Tanaka et al., 2019). Predicting the dialog act of the next response in *TEACh* will provide insights into a model's ability to provide appropriate dialog responses. This is particularly useful for *Follower* utterances to enable the agent to identify when to ask for more instructions or additional information to accomplish a sub-task. We modeled this as a classification task where we provide dialog history until a particular turn as input and predict the dialog act of the next turn. In addition to providing dialog history, we also tested this to see if providing next turn speaker information will improve the performance of the model. Similar to our dialog act classification model in Section 4.1 we fine-tuned a *RoBERTa-base* model for predicting the dialog act of the next utterance. In Table 3, we present results for next dialog act prediction. We observe a significant improvement in the performance for next dialog act prediction when the next utterance is from the *Follower* and the speaker information or previous utterances dialog act is added to the input. We hypothesize that the accuracy in this task is low compared to similar tasks in other task-oriented dialog datasets because this dataset does not enforce turn taking. The *Commander* or *Follower* may break up a single intent into multiple utterances and one may anticipate the next response from the other before it is asked. For example, if the *Commander* has asked the *Follower* to slice a tomato, the *Commander* may expect that the *Follower* is likely to then ask for the locations of the tomato or the knife and may start providing this information before the *Follower* has asked for it. Further, the *Commander* or *Follower* may have responded directly to visual cues or actions taken by the other in the environment. Hence, visual or environment information is likely also important for predicting future dialog acts.

4.3 Plan Prediction

In robotics, task planning is the process of generating a sequence of symbolic actions to guide high-level behavior of a robot to complete a task (Ghallab et al., 2016). In this experiment, we consider a simple plan representation where a task plan consists of a sequence of object manipulations that need to be completed in order for the task to be successful. An example is included in Figure 3 When executing such a plan, the robot will need to navigate to required objects and additional steps may be required based on the state of the environment (for example if the microwave is too full, the robot may need to partially clear it first).

However, it should be possible to generate the plan for a task based on the dialog alone. We explore two settings for this

- `Game-to-Plan`: Given the entire dialog from a gameplay session, predict the plan - that is, all object interaction actions taken during that gameplay session.
- `Dialog-History-to-Plan`: Given a portion of dialog history from a gameplay session, predict the object interaction actions that need to occur until the next dialog utterance.

The `Game-to-Plan` setting is more likely to be useful for post-hoc analysis of such situated interactions after they have occurred, whereas the `Dialog-History-to-Plan` setting can be used to build an embodied agent that engages in dialog with a user and executes actions in a virtual environment based on information obtained in the dialog. At any point in time, such an agent would predict the next few object interactions to be accomplished given the dialog history so far, complete

Language Input:	
DH	how can i help <<TURN>> please serve 1 slice of tomato in a bowl <<TURN>> where can i find a bowl <<TURN>> the bowl is in the microwave
DH + DA	<<Follower>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>> <<TURN>> <<Follower>> where can i find a bowl <<TURN>> <<Commander>> the bowl is in the microwave
DH + DA + Filter	<<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>>
Language Output:	
Pickup Tomato -- Place CounterTop -- Pickup ButterKnife -- Slice Tomato -- Place CounterTop – Pickup TomatoSliced -- ToggleOff Microwave -- Open Microwave -- Place Bowl -- Pickup Bowl	

Figure 3: Sample input and output for plan prediction showing incorporation of speaker and dialog act information.

	Game-to-Plan											
	Percentage of valid plans				Plan tuple precision				Plan tuple recall			
	Valid seen	Valid unseen	Test seen	Test unseen	Valid seen	Valid unseen	Test seen	Test unseen	Valid seen	Valid unseen	Test seen	Test unseen
DH	24.31	**30.39**	**28.18**	28.69	72.67	**73.93**	73.48	**78.53**	37.06	**34.35**	37.46	**36.00**
+ DA	25.97	23.86	19.89	26.83	**75.29**	73.0	**74.81**	77.52	**38.18**	33.7	**39.28**	35.31
+ Filter	**37.57**	29.41	27.62	**32.94**	71.29	70.94	69.80	75.45	34.33	31.61	35.45	33.42
	Dialog-History-to-Plan											
DH	**23.76**	23.69	25.41	24.45	72.97	73.47	**75.65**	**78.64**	36.38	34.06	**39.11**	**36.53**
+ DA	24.31	**30.39**	**28.18**	**28.69**	72.67	**73.93**	73.48	78.53	**37.06**	**34.35**	37.46	36.0
+ Filter	26.52	23.69	25.41	28.01	**73.66**	69.88	71.67	74.33	36.08	31.29	35.83	33.12

Table 4: Plan prediction results. Using dialog act information helps increase the fraction of valid generated plans but not as much with plan precision or recall.

them and then use another module that makes use of subsequent dialog act prediction (section 4.2) to engage in further dialog with the user.

We model plan prediction as a sequence to sequence task where the input consists of the dialog / dialog history, and the output as a sequence of alternating object interaction actions (eg: `Pickup`, `Place`, `ToggleOn`) and object types (eg: `Mug`, `Sink`). We experiment with augmenting the dialog history with dialog act information (+ DA information) and filtering the input dialog to only contain utterance segments annotated as being of type `Instruction` (+ filter) We fine-tune a `BART-base` model for this task and evaluate different experimental conditions on the following metrics:

- Fraction of valid plans: Fraction of generated output sequences that consist of alternating valid actions and object types. (For example `(Pickup, Mug), (Place, Sink) (ToggleOn, Faucet)` is a valid sequence while `(Pickup, Mug) (Sink) (ToggleOn, Faucet)` and `(Pickup, Mug) (Place) (ToggleOn, Faucet))` are not due to the missing action for `Sink` and the missing object for `Place` respectively.
- Precision of (action, object) tuples: We identify a valid object type followed by a valid action as an (action, object) tuple and precision is the fraction of such tuples in the generated output present in the ground truth plan.
- Recall of (action, object) tuples: Recall is the fraction of (action, object) tuples in the ground truth plan present in the generated output.

The results are included in Table 4. We notice that addition of dialog act information and filtering to relevant dialog acts improves performance in some splits but not others. More improvements are seen in the `Dialog-History-to-Plan`

DH	how can i help <<TURN>> please serve 1 slice of tomato in a bowl
DH + ST	<<Follower>> how can i help <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl
DH + ST + DA-E	<<Follower>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> please serve 1 slice of tomato in a bowl <<Instruction>>
DH + DA-E	how can i help <<ReqForInstruction>> <<TURN>> please serve 1 slice of tomato in a bowl <<Instruction>>
DH + ST + DA-SE	<<Follower>> <<ReqForInstruction>> how can i help <<ReqForInstruction>> <<TURN>> <<Commander>> <<Instruction>> please serve 1 slice of tomato in a bowl <<Instruction>>

Figure 4: Language Input Variants for EDH.

	EDH Validation				**EDH Test**			
	Seen		*Unseen*		*Seen*		*Unseen*	
Language Input	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]
DH	7.9 [1.0]	7.1 [3.3]	6.7 [0.4]	3.9 [1.5]	10.5 [0.5]	7.9 [3.2]	7.5 [0.7]	5.6 [1.9]
+ ST	6.7 [0.5]	7.4 [2.8]	6.7 [0.8]	4.0 [1.5]	9.8 [0.9]	8.3 [2.9]	7.1 [0.8]	6.6 [1.7]
+ DA-E	8.5 [0.6]	**8.2 [3.3]**	6.7 [0.5]	**5.0 [1.9]**	**12.2 [1.2]**	8.6 [3.7]	7.4 [0.8]	6.1 [2.3]
+ DA-SE	7.8 [1.8]	6.4 [4.0]	7.2 [0.6]	4.6 [1.6]	11.0 [0.7]	**10.1 [4.3]**	**7.7 [0.8]**	6.2 [1.8]
+ ST + DA-SE	**8.7 [1.0]**	7.3 [2.6]	**7.5 [0.8]**	4.4 [1.8]	9.9 [0.7]	8.0 [2.9]	7.0 [0.7]	**7.2 [2.2]**

Table 5: We experiment whether addition of speaker or dialog act information improves performance of the Episodic Transformer (E.T.) model on the Execution from Dialog History (EDH) task. In most cases, speaker information is not found to be beneficial but adding dialog acts at the end or start and end of an utterance is seen to provide small improvements in performance.

setting compared to the `Game-to-Plan` setting. We hypothesize that this is because the model is able to automatically identify the dialog act from the utterance text and hence does not need it to be explicitly specified.

4.4 Execution from Dialog History

The Execution from Dialog History (EDH) task defined in the Padmakumar et al. 2021 is an extension of the above task. Instead of simply predicting important object interactions, given dialog history and past actions in the environment, a model is expected to predict a full sequence of low level actions to accomplish the task described in the dialog. Action sequences predicted by the model are executed in the virtual environment and models are evaluated based on how many required object state changes are accomplished. The metrics used for this task include the fraction of successful state changes (goal condition success rate or GC), the fraction of sessions for which all state changes were accomplished (success rate or SR) and Trajectory Length Weighted versions of these metrics that multiply the metrics with the ratio of the ground truth path length to the predicted path length - where a lower value of the trajectory weighted metric suggests that the model used longer sequences of actions to accomplish the same state changes.

We borrow the Episodic Transformer (E.T.) model proposed in Padmakumar et al. 2021 and vary the language input (with a baseline of just the dialog history (DH)) by adding speaker tags (+ST) and ground-truth dialog act tags at the start (+DA-S), end (+DA-E) or both (+DA-SE). We present the results for selected set of experiments in Table 5. We observe small performance improvements on success rate of up to 2 points when the language input is marked up with dialog acts, either at the end or start and end of an utterance, but less benefit is observed from speaker information. We believe that stronger improvements will likely be observed when using a more modular approach (eg: (Min et al., 2021)) where it is easier to decouple the effects of errors arising from language understanding from those arising from navigation which is the most difficult component when predicting such

low-level actions (Blukis et al., 2022; Jia et al., 2022; Min et al., 2021).

5 Conclusion

We propose a new dialog act annotation framework for embodied task completion dialogs and use this to annotate the *TEACh* dataset - a dataset of over 3,000 unconstrained, situated human-human dialogs. We evaluate baseline models for predicting dialog acts of utterances, demonstrate that predicting future dialog acts from past ones is much more difficult in dialog datasets that are not constrained by turn taking. Towards guiding agent actions in the environment beyond dialog, we show explore the benefit of dialog acts in the generation of plans, and improve end-to-end performance in the *TEACh* Execution from Dialog History task.

6 Future Work

Unlike the majority of dialog datasets, situated or otherwise, utterances in the *TEACh* dataset are not constrained by a pre-designed dialog act schema or by turn taking. We observe that this makes it much more difficult than expected to predict subsequent dialog acts given past ones - the predictability of which has been typically used to design dialog simulators (Schatzmann and Young, 2009; Keizer et al., 2010). We believe that annotation of this large and more natural dataset will aid in the development of more realistic dialog simulators, which can in turn result in the development of more natural dialog agents. Further, in *TEACh*, visual cues or actions taken by the agent in the environment might play an important role for predicting future dialog acts. This would be an interesting direction to explore for future. Finally, we hypothesize that there is considerable scope in using such annotated dialog acts to develop modular models for embodied task completion that involve better language understanding, and to generate realistic situated dialogs for data augmentation.

References

Cristian-Paul Bara, CH-Wang Sky, and Joyce Chai. 2021. Mindcraft: Theory of mind modeling for situated dialogue in collaborative tasks. In *Proceedings of EMNLP 2021*, pages 1112–1125.

Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. 2022. A persistent spatial semantic representation for high-level natural language instruction execution. In *Conference on Robot Learning*, pages 706–717. PMLR.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*.

Harry Bunt, Dirk K. J. Heylen, Catherine Pelachaud, Roberta Catizone, and David R. Traum. 2009. The dit++ taxanomy for functional dialogue markup. In *EDAML@AAMAS, Workshop Towards a Standard Markup Language for Embodied Dialogue Acts*, pages 13–24.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525.

Ta-Chung Chi, Minmin Shen, Mihail Eric, Seokhwan Kim, and Dilek Hakkani-Tür. 2020. Just ask: An interactive learning framework for vision and language navigation. In *AAAI 2020*, pages 2459–2466. AAAI Press.

Mark G. Core and James F. Allen. 1997. Coding dialogs with the damsl annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128.

Felix Gervits, Antonio Roque, Gordon Briggs, Matthias Scheutz, and Matthew Marge. 2021. How should agents ask questions for situated learning? an annotated dialogue corpus. In *Proceedings of the SIGDIAL 2021*, pages 353–359.

Felix Gervits, Ravenna Thielstrom, Antonio Roque, and Matthias Scheutz. 2020. It's about time: Turn-entry timing for situated human-robot dialogue. In *Proceedings of the SIGDIAL 2020*, pages 86–96.

Malik Ghallab, Dana Nau, and Paolo Traverso. 2016. *Automated planning and acting*. Cambridge University Press.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Zhiwei Jia, Kaixiang Lin, Yizhou Zhao, Qiaozi Gao, Govind Thattai, and Gaurav Sukhatme. 2022. Learning to act with affordance-aware multimodal neural slam. *arXiv preprint arXiv:2201.09862*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126.

Simon Keizer, Milica Gasic, Filip Jurcicek, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Parameter estimation for agenda-based user simulation. In *Proceedings of the SIGDIAL 2010 Conference*, pages 116–123.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of EMNLP*, pages 4903–4912.

Ji Young Lee and Franck Dernoncourt. 2016. Sequential short-text classification with recurrent and convolutional neural networks. In *NAACL HLT*, pages 515–520.

Stefano Mezza, Alessandra Cervone, Evgeny A. Stepanov, Giuliano Tortoreto, and Giuseppe Riccardi. 2018. Iso-standard domain-independent dialogue act tagging for conversational agents. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3539–3551.

So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. 2021. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2021. Teach: Task-driven embodied agents that chat. *arXiv preprint arXiv:2110.00534*.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. 2021. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952.

Shachi Paul, Rahul Goel, and Dilek Hakkani-Tür. 2019. Towards universal dialogue act tagging for task-oriented dialogues. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, pages 1453–1457. ISCA.

Eugénio Ribeiro, Ricardo Ribeiro, and David Martins de Matos. 2019. Deep dialog act recognition using multiple token, segment, and context information representations. *J. Artif. Intell. Res.*, 66:861–899.

Amrita Saha, Mitesh Khapra, and Karthik Sankaranarayanan. 2018. Towards building large scale multimodal domain-aware conversation systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jost Schatzmann and Steve Young. 2009. The hidden agenda user simulation model. *IEEE transactions on audio, speech, and language processing*, 17(4):733–747.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of NAACL-HLT*, pages 3795–3805.

Ayush Shrivastava, Karthik Gopalakrishnan, Yang Liu, Robinson Piramuthu, Gökhan Tür, Devi Parikh, and Dilek Hakkani-Tür. 2021. VISITRON: visual semantics-aligned interactively trained object-navigator. *CoRR*, abs/2105.11589.

Koji Tanaka, Junya Takayama, and Yuki Arase. 2019. Dialogue-act prediction of future responses based on conversation history. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*, pages 197–202.

Jason D Williams, Matthew Henderson, Antoine Raux, Blaise Thomson, Alan Black, and Deepak Ramachandran. 2014. The dialog state tracking challenge series. *AI Magazine*, 35(4):121–124.

Steve Young. 2007. Cued standard dialogue acts. *Report, Cambridge University Engineering Department*, 2007.

A Further Experiment Details

A.1 Dialog Act Classification and Next Turn Dialog Act Prediction

Both for dialog act classification and next turn dialog act prediction models, we finetune a `RoBERTa-base` model for multiclass classification with 18 classes (our target number of dialog acts). For all the experiments were run using Huggingface library and the publicly available pre-trained models. Additional to the utterance we provide dialog-context and speaker information (mentioned as dialog history (DH) and Speaker Info (SI)) and train the classifiers for a maximum sequence length of 512 tokens. When the input exceeds 512 tokens we truncate from left i.e., we keep the most recent context. We use a batch size of 16 per GPU and accumulate gradients across 4 GPU instances. We use a learning rate of $2e-05$ and train for 5 epochs.

A.2 Plan Prediction

For the plan prediction task, we finetune a `bart-base` model, treating the problem as sequence to sequence prediction. A sample input and output from the `Game-to-Plan` version of the task are included below:

Sample Input:

```
what do I do? <<TURN>> making
coffee <<TURN>> grab a mug
<<TURN>> where is tyhe mug?
<<TURN>> on the counter next to
you <<TURN>> empty, and wash
<<TURN>> should I wash the mug
<<TURN>> place in coffee maker
after cleaning <<TURN>> yes
<<TURN>> okay <<TURN>> turn on
water <<TURN>> turn off <<TURN>>
place in coffee maker next to
sink <<TURN>> empty first
<<TURN>> turn on <<TURN>> great
job....we're done... <<TURN>>
```

Sample Output:

```
Pickup Mug Pour SinkBasin Place
SinkBasin ToggleOn Faucet
ToggleOff Faucet Pickup Mug Pour
SinkBasin Place CoffeeMachine
ToggleOn CoffeeMachine
```

Note that we do not include any punctuation in the output sequence to demarcate (action, object) tuples and instead post process the generated sequence deleting any action not followed by an object or object not preceded by an action for evaluation. Also, while we use ⟨⟨TURN⟩⟩ in the above example to demarcate turns, in actual implementation, the default BART separator token is used.

All experiments are run using the HuggingFace library and pretrained models [2]. We use a batch size of 2 per GPU accumulating gradients from batches on 4 GPUs of an AWS 'p3.8xlarge' instance leading to an effective batch size of 8. Training was done for 20 epochs. We use the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.99, \epsilon = 1e-08$ and weight decay of 0.01. We use a learning rate of $5e-05$ with a linear warmup over 500 steps. Where necessary, we right-truncate the input to the model's limit of 1024 tokens as we believe that when an incomplete conversation must be used, the model may be able to infer most of the necessary steps from the task information which is likely to be indicated by the first few utterances of the conversation.

The primary hyperparameter tuning we experimented with involved the position at which the dialog act was inserted relative to the utterance, which was one of

- START_OF_SEGMENT - Start of the utterance segment
- END_OF_SEGMENT - End of the utterance segment
- START_END_SEGMENT - Start and end of the utterance segment

and the format used to insert dialog act information, which was one of

- NO_CHANGE_TEXT - The name of the dialog act is inserted in Camel case as a part of the input text to the model.
- FILTER - Retain only utterances marked with the dialog act INSTRUCTION. Additionally, the name of the dialog act is inserted in Camel case as a part of the input text to the model.
- TAGS_IN_TEXT - The name of the dialog act in Camel case is surrounded by ⟨⟨⟩⟩.
- TAGS_SPL_TOKENS - The name of the dialog act in Camel case is surrounded by ⟨⟨⟩⟩ and this is specified as being a special token so that it does not get split by the tokenizer.
- SPLIT_WORDS_TEXT - The name of the dialog act is split into individual words (for example, REQUESTFORINSTRUCTION becomes "request for instruction") and these are inserted into the text.

We also tuned whether speaker information was passed to the model. None of the format, position or speaker tag choices were found to consistently outperform the other.

For the DH rows in table 4, neither the position, nor the format of dialog acts is relevant as no dialog act information is used. We also do not filter utterances. The best +DA row in the `Game-to-Plan` setting used dialog acts in format SPLIT_WORDS_TEXT in position END_OF_SEGMENT with speaker tags. The best +Filter row in the `Game-to-Plan` setting used dialog acts in format START_END_SEGMENT

[2] https://huggingface.co/

without speaker tags. The best +DA row in the `Dialog-History-to-Plan` setting used dialog acts in format SPLIT_WORDS_TEXT in position START_OF_SEGMENT without speaker tags. The best +Filter row in the `Dialog-History-to-Plan` setting used dialog acts in format END_OF_SEGMENT without speaker tags.

A.3 Execution from Dialog History

We adapt the Episodic Transformer (E.T.) model first introduced in (Pashevich et al., 2021) and used for baseline experiments in (Padmakumar et al., 2021) on the TEACh dataset. We keep all training parameters constant from (Padmakumar et al., 2021) and primarily experiment with the input format as described in the main paper. Unlike our previous experiments, since the language encoder of the E.T. model is trained from scratch using only the vocabulary present in the training data, we insert dialog acts and speaker indicators as individual tokens in the input that will be treated identically to other text tokens.

B Dialog Acts

In Table 6 we add further examples for each dialog act (for both *Follower*and *Commander*) from different *TEACh* tasks to demonstrate the difference in type of utterances we observe in the dataset.

Dialog Act	Task	Agent: Example
`Instruction`	Water Plant	*Commander*: The plant by the sink needs to be watered
	Plate Of Toast	*Commander*: please slice bread and toast 1 slice
	Plate Of Toast	*Commander*: lets make a slice of toast
`InfoObjectLocAndOD`	Plate Of Toast	*Commander*: knife is in the fridge
	Plate Of Toast	*Commander*: the clean plate is on the white table
	Clean All X	*Commander*: right cabinet under the sink
`Acknowledge`	Make Coffee	*Commander*: we are done!
	Clean All X	*Follower*: Plate is clean
	N Slices Of X In Y	*Follower*: found it
`ReqForInstruction`	Put All X On Y	*Follower*: how can I help
	Put All X On Y	*Follower*: what are my directions
	Plate Of Toast	*Follower*: what is my task today
`FeedbackPositive`	Plate Of Toast	*Commander*: good job
	Put All X In One Y	*Commander*: that's it good job
	Water Plant	*Commander*: thank you its seems to be done
`Greetings`	Make Coffee	*Commander*: Hi how are you today?
	Water Plant	*Follower*: Good day
	Boil X	*Commander*: Good morning
`ReqForObjLocAndOD`	Clean All X	*Follower*: where is the dirty cookware?
	Plate Of Toast	*Follower*: Can you help me find knife?
	Put All X In One Y	*Follower*: where is the third one?
`InformationOther`	Make Coffee	*Commander*: Don't take martini glass
	Boil X	*Commander*: You keep walking past them
	Boil X	*Commander*: That looks cooked already
`Confirm`	Put All X In One Y	*Follower*: was that everything
	Salad	*Commander*: you can see the toaster right?
	N Slices of X in Y	*Follower*: Shall I turn off the water?
`RequestOtherInfo`	Breakfast	*Follower*: how many slices of each?
	Clean All X	*Follower*: what pieces?
	Plate Of Toast	*Follower*: shall i take it to the toaster now
`MiscOther`	Sandwich	*Commander*: One sec
	Salad	*Commander*: Common!!
	Breakfast	*Commander*: Thant's my bad...Sorry
`RequestMore`	N Cooked Slices Of X In Y	*Follower*: Is there anything more I can help with?
	Salad	*Follower*: what else would you like me to do
	Clean All X	*Follower*: Any more tasks?
`OtherInterfaceComment`	Plate of Toast	*Follower*: Finish and report a bug?
	Clean All X	*Follower*: refresh the page
	Put All X On Y	*Follower*: connection is slow
`Affirm`	Water Plant	*Commander*: yes, you can use the green cup
	Breakfast	*Commander*: yes, toast the bread
	Put All X On Y	*Commander*: yes please
`NotifyFailure`	Make Coffee	*Follower*: It's not turning on the coffee.
	N Slices Of X In Y	*Follower*: tomato won't fit in those
	Sandwich	*Follower*: can't seem to grab the knife in cabinet
`Deny`	Make Breakfast	*Commander*: No don't toast the bread
	Salad	*Commander*: don't
	Plate of Toast	*Commander*: don't think so
`AlternateQuestions`	N Cooked Slices Of X In Y	*Follower*: Do I boil it or slice it?
	Clean All X	*Follower*: To the left or right of the stove?
	Make Coffee	*Follower*: This mug or the other one?
`FeedbackNegative`	Make Coffee	*Commander*: you don't have the correct mug
	N cooked Slices of X in Y	*Commander*: task not complete
	Plate of Toast	*Commander*: wrong plate

Table 6: Example utterances for Dialog act labels that could be observed in different *TEACh* tasks from *Commander*and *Follower*.

Symbol and Communicative Grounding through Object Permanence with a Mobile Robot

Josue Torres-Fonseca
Boise State University
1910 W University DR
Boise, ID 83725
josuetorresfonse@u.boisestate.edu

Catherine Henry
Boise State University
1910 W University DR
Boise, ID 83725
catherinehenry@u.boisestate.edu

Casey Kennington
Boise State University
1910 W University DR
Boise, ID 83725
caseykennington@boisestate.edu

Abstract

Object permanence is the ability to form and recall mental representations of objects even when they are not in view. Despite being a crucial developmental step for children, object permanence has had only some exploration as it relates to symbol and communicative grounding in spoken dialogue systems. In this paper, we leverage SLAM as a module for tracking object permanence and use a robot platform to move around a scene where it discovers objects and learns how they are denoted. We evaluated by comparing our system's effectiveness at learning words from human dialogue partners both with and without object permanence. We found that with object permanence, human dialogue partners spoke with the robot and the robot correctly identified objects it had learned about significantly more than without object permanence, which suggests that object permanence helped facilitate communicative and symbol grounding.

1 Introduction

Communicative grounding is the process of mediating what words mean (Clark, 1996) and *symbol grounding* is the establishment of connections between language and the perceptual, physical world (Harnad, 1990). Following Larsson (2018) that explained how symbol grounding is a side effect of communicative grounding, children who are learning their first language cannot learn symbol grounding without simultaneously being engaged in communicative grounding. Consider the following example, within the physical space of a room. A child (C) picks up a ball (B) and a caregiver (P) engages in dialogue with the child about the ball:

(1) a. (C picks up a B and looks at it)
b. P: That's a ball!
c. C: ball
d. P: Ball! Very good!

Communicative grounding happens between P and C during this interaction as P offers *ball* as a word with a semantic potential and C understands B to be an extension of *ball*. At the point (1)-b symbol grounding takes place between C and B where C links the word *ball* to the object in their hand. Communicative grounding then follows when C says *ball* and receives a positive confirmation from P, resulting in knowledge that P has experienced an interaction with C when C heard and demonstrated understanding of *ball*, and C received confirmation of understanding of the word *ball* from P.

But what happens in Example (1) when C moves their attention to a different object? It is the case that the C has grounded the word *ball* using their experience with B, and P acknowledges that C has done so, but does it matter that the object is no longer in view? Prior work explored the interplay between communicative and perceptual grounding (Chai et al., 2014; Larsson, 2018), but there is very little work on how *object permanence* plays a role in the communicative and symbol grounding process. Piaget identified object permanence in the child development process within the sensorimotor stage—a period that lasts from birth to nearly two years old (i.e., beginning before children can speak) when children largely interact with and understand the world through their sensorimotor experience (Piaget, 2013; Bremner et al., 2015). Moreover, children who are learning their first words are *egocentric* in that they have not yet developed the capability of understanding another person's point of view (i.e., of an object) (Repacholi and Gopnik, 1997). A lack of object permanence means that objects that children observe, but are then out of view no longer exist, and are separate and distinct objects if the child observes them again.[1]

[1]Lack of object permanence is the common assumption that holds for most vision and language datasets, e.g., refCOCO (Yu et al., 2016) where referring expressions to ob-

Proceedings of the SIGdial 2022 Conference, pages 124–134
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Moore and Meltzoff (1999) suggested that as early as four months, a child begins to recognize that objects have permanence even when the child is not actively observing them—an ability that the child can leverage before they start to learn language—but this knowledge has been ignored in prior research. Therefore, in this paper, we ask the question: *Does object permanence matter for communicative grounding and symbol grounding in an automated learning spoken dialogue system?* We hypothesize that it does matter, particularly for first-language acquisition in a spoken dialogue system (SDS) that has no prior exposure to language. We test our hypothesis in a human-robot interaction (HRI) task where we task human participants to interact with a robot and observe that the robot has been able to utter words in the right context. We use a survey to measure the perceptions of the human participants in order to establish that communicative grounding took place, and we measure the number of words that the robot "learned" during the interaction to determine if communicative and symbol grounding took place. We find through our experiment that symbol and communicative grounding are affected by object permanence, leading to increased user engagement and a more responsive and effective spoken dialogue system that learns word groundings as it interacts.

In the following section, we compare our work to others then explain our method for tracking object permanence using a *simultaneous localization and mapping* (SLAM) module and the the robot-ready SDS system that we used. We then explain our experiment and conclude.

2 Background & Related Work

Object permanence is a crucial milestone in cognitive development, and it has been suggested by Moore and Meltzoff (1999) that as early as four months this milestone is reached. Tomasello and Farrar (1984) shows that as infants enter the sixth stage of object permanence development (where children understand that objects completely removed from their view still exist) they start to learn relational words. A more recent study explores the development of search behavior in 7 month old infants after they guide them in understanding the effects of their actions upon hidden objects. This indicates that object permanence is crucial in searching behavior as it leads to the understanding that infants have the ability to cause hidden objects to reappear (O'Connor and Russell, 2015).

Bechtle et al. (2015) worked towards developing a sense of object permanence in robots through creating a simulated experimental setup where a robot learns how the movements of its arms (one holding a shield) affect the visual detection of an object in a scene. Although, not directly related to object permanence, Platonov et al. (2019) is more closely related to grounding as they create a SDS which is able to create a 3D model of a physical block world and answer spatial questions about it. Roy et al. (2004) also explored spatial reasoning within a physical world through the creation of a robot called Ripley which performed grounding of spatial language that could not be understood under fixed-perspective assumptions.

Of similar importance in cognitive development is communicative grounding. Researchers, notably Chai et al. (2014), have investigated how the collaborative efforts of a robot in situated human-robot dialogue affects both perceived and true grounding which involved a situated setup of objects similar to our experiment. This notion of common ground and communicative grounding has also been explored in other human-robot interaction work (Kiesler, 2005; Powers et al.; Stubbs et al., 2007, 2008; Peltason et al., 2013) and work involving human interactions with virtual agents (Pustejovsky et al., 2017). Our work extends and builds on prior work as we focus on using object permanence in a robot to improve its language learning abilities.

3 Proposed System

In this section we explain how we modeled the dialogue for language learning, integrated with robot modules. We first explain the choice of robot: Digital Dream Lab's Cozmo robot. Plane et al. (2018) showed that participants perceived Cozmo as young and with potential to learn, which is precisely the setting and perception that we want dialogue partners to have when interacting with Cozmo. Cozmo is small, has a track for movement, a lift and a head with an OLED display which allow it to display its eyes. Within the head is a small camera and a speech synthesizer (with a "young" sounding voice). For this study we make use of Cozmo's camera for object detection, track for navigation and most importantly Cozmo's built-in

jects depicted in images only offer a single visual experience (though multiple referring expressions) to the objects.

SLAM (Simultaneous Localization and Mapping) functionality for object permanence. Cozmo has no microphone, so we use an external microphone.

The system outlined in this paper uses the incremental framework ReTiCo (Michael and Möller, 2019; Michael, 2020) extended for multimodal use with Cozmo (Kennington et al., 2020), leveraging existing modules as well as the newly developed Object Permanence module. The full SDS is depicted in Figure 3. The modules include: Object Detection, Feature Extraction, Automatic Speech Recognition, Natural Language Understanding, Grounded Semantics, Action Management (Navigation & Speaking), and Object Permanence.

Object Detection The Object Detection module uses YOLO object detection (Redmon et al., 2016). The model we used was pre-trained on the MSCoco dataset (Lin et al., 2014) containing 91 object types with a total of 2.5 million labeled objects in 328 thousand images. We apply this model as a means for object region classification in order to draw bounding boxes around objects in images received from Cozmo's Camera. We discard the labels and only use the bounding box information as to avoid the use of a pretrained vocabulary since children are born without linguistic knowledge. The output of this module is the bounding box information of the objects in view to Cozmo.

Feature Extraction The Feature Extraction module uses CLIP (Radford et al., 2021) a neural network trained on a variety of (image, text) pairs. This module takes an image and bounding box information, extracts each sub-image containing each object, then passes those through CLIP's image encoder which returns image features encoded by the vision portion of the CLIP model. This module outputs a vector of size 512 for each detected object, for each frame. In our case only one object will be detected in an image, though as the robot shifts and moves, multiple frames of the object will results in multiple CLIP vector representations of that object. Taken together, the Object Detection and Feature extraction modules provide a way of isolating and extracting features from objects; children likewise have experienced objects physically (i.e., visual, tactile) before they learn that words denote objects. Both modules use models that were trained using language data which certainly affects functionality of the modules. We ignore the language aspects of the models, and leave for future work developing models (e.g., object region detection) that are trained without language data.

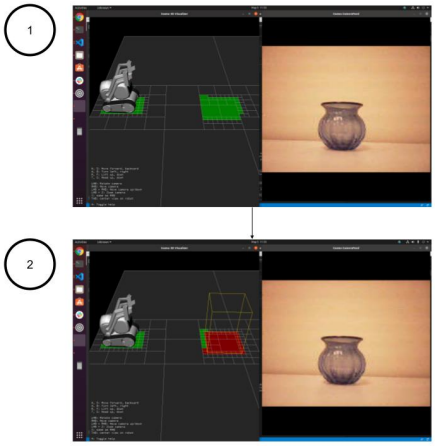


Figure 1: Visualization of the creation of a custom object in SLAM. In 1, the object is not yet observed, but in 2 the object is placed in the SLAM space.

Automatic Speech Recognition The Automatic Speech Recognition (ASR) module transcribes user speech. We use Google's speech to text API. The output is the word-level transcription.

Natural Language Understanding The Natural Language Understanding (NLU) module takes in the transcribed speech from the ASR and determines the dialogue act (i.e., *intent*) of the user using RASA (Bocklisch et al., 2017) an open source NLU library. Specifically, we use RASA to categorize user speech into 5 different dialogue acts:

- positive user feedback (e.g., *yes*)
- negative user feedback (e.g., *no*)
- where questions (e.g., *where is the can?*)
- what questions (e.g., *what is that?*)
- statements (e.g., *that is red.*)

The positive and negative user feedback is used to document the number of questions that Cozmo answered correctly and incorrectly from the participant. We categorize *where* and *what* questions so that Cozmo can differentiate between initiating finding behavior (*where* questions) and answering questions using the best known word about an object (*what* questions). This signals to our system when it should be in a state of learning how words ground into images, or whether it should be exploiting what it knows in order to locate and identify an object it has seen. This fairly simplistic ontology of dialogue acts is in line with child development; children can infer intent of positive and negative

feedback, as well as simple questions like location before they are able to speak, albeit often through extra-linguistic information such as prosody and affective displays (see (Locke, 1995), chapters 3-5). We trained RASA on 19 hand-crafted examples of positive user feedback, 10 examples of negative user feedback, 25 examples of what questions, 22 examples of where questions, and 747 examples of statements that we extracted from random samples of text from Wikipedia. We train on 747 examples of statements because statements are the most difficult to identify as there are many different variations of statements therefore requiring many training examples.

Grounded Semantics The Grounded Semantic Module performs symbol grounding by mapping heard words (though the ASR) to observed objects. The module makes use of the the Words as Classifiers (WAC) model (Kennington and Schlangen, 2015). In the WAC model, each word is represented by its own classifier trained on positive and negative examples of real-world referents and has been shown to learn words with only a few examples, which is critical for our task that is intended to mimic how children fast-map words to objects. The module learns as it "hears" a word (i.e., a recent update from the ASR module) and is currently observing an object. The WAC model associates words with the detected objects (i.e., represented as CLIP vectors) as positive examples. The module systematically trains individual logistic regression classifiers for each word as it hears words and associates those words with objects. Negative examples for training are randomly sampled from positive vectors associated with other words (the system must have heard at least two words and associated some objects with them in order to train). The classifiers are trained every time an utterance is spoken and after observing an object every 20 added frames.

The Grounded Semantics module has two modes: explore and exploit. In the explore mode, the module associates words with objects and trains the individual word classifiers as explained above. In the exploit mode, the module instead uses the recently heard words and either attempts to identify the object that is the best fit for the description or it attempts to determine which word is the best fit for an object that is currently under observation. The module's mode is determined by the speech act as signalled by the NLU module, explained above.

Action Management For dialogue (and robot action) management we use PyOpenDial (Jang et al., 2019). This module acts as a broker of the entire dialogue state to map from states to actions. In our case, the primary actions are `explore` when the robot drives around looking for objects, `find` when the dialogue partner asks about an object, `learn` when the robot should be associating words with objects, and `answer` when the robot should utter something in response to a dialogue partner's *where* or *what* dialogue act. The *explore* action is the default. In the *explore* state, Cozmo randomly drives in front of one of the 7 different objects (see Figure 4).

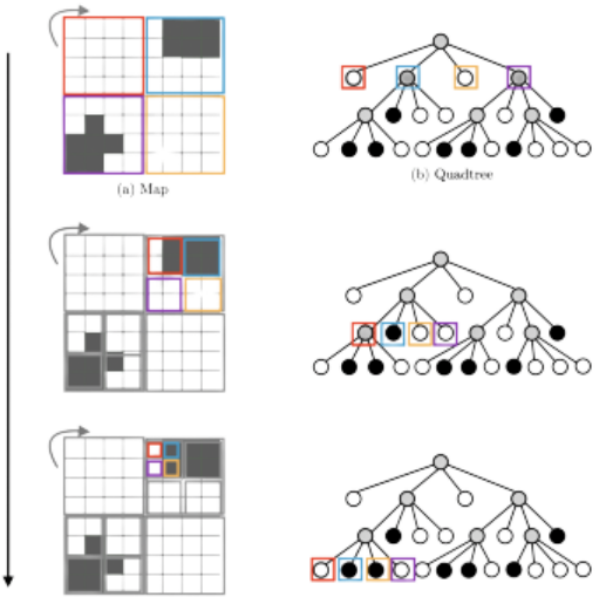


Figure 2: Quad-tree Maps are space efficient alternatives to an an occupancy map where open space is compressed into a single "unoccupied" cell. White means no cells are occupied, grey means some, and black means all. The root node represents the entire map and children are arranged in clockwise order. The first child node corresponds to the 4x4 grid in the upper left.

Object Permanence The Object Permanence module is an application of a SLAM module that is part of the Cozmo robot's functionality. The goal of the SLAM module is to track the position and location of observed objects in a 3-dimensional space. The surface that the robot can drive on is a 2-dimensional plane that the SLAM module breaks into very small cells. The SLAM module then uses *quad-tree maps* (Finkel and Bentley, 1974) to determine which cells are occupied and which ones are free. Representing the space as a quad-tree map allows SLAM to store and retrieve object lo-

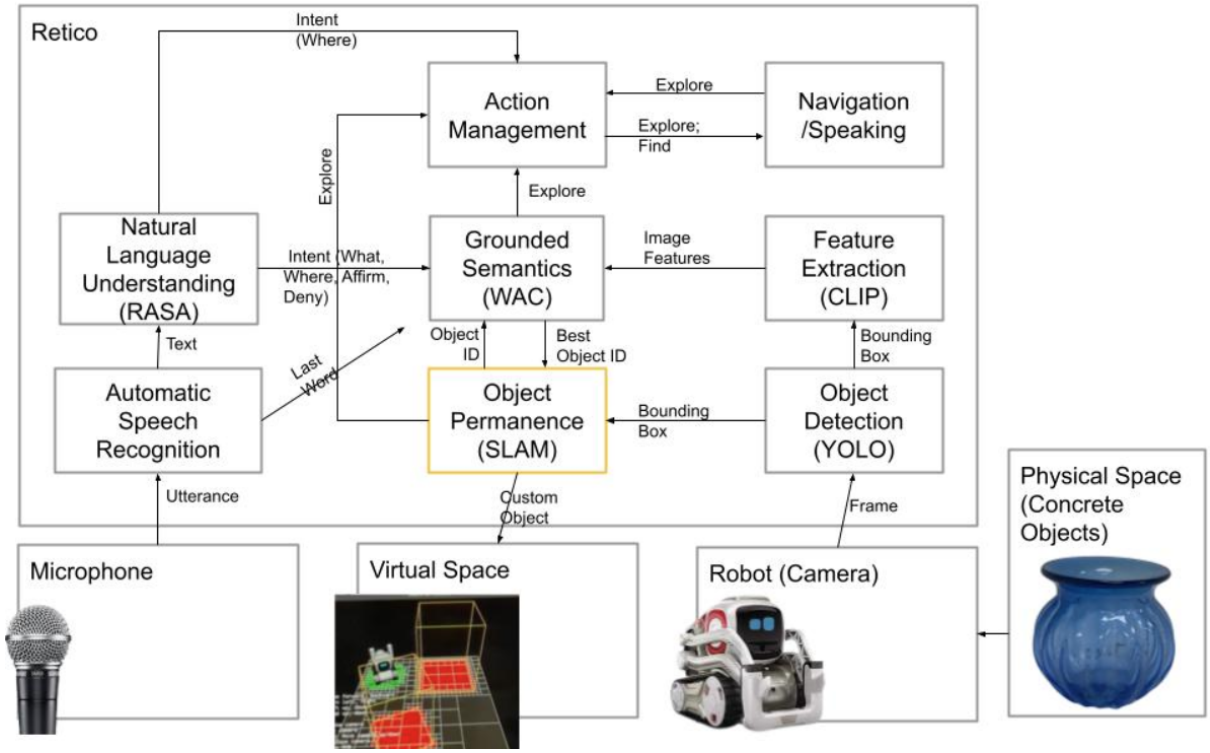


Figure 3: Schematic of our system.

cations efficiently. An example of a quad-tree map is shown in Figure 2. While we do not argue that humans use quad-tree maps for organizing object permanence, it serves as a functional approximation of what object permanence affords: the ability to remember objects and their locations.

When the system is first invoked, there is initially no history of observed objects and the robot's starting point becomes the point of reference for everything that the robot will observe. As the robot moves (i.e., drives forward or backward, and turns left or right) the SLAM module can track precisely how far and in which direction the robot has moved from its origin. The original SLAM module for Cozmo is designed to track specific objects based on a marker code (i.e., three blocks each with QR-like symbols). We extended the functionality of the SLAM module to include any object that is observed by the Object Detection module described above. We use that module's bounding box information (See Figure 1) and current observed location relative to where Cozmo is facing to infer the object's location and uniqueness. The uniqueness here is important because if the robot moves away from the object then returns to it later, the robot should be able to identify the object as one that has been seen before, not as a new object. For each new object that the robot observes, the Object Permanence module assigns a unique identifier. The unique identifier is shared with the Grounded Semantics module so it can associate specific objects (i.e., their CLIP vectors) with words that were used to describe those specific objects.

Traditional symbol grounding generally only visualizes representations of objects and associates those with referring expressions or descriptions, but the identity of objects is discarded during testing. Here, the WAC model not only learns word groundings through experience as it observes words uttered in association with observing objects, but it also uniquely identifies each object and keeps a history of its visual experience with each object regardless of how they were referred to or described. Importantly, the SLAM functionality does not just identify unique objects, it gives the robot the ability to return directly to that object without colliding with other objects because it tracks all objects that Cozmo has observed.

System Task Behavior The default action for Cozmo is `explore` which is done by randomly choosing a position at one of the drawn squares in front of all seven objects shown in 4 and centering its camera to the closest object. Once in front of an object, Cozmo waits up to 10 seconds for an utterance from a dialogue partner. If the partner utters

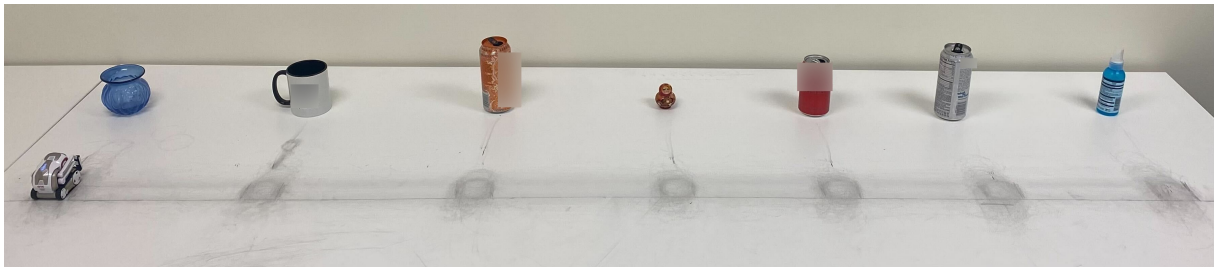


Figure 4: The seven objects used for our experiment

something, Cozmo assumes the words are about the object in view and the Grounded Semantics module learns by associating the last uttered word with the object. If no utterance is given, Cozmo moves away from the object and continues to `explore`. If the dialogue partner continues to speak, Cozmo remains in front of the object.

The `find` state is activated when the NLU detects that the dialogue partner has uttered a *where* dialogue act. For example, *where is a can?* would result in a detected `find` dialogue act. This triggers the Grounded Semantics module to find the object in its history that is the most probable fit for the description (in this case, the word *can* might ground more strongly to one object compared to others). The Grounded Semantics module then signals to the Object Permanence module to drive to and face the object with the specified identifier. Once the robot reaches the object, it utters back the description (i.e., *can*). At this point the dialogue partner can utter positive or negative feedback. When the Object Permanence module is not available (i.e., our baseline system version), Cozmo randomly explores objects and the Grounded Semantics module determines if the description fits the currently observed object using the last word in the utterance. If the probability of the model is above 0.5, then Cozmo repeats what it heard to signal that Cozmo found the object. The user can then utter positive or negative feedback.

Another dialogue act is the *what* question. If the robot is currently looking at an object, then the system assumes the *what* dialogue act is about the currently observed object and looks through its history to find the best known word for the object currently in view and utters the best known word.

In the following section, we will explain how we evaluated our SDS and whether or not Cozmo's language learning abilities improved with the use of the Object Permanence module.

4 Evaluation

In this section, we explain how we evaluated our model with human participants to determine if Cozmo performs communicative and symbol grounding more effectively with Object Permanence. We compare two versions of our system: one that did not have an access to the Object Permanence module and one that did. Our evaluation included objective measures logged by the system and by the participants used to measure symbol grounding by tracking correctly "learned" words, as well as subjective measures collected using participant questionnaires used to measure communicative grounding.

Procedure Study participants met in our lab located near Boise State University's Computer Science building. The lab is setup for the participant interaction as follows. A large table is setup with 7 objects on the table as shown in Figure 4. We chose the 7 objects to vary shape and color, but wanted to have a degree of overlap for words that might be used to describe them (e.g., *can* or *blue*).

In front of each object is a straight line drawn on the table and a box.[2] Cozmo is placed in front of the leftmost object. The microphone that feeds into the ASR module is positioned in front and to the left of the table with the objects and Cozmo. The participant stands or sits (as they prefer) at the front of the table. Cozmo is not introduced to the participant until the participant has signed a consent form and the task has been explained to them. The experimenter was present to examine the state of the robot and the microphone, answer any questions the participant may have, and troubleshoot any problems that arose. The experimenter was permitted to offer a constrained set of coaching tips to the participant during the experiment, given the participant needed a reminder of their task or

[2]The line and box does not affect Cozmo, it is just there to help the participant adjust Cozmo when needed.

the instructions. Following each interaction with Cozmo, the participant was instructed to complete a questionnaire. Following the completion of the experiment and surveys, the participant was paid $10. We recruited 24 participants to interact with Cozmo for two twenty-minute periods over the course of a single session. Most study participants recruited were from the Boise State University Department of Computer Science. 18 of the participants were male; 6 were female. The entire time for each participant was approximately one-hour.

After signing the informed consent, Cozmo was introduced to the participant, with the following explanation; (1) Cozmo has a camera that can see the world; (2) Cozmo has a microphone and can hear them; (3) Cozmo doesn't know anything, but would like to know more about the world; (4) for the next 20 minutes, it is your job to teach Cozmo as many words as they can, about the seven objects in front of Cozmo; (5) Cozmo will move in front of an object. If Cozmo does not hear you speak he will move on to a new object. If Cozmo does hear you speak, then it will observe the object and repeat the word he learned. The word Cozmo learned will always be the last word you spoke. Do not teach Cozmo any more words until it repeats this word. For every word you teach it, you can write it down so you can keep track; (6) if Cozmo is not on the square in front of the object when he moves to an object you must readjust it to that square; (7) After teaching Cozmo about two different words for two different objects you can and should ask it *what* and *where* questions to check his knowledge; (8) For every question he gets right answer "yes" and put down a tally mark to record a correct answer for every question he gets wrong answer "no" only; (9) Only speak to Cozmo when he is in front of an object.

We used an A/B design, meaning that each participant went through the same procedure twice, once with Cozmo having access to Object Permanence and once without access. To mitigate priming effects, the order in which the test condition was presented was alternated.

Two System Versions The test condition is the system version that had access to Object Permanence and is explained in Section 3. The baseline point of comparison for this study was a system that did not have access to the Object Permanence module. The overall functionality of the baseline system was the same as the system with access to Object Permanence, except that the system could not track and locate objects when participants asked them to. The Grounded Semantics module in this case only performs traditional symbol grounding between words and visual representations—not specific objects. This meant that the robot behavior when a `find` dialogue state was entered (i.e., after a *where* dialogue act from the participant) was different: instead of moving directly to the identified object, the robot would move towards a random object one at a time and check each object to determine if they matched the description. If an object did match, the robot would repeat the description to the participant, who then in turn offered positive or negative feedback. Under the best circumstances for the baseline system, the robot would randomly move towards an object that fit the description on the first attempt. But if the first object did not fit the description, then the robot moved towards a different object and repeated until an object matched the description. To give the baseline version a higher chance of the robot actually finding the objects, the objects were placed in a line and the robot systematically drove directly to a randomly selected object. This was designed to give the baseline system version some degree of the object permanence functionality as a stronger point of comparison.

Metrics All module communication is logged using the Platform for Situated Intelligence (Bohus et al., 2017). We specifically track the number of utterances made by the participants, including positive and negative feedback, and the number of questions asked. The participant themselves keep track of the number of questions Cozmo correctly answers (i.e., if Cozmo correctly identified an object). These metrics act as a way to measure symbol and communicative grounding, as well as engagement (i.e., more utterances means more engagement).

We evaluate the robot based on questionnaire responses filled out by the participants following each interaction to establish that communicative grounding took place. We used the Godspeed Questionnaire (Bartneck et al., 2009), a 5-point Likert-scaled questionnaire with 24 questions using negative (left side) to positive (right side) ratings of a robot's anthropomorphism, animacy, likeability, and perceived intelligence. In addition to the Godspeed questions, we asked the participants the following questions to further ascertain their perceptions of our system and robot (some items have boldface text to link them with results):

(Mean / std. dev)	baseline	with obj. perm.	p-value
Heard Words	15.9 / 3.9	18.8 / 4.7	0.02
Questions Asked	10.7 / 3.2	20.0 / 6.7	3.7e-7
% Correct	70.0 / 22.2	82.7 / 12.1	0.02

Table 1: The effect of object permanence on a language acquisition task

Interesting	0.0048
Spend more Time	0.049
Responsive	0.083
Intelligence	0.10

Table 2: Statistical Significance between values with and without Object Permanence using a t-test.

(Mean / std. dev)	1st Interaction (A)	1st Interaction (B)	2nd Interaction (A)	2nd Interaction (B)
Heard Words	19.4 / 4.7	16.2 / 2.7	18.3 / 5.0 (0.56)	15.6 / 4.9 (0.72)
Questions Asked	16.4 / 4.0	12.0 / 3.5	23.0 / 7.5 (0.01)	9.7 / 2.7 (0.13)
% Correct	83.2 / 11.5	74.7 / 20.4	82.1 / 13.0 (0.84)	64.9 / 23.7 (0.30)

Table 3: The effect of initial setting on a language acquisition task (A is for with Object Permanence and B is for without. Furthermore, the values in parentheses near the values for the second interaction represent the p-values between the values in 1st Interaction compared to 2nd Interaction for A and B)

- How attached to the robot did you feel?
- How **interesting** was the robot to interact with?
- Would you like to **spend more time** with the robot?
- How many years old do you think the robot is (in terms of its behavior)?

Results Table 1 shows the effect object permanence has on Cozmo's language acquisition abilities.[3] It is clear that with object permanence, Cozmo is perceived to learn language better than without object permanence as shown by the statistical significance values. This suggests that object permanence does appear to have an affect on symbol grounding especially as Cozmo not only hears more words on average per participant with the test condition than without, his accuracy in answering questions also increases by approximately 13%.

Relating to participant perceptions of the robot and interaction, we find that overall the mean values for the ratings were higher for the test condition than the baseline except for three questions which relate to kindness, and feelings of calmness and interest at the beginning of the interaction. Therefore, showing that overall, the test condition positively influenced users' perception of Cozmo. Furthermore, we observe that with object permanence, participants believed that Cozmo learned better than without, as seen by the overall higher intelligence and responsiveness scores in Figure (1), though note that the difference in perceived intelligence is not significant, which tells us that the baseline system was still viewed positively and therefore provided a high point of comparison.

Participants on average estimated Cozmo's age with the test condition at 3.5 years of age compared to 2.6 years of age with the baseline, suggesting that Cozmo was perceived to be more intellectually advanced with the test condition, but still an early language learning child, which also tells us that the robot did not exhibit behaviors that participants perceived as too advanced for our task. We also observe higher responsiveness in the object permanence version which likely results from participants observing that Cozmo answered questions quickly and with high accuracy, suggesting that communicative grounding was better with the object permanence version (see Appendix for more results comparing perceived Intelligence and Responsiveness).

Finally, ratings for interest and desire to spend more time with the robot are significantly higher with object permanence than without. This is especially evident when observing that the mean value for interest at the end of the interaction is 4.7 with the test condition and 4.2 without; the average increase in interest from the beginning of the interaction for the test condition is 0.54 as compared to 0.83 without. Furthermore, using questions asked as a measurement for engagement (since it shows active interest in what Cozmo is learning) we observe that with Object Permanence, Cozmo is asked

[3]Nine interactions had to be restarted due to unexpected events (e.g., Cozmo rolled off the table) which affected the SLAM map and learned words, but this happened at roughly the same frequency for both settings. Cozmo also picked up his own voice in the microphone in both settings, but this also happened at roughly the same frequency for both settings so we decided to leave it as part of the data.

approximately 9 more questions than without showing that object permanence has a significant effect on engagement (see Table 3). This is crucial, because the interaction itself needs to motivate human participants to "buy into" the robot's language learning by spending time and effort helping it learn. See also Figure 5 in the Appendix for more results.

5 Conclusion

We conducted an experiment with twenty-four participants who performed a language acquisition task with Cozmo both with and without object permanence. We analyzed our results by comparing the participants' survey responses to measure communicative grounding and number of words heard, questions asked, and percent of questions answered correctly to measure symbol grounding between the experimental and control interactions. We found that a robot with object permanence resulted in improved communicative and symbol grounding due to stronger engagement from the participant and a higher percentage of correct answers from Cozmo. User perceptions of Cozmo with object permanence also greatly improved overall. This indicates that object permanence does in fact have a positive affect on communicative and symbol grounding. Our findings suggest that an understanding of object permanence is a necessary component of any spoken dialogue system built to reach the potential of natural dialogue between humans.

Acknowledgements Thanks to the anonymous reviewers for their very useful feedback. This material is based upon work supported by the National Science Foundation under Grant No. 2140642.

References

Christoph Bartneck, Dana Kulić, Elizabeth Croft, and Susana Zoghbi. 2009. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. *International Journal of Social Robotics*, 1(1):71–81.

Sarah Bechtle, Guido Schillaci, and Verena V Hafner. 2015. First steps towards the development of the sense of object permanence in robots. In *2015 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 283–284.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management.

Dan Bohus, Sean Andrist, and Mihai Jalobeanu. 2017. Rapid Development of Multimodal Interactive Systems: A Demonstration of Platform for Situated Intelligence. In *Proceedings of ICMI*, Glasgow, UK. ACM.

J. Gavin Bremner, Alan M. Slater, and Scott P. Johnson. 2015. Perception of object persistence: The origins of object permanence in infancy. *Child Development Perspectives*, 9(1):7–13.

Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. 2014. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 33–40, Bielefeld, Germany.

Herbert H Clark. 1996. *Using Language*. Cambridge University Press.

Raphael A Finkel and Jon Louis Bentley. 1974. Quad trees a data structure for retrieval on composite keys. *Acta informatica*, 4(1):1–9.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D*, 42(1-3):335–346.

Youngsoo Jang, Jongmin Lee, Jaeyoung Park, Kyeng-Hun Lee, Pierre Lison, and Kee-Eung Kim. 2019. PyOpenDial: A python-based Domain-Independent toolkit for developing spoken dialogue systems with probabilistic rules. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 187–192, Hong Kong, China. Association for Computational Linguistics.

Casey Kennington, Daniele Moro, Lucas Marchand, Jake Carns, and David McNeill. 2020. rrSDS: Towards a robot-ready spoken dialogue system. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 132–135, 1st virtual meeting. Association for Computational Linguistics.

Casey Kennington and David Schlangen. 2015. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 292–301, Beijing, China. Association for Computational Linguistics.

S Kiesler. 2005. Fostering common ground in human-robot interaction. In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pages 729–734.

Staffan Larsson. 2018. Grounding as a Side-Effect of grounding. *Top. Cogn. Sci.*

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

John L Locke. 1995. *The Child's Path to Spoken Language*. Harvard University Press.

Thilo Michael. 2020. Retico: An incremental framework for spoken dialogue systems. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 49–52, 1st virtual meeting. Association for Computational Linguistics.

Thilo Michael and Sebastian Möller. 2019. Retico: An open-source framework for modeling real-time conversations in spoken dialogue systems. In *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2019*, pages 134–140. TUDpress, Dresden.

M. Keith Moore and Andrew N. Meltzoff. 1999. New findings on object permanence: A developmental difference between two types of occlusion. *British Journal of Developmental Psychology*, 17(4):623–644.

Richard J O'Connor and James Russell. 2015. Understanding the effects of one's actions upon hidden objects and the development of search behaviour in 7-month-old infants. *Dev. Sci.*, 18(5):824–831.

Julia Peltason, Hannes Rieser, Sven Wachsmuth, and Britta Wrede. 2013. On grounding natural kind terms in human-robot communication. *KI-Künstliche Intelligenz*, 27(2):107–118.

Jean Piaget. 2013. *The construction of reality in the child*, volume 82. Routledge.

Sarah Plane, Ariel Marvasti, Tyler Egan, and Casey Kennington. 2018. Predicting perceived age: Both language ability and appearance are important. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 130–139, Melbourne, Australia. Association for Computational Linguistics.

Georgiy Platonov, Benjamin Kane, Aaron Gindi, and Lenhart K Schubert. 2019. A spoken dialogue system for spatial question answering in a physical blocks world. *arXiv:1911.02524 [cs]*.

Aaron Powers, Adam Kramer, Shirlene Lim, Jean Kuo, Sau-Lai Lee, and Sara Kiesler. Common ground in dialogue with a gendered humanoid robot. Accessed: 2022-5-2.

James Pustejovsky, Nikhil Krishnaswamy, Bruce Draper, Pradyumna Narayana, and Rahul Bangar. 2017. Creating common ground through multimodal simulations. In *Proceedings of the IWCS workshop on Foundations of Situated and Multimodal Communication*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

B M Repacholi and A Gopnik. 1997. Early reasoning about desires: evidence from 14- and 18-month-olds. *Dev. Psychol.*, 33(1):12–21.

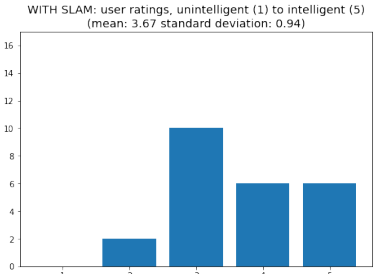
D. Roy, Kai-Yuh Hsiao, and N. Mavridis. 2004. Mental imagery for a conversational robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(3):1374–1383.

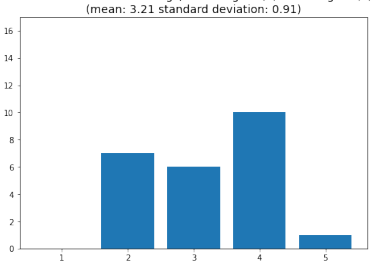
Kit Stubbs, David Wettergreen, and Illah Nourbakhsh. 2008. Using a robot proxy to create common ground in exploration tasks. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction, HRI 2008, Amsterdam, The Netherlands, March 12-15, 2008*, pages 375–382. unknown.

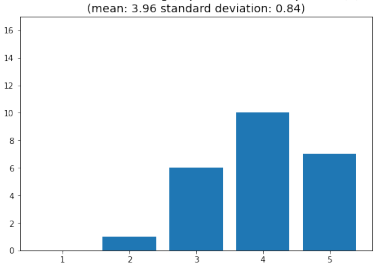
Kristen Stubbs, Pamela J Hinds, and David Wettergreen. 2007. Autonomy and common ground in Human-Robot interaction: A field study. *Intelligent Systems, IEEE*, 22(2):42–50.

Michael Tomasello and Michael Jeffrey Farrar. 1984. Cognitive bases of lexical development: object permanence and relational words*. *J. Child Lang.*, 11(3):477–493.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. 2016. Modeling context in referring expressions. In *Computer Vision – ECCV 2016*, pages 69–85, Cham. Springer International Publishing.







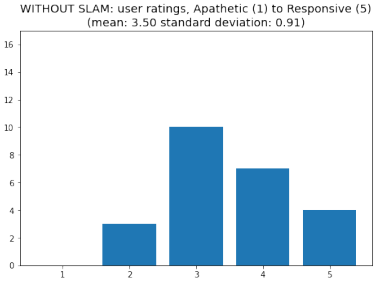


Figure 5: Engagement ratings for Cozmo with and without object permanence

Figure 6: Intelligence and Responsiveness ratings for Cozmo with and without object permanence

Towards Personality-Aware Chatbots

Daniel Fernau[†] **Stefan Hillmann**[†]
Nils Feldhus[◇★] **Tim Polzehl**[†◇★] **Sebastian Möller**[†◇]
Technische Universität Berlin[†]
German Research Center for Artificial Intelligence (DFKI)[◇]
{firstname.lastname}@dfki.de

Abstract

Chatbots are increasingly used to automate operational processes in customer service. However, most chatbots lack adaptation towards their users which may results in an unsatisfactory experience. Since knowing and meeting personal preferences is a key factor for enhancing usability in conversational agents, in this study we analyze an adaptive conversational agent that can automatically adjust according to a user's personality type carefully excerpted from the Myers-Briggs type indicators. An experiment including 300 crowd workers examined how typifications like extroversion/introversion and thinking/feeling can be assessed and designed for a conversational agent in a job recommender domain. Our results validate the proposed design choices, and experiments on a user-matched personality typification, following the so-called law of attraction rule, show a significant positive influence on a range of selected usability criteria such as overall satisfaction, naturalness, promoter score, trust and appropriateness of the conversation.

1 Introduction

In today's rapidly emerging technology-driven world, chatbots are becoming a more significant factor in customer interaction. Next to voice-driven assistants, text-based conversational agents—commonly known as chatbots—have attracted significant attention in recent years. Chatbots are designed to interact with humans using natural language and are commonly used on messaging platforms and websites (Dale, 2016; Gnewuch et al., 2018). With recent advancements in the field of artificial intelligence (AI), organizations are starting to realize the potential of chatbots to automate their customer service operations and hence reduce costs (Adam et al., 2020). Furthermore, it was predicted that 80% of organizations would have deployed a chatbot by 2020 (Sandbank et al., 2017). However, the quality of today's systems does not seem to meet customer expectations (Gnewuch et al., 2018). A key obstacle preventing most chatbots from being successful is that the interaction lacks humanness and naturalness (Schuetzler et al., 2014; Gnewuch et al., 2018). Several studies have investigated social cues and their positive effect on users' perceived social presence, trust, enjoyment, and usage intentions (Zumstein and Hundertmark, 2017; Ahmad et al., 2020). However, it has also been shown that social cues may have a negative effect that ends up irritating the user (Louwerse et al., 2005).

Studies about the nature and quality of human-machine interactions have identified personality as an essential factor for this issue (Chaves and Gerosa, 2021). Personality is a stable pattern that provides a measure for a person's behavior (und Gregory J Feist, 2002). Traditionally, personality is assessed by questionnaires; current approaches, however, make it possible to use human-generated data from social media or online forums (Boyd and Pennebaker, 2017). A person's language can provide information about the user's personality (Pennebaker and King, 1999; Boyd and Pennebaker, 2017; John et al., 1988).

To address these challenges and leverage modern technologies, the development of a personality type-indicator adaptive chatbot that automatically adapts to a user's presumed personality type is proposed in this work. The studies analyzes the impact of the so-called "law of attraction," according to which users reported higher communication interaction, human-likeliness, preference, and friendliness when interacting with a chatbot that has equal personality traits (Ahmad et al., 2020; Park et al., 2012). However, the studies introduced did not produce statistically significant results except for Ahmad et al.'s (2020) work (Ahmad et al., 2020). Their study did not require full interaction with an

★ Corresponding authors

Proceedings of the SIGdial 2022 Conference, pages 135–145
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

applied chatbot, but rather examined the perception of different personalities in a chatbot by showing their participants screenshots of the interactions. In our empirical quantitative user study, we therefore evaluate how adapted personality types are perceived by chatbot users for the domain of a job recommender chatbot and whether or not personality type-based adaptation can lead to higher overall satisfaction, usability, trust, and appropriateness.

Furthermore, there exist very few works about design criteria for how to realize personality in terms of chatbot design. This paper seeks to contribute to this area by giving design implementation details.

2 Related Work

2.1 Personality and MBTI Typification

Looking in the psychologically motivated literature of personality assessment and analysis the predominantly used model is the so called five factor model (FFM) (McCrae and Costa, 1987; McCrae and John, 1992). However, and despite overt scientific criticism, e.g. (Pittenger, 1993; Boyle, 1995), when looking into concurrent practical application outside the scientific community the application of Myers-Briggs Type Indicator (MBTI) as a pre-employment assessment in career and job seeking processes, all originating to (McCaulley and Martin, 1995), has gained substantial popularity. In this work, we therefore adopt and extend the principles of MBTI typification into a job recommender chatbot interaction while taking good care of MBTI validity and type indicator selection for our experiments. MBTI is a personality theory classifying people into the combination of four types resulting in one of 16 distinct classifications (McCrae and Costa, 1987), rather than continuous dimensions native to FFM. This distinction leads to a difference in the meaning of each combination. The MBTI consists of four dichotomies: Extroversion (E) vs. Introversion (I), Sensing (S) vs. Intuition (N), Thinking (T) vs. Feeling (F), and Judging (J) vs. Perceiving (P) (Myers-Briggs et al., 1998). (McCrae and Costa Jr, 1989) examined the degree of empirical convergence between the Big 5 and the MBTI. Their results show that each MBTI type is correlated to at least one Big 5 trait. The largest study in Furnham (1996) shows large correlations[1] for I/E with Extroversion, and P/J with Conscientiousness and medium correlation between N/S with Openness and T/F with Agreeableness.

Introversion	**Extroversion**
problem talk	pleasure talk
single topic	many topics
few semantic errors	many semantic errors
few self-references	*many self-references*
formal	*informal*
many tentative words	*few tentative words*
many nouns, *adjectives*	*many verbs*, *adverbs*
prepositions	*pronouns*
many words per sentence	*few words per sentence*
many articles	few articles
many negations	few negations
few positive words	*many positive emojis*
less emojis	*few negative emotions*
many negative emotions	*affiliative humor*
(*bad emojis*)	

(cues in *italic* were used in our study)

Table 1: Overview of linguistic cues for I/E as by (Ruane et al., 2020; Mairesse et al., 2007; Pennebaker and King, 1999; Mehl et al., 2006; Scherer, 1979; Furnham, 1990; Gill and Oberlander, 2002)

2.2 Link between Personality and Language

According to John et al. (1988), a modern approach to infer personality is inferring it from language based on the lexical hypothesis (John et al., 1988). Over the years, subsequent research has refined this theory. As a system, the lexical hypothesis is considered to be a general approach with implications for cross-cultural diversity, cognitive theories, and other areas of psychology (Digman, 1990). The hypothesis states that each person has different opinions and preferences which are expressed in a person's language (John et al., 1988). Thus, in language analysis based on personality vocabulary, one should use a clearly defined list of the most important characteristics (John et al., 1988). Which characteristics to utilize to design a chatbot's personality is explained in the following.

Prior work has mapped linguistic cues for each of the personality traits (Boyd and Pennebaker, 2017; Pennebaker and King, 1999; Mairesse et al., 2007; Ruane et al., 2020). As already indicated, the application of these cues in the literature is predominantly derived from the FFM, as research lacks linguistic cues based on the MBTI (Furnham,

[1] According to Cohen (1988), a correlation > 0.1 is considered as low, > 0.3 as medium, and > 0.5 as large (Cohen, 1988)

Thinking	Feeling
swearing	*longer words*
anger	*shorter sentences*
negations	*positive emotions*
references to facts	*cheerful*
less mentions to emotions	*many self-references*

(cues in *italic* were used in our study)

Table 2: Overview of linguistic cues for T/F as by (Ruane et al., 2020; Pennebaker and King, 1999)

1996).

Selecting carefully our experimentation scope, this study focuses on two of the four dichotomies, namely I/E and T/F, for a essential reasons. Both dichotomies show respective correlations to extroversion and agreeableness offering well established linguistic cues (Ruane et al., 2020; Mairesse et al., 2007; Pennebaker and King, 1999; Mehl et al., 2006; Scherer, 1979; Furnham, 1990; Gill and Oberlander, 2002) drawn from the FFM. The I/E dichotomy has the strongest correlation to the FFM's extroversion scale. Among all four MBTI dichotomies, however, with the correlation to the FFM being the lowest between T/F and agreeableness, there is no significant difference to the other scales when compared with McCrae and Costa's study (1989) (McCrae and Costa Jr, 1989). Table 1 and 2 show the overview of linguistic cues for extroversion and agreeableness as adapted to I/E and T/F for the presented study.

Obtaining MBTI types is typically done by questionnaires, e.g. Form M (93 items). Due to availability and transparency reasons, this study excerpts from the open-source Open Extended Jungian Type Scales (OEJTS) questionnaire (Jorgenson, 2015) provided from openpsychometrics[2].

2.3 The Law of Attraction

The law of attraction is the central theory to adapt a chatbot in order to achieve greater usability. According to this theory, people seek out those similar to them and prefer to interact with people with similar traits. As explained by (Infante et al., 1997), the perceived similarity is the degree to which we believe someone's characteristics are similar to our own. These characteristics can include several factors such as demographics, political views, and personality. Many studies in psychology and communication have confirmed this rule (Blankenship et al., 1984; Nass and Lee, 2001). Originating from the observations of Human-Human Interaction (HHI), this concept is frequently applied to Human-Computer Interaction (HCI) as well.

Transferred to HCI, the law of attraction states that a user prefers to interact with a computer that has matched personality types rather than mismatched ones. When matched, information from the computer has also been rated as better and more trustworthy (Zumstein and Hundertmark, 2017). Specifically for the Big 5 theory, a study found that for a sub-dimension of the trait extroversion, dominant people prefer to interact with a dominant counterpart, and vice versa for the submissive trait (Moon and Nass, 1996). Several other studies in the field of HCI also confirmed the law of attraction (Ahmad et al., 2020; Smestad, 2018; Lee and Nass, 2005). However, some studies do not support the law of attraction in the area of HCI (Isbister and Nass, 2000; Liew and Tan, 2016), suggesting that the applicability may also depend on a concrete scenario or application. A supporting argument comes from the field of Human-Robot Interaction (HRI), e.g. the analysis of task dependency in (Tay et al., 2014).

3 Chatbot Design

Our personality-adaptive chatbot prototype is based on the Microsoft Azure Bot Framework and is built in the browser, allowing it to be embedded into various channels. Depending on the input personality, the respective conversation tree is activated for the task of job recommendation divided into two sub-dialogs. The first sub-dialog generally greets the user, while the second one asks job-related questions to give a personality-based recommendation.

To design the chatbot's personality type, the previously introduced linguistic cues were used. Table 3 and 4 show the applied cues including their degree for the four differently designed characters of the chatbot. For the analysis of the chatbot responses, both the Python library *spaCy*[3] and the service *Count Wordsworth*[4] were utilized. In contrast to other studies, this table enhances the transparency of the degree of linguistic cues applied, whereas related work oftentimes does not include a description of exact design choices.

[2]The Open Extended Jungian Type Scales (OEJTS) can be accessed under: https://openpsychometrics.org/tests/OEJTS developed by Jorgenson (Jorgenson, 2015)

[3]https://spacy.io

[4]https://countwordsworth.com

	ET	EF	IT	IF
Manipulating E and I				
Percentage of I/we	3.81%	3.83%	2.41%	2.67%
I, me	12	11	11	10
first person	44	41	32	30
verbs	54	44	64	53
verbs by WR	17.14%	15.33%	14.04%	14.17%
adverbs	19	20	24	17
adverbs by WR	6.03%	6.97%	5.26%	4.55%
pronouns	51	48	66	48
pronouns by WR	16.19%	16.72%	14.47%	12.83%
affiliative humor	1	1	0	0
informal words	18	13	1	2
Total words	315	287	456	374
articles	6	4	36	29
articles by IR	1.90%	1.39%	7.89%	7.75%
nouns	41	23	80	69
nouns by IR	13.02%	8.01%	17.54%	18.45%
adjectives	13	18	39	30
adjectives by IR	4.13%	6.27%	8.55%	8.02%
prepositions	35	31	83	62
prepositions by IR	11.11%	10.80%	18.20%	16.58%
tentative words*	2	1	8	9
third person (formality)	5	5	11	7
Manipulating T and F				
words per sentence	8.75	8.46	12.32	11.32
emojis emotion negative	2	0	2	0
words related to swearing/anger	6	0	3	0
aggressive humor	0	0	1	0
references to facts	2	0	2	0
average length of words	3.98	4.11	4.54	4.70
words related to emotion	8	13	5	11
emojis emotion positive	7	25	0	1
emojis neutral	13	26	0	0
neutral humor	0	0	0	1

WR: word ration, IR: interaction ratio, *e.g. *would/could*

Table 3: Linguistic cues applied for personality expression

Overall, due to the short nature of the interactions, the metrics concerning word counts, sentence length, and word length were hard to manipulate when designing the messages, as there were too many dependencies on other metrics such as references to facts.

4 User Study

The experiment consists of five steps. (1) Users fill out a short 12-item personality self-report according to *OEJTS*. (2) Participants interact with our chatbot, with random assignment of matched or mismatched personality type. (3) Users assess first interaction by nine usability items. (5) Participants again interact with our chatbot, this time seeing the alternative personality type as in step 2. (6) User again assess nine usability items plus questions on preference of one version over the other.

Report:	ET	EF	IT	IF
Metrics of linguistic cues where T higher than F				
words per sentence	8.75	8.46	12.32	11.32
emojis emotion negative	2	0	2	0
words related to swearing/anger	6	0	3	0
aggressive humor	0	0	1	0
references to facts	2	0	2	0
Metrics of linguistic cues where F higher than T				
average length of words	3.98	4.11	4.54	4.70
words related to emotion	8	13	5	11
emojis emotion positive	7	25	0	1
emojis neutral	13	26	0	0
neutral humor	0	0	0	1

Table 4: Overview of the metrics of linguistic cues to design personality for T/F.

The first part of the study is a survey is a 12-item personality self-report based on the OEJTS. For this study, each of the nine highest scoring items on the I/E and the T/F scales are used in this experiment. Additionally, each dichotomy has further been divided into six items of the E/I types and six items of the T/F types. The selected items were assessed by using a five-point Likert scale in between"Strongly agree," "Agree," "Neither agree nor disagree," "Disagree," and "Strongly disagree."

Depending on the users personality type, two chatbots were automatically selected to be tested in step 2 and step 5, of which one is designed to be perceived the same personality type as the user (matched), whereas the other one represents the opposite option settings (mismatched).For example, if a user is classified as EF (extroverted-feeling), they interacted with both an EF and an IT designed chatbot, in random order. The extroverted chatbot was named Carla and the introverted one was named Sophia to achieve the effect that users are more likely to share personal information if the chatbot appears to be female (Toader et al., 2020).

The topic of the interactions in step 2 and 5 is to chat about personal and job-related preferences to recommend a suitable job. The job recommendations given by the chatbot in the end of the conversation are hand crafted and based on the personality of the user. Note that we do not analyze the performance of any recommendation accuracy, nor the users' acceptance towards it. In this work, we focus on the impact of personality on the usability of the interaction explicitly. In more detail, the

conversation starts with some general questions regarding the name, origin, and personal preferences. Afterwards, the chatbot commences asking about job-related preferences. Three questions are asked that are based on additional items of the OEJTS. For example, one item of the OEJTS to measure extroversion assesses whether the user "works best in groups" or "works best alone."

Further, the chatbot is designed to be between the edges of an intra- and an interpersonal chatbot within a closed domain, offering limited functionality (Nimavat and Champaneria, 2017). Hence, the chatbot only allows the user to answer the questions instead of providing functionality that answers custom questions of the user. This limitation was explicitly clarified at the beginning of the survey to avoid false expectations. Moreover, the users have also been instructed of another limitation of the current state of chatbot prototype implementation, namely that writing multiple messages is not supported. This means that all information has to be put into a single message.

The usability questionnaire applied consists of nine items that are asked after each chatbot interaction: two items that compare both chatbots with each other and five general items about the participants. First, the nine items that are asked directly after each conversation with the chatbot are introduced. These items are split into four items derived from ITU telecommunication standardization sector (ITU-T) Recommendation P.851 (Rec, 2003), while the other five items are custom-designed. Adapted to the personality domain, four items were selected that are related to the following factors: acceptability, naturalness, and promoter score. For these items (among others), it was demonstrated that acceptability and naturalness are well generalized (Möller et al., 2007). The personality factor from ITU-T was not suitable for the experiments at hand due to the strong focus on personality type differentiation of this study. Hence, five custom items were designed to measure whether the design choices applied could be perceived by the participants when interacting with the different chatbots. These nine usability items were assessed using the same five-point Likert scale from above. In addition, two items were designed to directly compare Carla (extroverted) and Sophia (introverted) head-to-head. The first item assesses which chatbot is being perceived as more adapted toward the users' preferences, while the second asks for the general preference when comparing both directly. For both items, users had the option to choose Carla, Sophia, both, or none. Eventually, five profiling questions were asked at the end of the survey regarding gender, age range, experience with chatbots, native language, and their current profession. All items are shown in **Appendix A**, also including the items used for comparison and general profile data.

4.1 Participants

300 participants were recruited using the the Crowdee (Naderi et al., 2014) crowdsourcing platform [5] across the U.S., Great Britain, and Australia. Participants were paid equally by minimum floor wage based on the estimated work duration of the task at hand.

From the general profile items we see, that 90% of the participants were English native speakers.52% of the participants were women and 46% men, while a minority was diverse (1%) or did not like to share their gender (1%1). All participants were older than 18 years, and the distribution among age classes was as follows: 18–25 (20%), 26–35 (36%), 36–45 (24%), 46–55 (15%), and <55 (5%). Regarding their experience with chatbots, a minority of 13% had never been in touch with a chatbot before. Moreover, 5% use a chatbot on a daily basis, while 20% interact with one at least monthly and 62% occasionally. In total, out of 300 crowd workers who participated, 266 valid responses can be considered. 32 participants did not complete the interactions or the questionnaires, or interactions could not successfully be logged. Furthermore, 2 participants were excluded from the study as outliers due to their scores being three times higher than the interquartile range.

From a preliminary analysis of the qualitative feedback we feel confident that the participants could solve the task as expected and generally enjoyed the study. The overall tone in qualitative feedback was positive, e.g. "Carla was the best one, [...] It was cool but scary.", "Sophia was great. Sounded like a real person was on the other end.", or "It was pretty fun speaking with the first one [extroverted], she was way more accurate with her job recommendations than Sophia.".

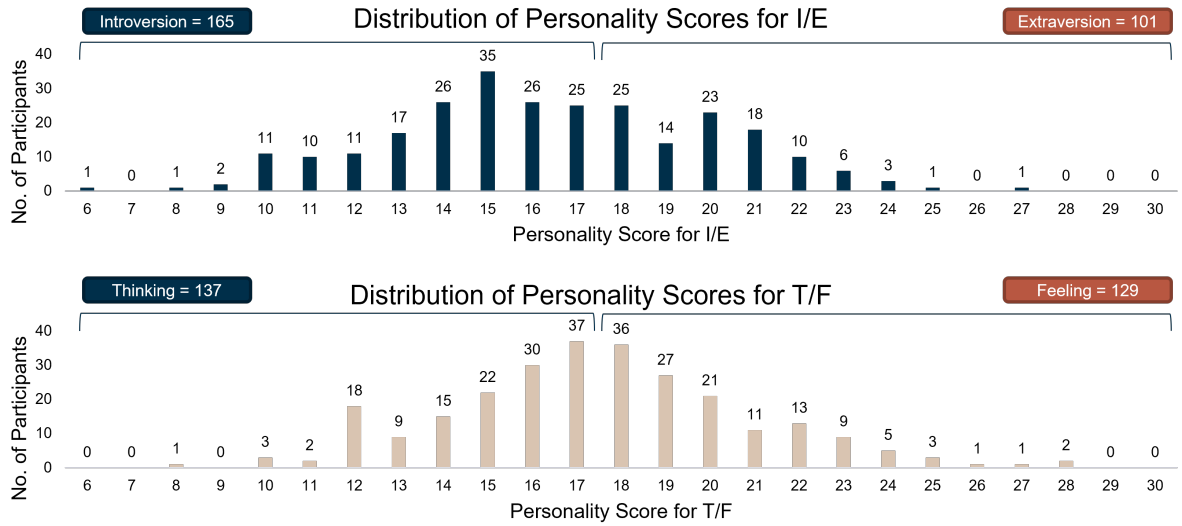


Figure 1: Distribution of personality type scores and counts, including classification boundaries; top for E/I, bottom for T/F dichotomies.

5 Results

5.1 Personality Type Distribution

Figure 1 shows the distributions of personality scores measured with the *OEJTS*. Both bar charts show the number of participants by personality score between 6 (low = **I**ntroversion or **T**hinking on the left) and 30 (high = **E**xtraversion or **F**eeling on the right), and the equal space binning threshold of 18 to differentiate the values into binary classes. The upper chart regarding I/E shows that the ratio between I and E classified participants is 62:38. More balanced is the distribution of T/F with a ratio of approximately 51:49 in the lower bar chart. All types are represented by at least 47 participants, with ET being the minority with 18% (47 participants), followed by EF with 20% (54). Among the introverted participants, IF represents 28% (75) and a majority of 34% are classified as IT (90). As no class is equal to or greater than twice the size of another, there are no imbalances in the overall distribution.

5.2 Results for the Law Of Attraction

In order to analyze the effect of the law of attraction, a one-sided t test was used to examine the statistically significant difference between the matched and mismatched scores of Q1-9 (see Appendix **??**). The test for significance was done at the level of $\alpha = 0.05$ for the following t-tests. It was not necessary to apply the Bonferroni correction, as we analyze the means of different items (i.e., data) between two groups. The one-sided test was applied, as we have expected higher usability ratings for all items (Q1-9) in the matched-condition due to the law of attraction. Additionally, a Chi-square test was used to examine whether the matched bots were preferred and whether an adaption of the matched bot could be perceived when both are directly compared.

	matched		mismatched	
Usability Item	**mean**	**SD**	**mean**	**SD**
Q1 Overall Satisfaction*	**3.94**	1.06	3.58	1.19
Q2 Naturalness*	**3.68**	1.07	3.45	1.12
Q3 Promoter Score*	**3.56**	1.12	3.20	1.19
Q4 Dialogue Length	3.50	0.10	**3.53**	1.01
Q7 Trustworthiness*	**3.52**	0.95	3.38	0.94
Q9 Appropriateness*	**3.74**	1.11	3.24	1.24

Table 5: Descriptive statistics (N=266) for Q1-4, Q7, and Q9 comparing matched with mismatched personality. * denotes a statistically significant difference of means ($p < 0.05$).

Shown in Table 5, there is a significant difference between the overall satisfaction (Q1) of the matched personality is significantly higher compared to the mismatched personality, t(265) = 4.016, p = < .001, d = .246. Moreover, the perceived naturalness (Q2) of the matched chatbots is significantly higher compared to the mismatched personality ones, t(265) = 2.782, p = .003, d = .171. Similarly, the matched personality type chatbot is more likely to be recommended to a friend (Q3)

[5] www.crowdee.com

compared to the mismatched personality, t(265) = 3.894, p = < .001, d = -.239. Furthermore, there is a significantly higher trustworthiness (Q7) in the matched personality than the mismatched one, t(265) = 2,015, p = .022, d = .124. Finally, also the matched personality scores significantly higher in appropriateness for the task at hand than the mismatched personality, t(265) = 4.572, p = <.001, d = .280.

These results support our assumption that a matched personality has a positive influence on the perceived usability of our job recommender chatbot. However, it seems that there is only a small effect of the matched personality adaption.

Despite explicit manipulation, results also show that no significant difference was perceived by the participants with respect to the dialogue length, t(265) = -0.373, p = .355.

5.3 Validation of Design Choices

Table 6 shows the results of our analysis on the impact of the design choices.

The one-sided t test found that the formality (Q5) of the introverted bot is significantly higher compared to the extroverted bot, t(265) = 24.571, p = < .001, d = 1.507. This strongly supports the assumption that the introverted bot is perceived as more formal than the extroverted, which corresponds to the design choices.

Moreover, the perceived trustworthiness of the introverted bot is significantly higher compared to the extroverted bot, t(265) = 6.840, p = <.001, d = .419, while there is also a significantly higher appropriateness of the introverted bot compared to the extroverted bot, t(265) = 9.190, p = <.001, d = -.563.

Message length (Q8) and Emotionality (Q6) were not perceived significantly differently, although messages from the introverted bot are perceived as longer compared to the extroverted bot, t(265) = -2.778, p = < .003, d = -.170. Finally, the bot design of Feeling (Q6) was also not perceived as significantly more emotional than the bot designed as Thinking, t(265) = -0.356, p = .361.

Finally, a direct comparison of both bots was examined with a Chi-square test to assess which chatbot was perceived as most adapted to the user. The results show no significant difference between the I/E personality type and a perceived adaption in the chatbot's behavior, $\chi^2(3) = 2.523$, p = .471.

	Introverted		**Extroverted**	
Usability Item	**mean**	**SD**	**mean**	**SD**
Q5_Formality*	**3.94**	0.95	1.97	1.15
Q7_Trustworthiness*	**3.68**	0.83	3.22	0.10
Q8_Message_Length	3.36	1.10	**3.55**	0.96
Q9_Appropriateness*	**3.94**	0.88	3.04	1.31
	Feeling		**Thinking**	
Q6_Emotional	2.73	1.01	3.56	1.06

Table 6: Descriptive statistics (N = 266) for Q5-9 MOS comparing the introverted and extroverted bot. * denotes a statistically significant difference of means (p < 0.05).

6 Discussion

In general, our results and expectations are in line with the law of attraction within a text-based conversational agent (Park et al., 2012) domain such that overall satisfaction, trustworthiness and appropriateness are significantly higher for the matched personality-based chatbot.

Also, the difference between the combination of ET and IF is much smaller compared to a scenario in which the user interacts with the bots EF and IT. For the first scenario, the messages of the bots only differ by 59 words; however, the second scenario offers 87 words in message length through the overall course of the dialogue.

A set of preliminary results may shed some light on the unexpected results. When looking at a one-sided t test within the sub-sample of EF and IT classified participants, the effect of perceived message length is also greater compared to the whole sample (t(144) = 3,863, p < .001, d = -.321). However, it is natural that the ET and IF types are more similar to each other compared to the EF and IT. Surprisingly, the differently designed emotionality of the messages did not yield significant results in terms of distinction. A possible explanation for this could be that the perception of emotionality is biased by the use of emojis, which are perceived as an emotional variable. The difference in the usage of emoticons between IF and ET is in favor of the ET type. Hence, the ET type could be perceived as more emotional given the higher number of emojis, which is also related to feeling. Therefore, similar to the design aspect of message length, the other combinations of IT and EF should show clearer results as EF is designed to be feeling and uses numerous emojis. A paired t test also supports this assumption where the EF type is per-

ceived as significantly more emotional than the IT, t(144) = 1,967, p = .026, d = .163. Hence, there might be an interference with the usage of emojis and the relationship towards feeling that was not designed clearly enough for those participants that were interacting with ET Carla and IF Sophia. Another possible reason for the lack of perceived emotionality, in general, could be that this study designed the T/F dichotomy under the assumption that there is a correlation with Big 5's agreeableness. Due to the lack of research modelling thinking and feeling linguistically, the linguistic cues of agreeableness were used to design T/F. The two traits correlate with each other (0.47) according to a study by McCrae and Costa (1989) (McCrae and Costa Jr, 1989). Nevertheless, they are not equal, which might result in an information loss or false interpretation other than what was intended. Further, the separation of the extroverted and introverted bot is also dependent on whether they were rated as the matched or the mismatched interaction, respectively. Our study shows, the law of attraction has an impact on the perception of the two chatbots. However, a subliminal study showed that there are no major differences when analyzing the scores within the samples of only matched interactions, the samples of only mismatched interaction, and the whole sample.

When investigating the results, regardless of the matched or mismatched personality, the introverted and formal-designed chatbots (introverted Sophia) were rated higher than the more informal ones (extroverted Carla). This also fits into the domain of job recommendation which is usually associated with professionalism where formality is required. The more formal bot also scores better on appropriateness and overall satisfaction.

For the evaluation, 266 people have interacted with it in a realistic scenario, and have rated the interaction by means of MOS. Similar studies either did not provide a direct interaction with the chatbot (Ahmad et al., 2020) (users only rated screenshots) or could only show tendencies with small sample sizes (Smestad, 2018; Ruane et al., 2020). Hence, to the best of our knowledge, this is the first study to show a statistically significant positive effect, though small, of automatically adapted matched personality of a chatbot (N = 266) toward usability, trust, and appropriateness for the task of job recommendation.

In addition, linguistic cues that correlate with certain personality traits were introduced (Pennebaker and King, 1999; Mairesse et al., 2007; Ruane et al., 2020) and the results presented in this paper further contribute to this body of research. They indicate that personality differences embodied in language were significantly perceived in two out of three design choices. These findings further validate that matched personality results in significantly higher usability scores (in all but one of the items used in our study) of a chatbot. Apart from that, trustworthiness and appropriateness (for the task of job recommendation) were also shown to be significantly better when matching the personality type compared to mismatching it. Our results are in line with previous research (Moon and Nass, 1996; Ahmad et al., 2020; Smestad, 2018; Lee and Nass, 2005; Zumstein and Hundertmark, 2017), while at the same time quantitatively demonstrating the effect of the law of attraction for a high number of participants (Park et al., 2012). In contrast to other studies, our study enhances the transparency of the degree of linguistic cues applied by precisely stating the numbers of linguistic cues; related work on chatbots with personality only described their exact measures briefly.

6.1 Future Research

In future work, we aim to examine whether a chatbot that automatically classifies the user's personality could become more accurate over time with a growing body of textual language to result in a personalized user experience. Additionally, it would be interesting to apply natural language generation (NLG) for the matched response generation of the chatbot to achieve even higher usability scores and higher overall flexibility. A similar approach to automatically create utterances that express a certain personality was developed with PERSONAGE (Mairesse and Walker, 2010).

A potential practical future experiment could be the steady recalculation of the user's personality for the saved conversation logs. This would allow a personality classification model to iteratively verify the user's personality traits with increasing text size. Based on the assumption that larger text samples will improve the accuracy of the predicted personality, the usability of the system could also be improved over time while it is in usage. However, storing the users' texts in business contexts to calculate their personality raises ethical as well as legal questions which have to be studied too.

Eventually, a more dedicated work comparing the selected dichotomies from MBTI along their impact on usability to scales and constructs derived from the FFM would be desirable in order to contribute to further personality theory validation.

References

Martin Adam, Michael Wessel, Alexander Benlian, et al. 2020. Ai-based chatbots in customer service and their effects on user compliance. *Electronic Markets*, 9(2):204.

Rangina Ahmad, Dominik Siemon, and Susanne Robra-Bissantz. 2020. Extrabot vs introbot: The influence of linguistic cues on communication satisfaction. In *AMCIS*, pages 0–10. Researchgate.

Virginia Blankenship, Steven M Hnat, Thomas G Hess, and Donald R Brown. 1984. Reciprocal interaction and similarity of personality attributes. *Journal of Social and Personal Relationships*, 1(4):415–432.

Ryan L Boyd and James W Pennebaker. 2017. Language-based personality: a new approach to personality in a digital world. *Current opinion in behavioral sciences*, 18:63–68.

Gregory J Boyle. 1995. Myers-briggs type indicator (mbti): some psychometric limitations. *Australian Psychologist*, 30(1):71–74.

Ana Paula Chaves and Marco Aurelio Gerosa. 2021. How should my chatbot interact? a survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction*, 37(8):729–758.

Jacob Cohen. 1988. Statistical power analysis for the behavioral sciences. hoboken.

Robert Dale. 2016. The return of the chatbots. *Natural Language Engineering*, 22(5):811–817.

John M Digman. 1990. Personality structure: Emergence of the five-factor model. *Annual review of psychology*, 41(1):417–440.

Adrian Furnham. 1990. Faking personality questionnaires: Fabricating different profiles for different purposes. *Current psychology*, 9(1):46–55.

Adrian Furnham. 1996. The big five versus the big four: the relationship between the myers-briggs type indicator (mbti) and neo-pi five factor model of personality. *Personality and individual differences*, 21(2):303–307.

Alastair J Gill and Jon Oberlander. 2002. Taking care of the linguistic features of extraversion. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 24, page 6. eScholarship.

Ulrich Gnewuch, Stefan Morana, Marc TP Adam, and Alexander Maedche. 2018. "the chatbot is typing…"–the role of typing indicators in human-chatbot interaction. In *Proceedings of the 17th Annual Pre-ICIS Workshop on HCI Research in MIS*, pages 0–5. Researchgate.

Dominic A Infante, Andrew S Rancer, and Deanna F Womack. 1997. Building communication theory. *Waveland PressInc.*

Katherine Isbister and Clifford Nass. 2000. Consistency of personality in interactive characters: verbal cues, non-verbal cues, and user characteristics. *International journal of human-computer studies*, 53(2):251–267.

Oliver P John, Alois Angleitner, and Fritz Ostendorf. 1988. The lexical approach to personality: A historical review of trait taxonomic research. *European journal of Personality*, 2(3):171–203.

Eric Jorgenson. 2015. Development of the Open Jungian Type Scales.

Kwan-Min Lee and Clifford Nass. 2005. Social-psychological origins of feelings of presence: Creating social presence with machine-generated voices. *Media Psychology*, 7(1):31–45.

Tze Wei Liew and Su-Mae Tan. 2016. Virtual agents with personality: Adaptation of learner-agent personality in a virtual learning environment. In *2016 Eleventh International Conference on Digital Information Management (ICDIM)*, pages 157–162. IEEE, The International Review of Research in Open and Distributed Learning.

Max M Louwerse, Arthur C Graesser, Shulan Lu, and Heather H Mitchell. 2005. Social cues in animated conversational agents. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 19(6):693–704.

François Mairesse and Marilyn A Walker. 2010. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, 20(3):227–278.

François Mairesse, Marilyn A Walker, Matthias R Mehl, and Roger K Moore. 2007. Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of artificial intelligence research*, 30:457–500.

Mary H. McCaulley and Charles R. Martin. 1995. Career assessment and the myers-briggs type indicator. *Journal of Career Assessment*, 3(2):219–239.

Robert R McCrae and Paul T Costa. 1987. Validation of the five-factor model of personality across instruments and observers. *Journal of personality and social psychology*, 52(1):81.

Robert R McCrae and Paul T Costa Jr. 1989. Reinterpreting the myers-briggs type indicator from the perspective of the five-factor model of personality. *Journal of personality*, 57(1):17–40.

Robert R McCrae and Oliver P John. 1992. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215.

Matthias R Mehl, Samuel D Gosling, and James W Pennebaker. 2006. Personality in its natural habitat: manifestations and implicit folk theories of personality in daily life. *Journal of personality and social psychology*, 90(5):862.

Sebastian Möller, Paula Smeele, Heleen Boland, and Jan Krebber. 2007. Evaluating spoken dialogue systems according to de-facto standards: A case study. *Computer Speech & Language*, 21(1):26–53.

Youngme Moon and Clifford I Nass. 1996. Adaptive agents and personality change: complementarity versus similarity as forms of adaptation. In *Conference companion on Human factors in computing systems*, pages 287–288. Association for Computing Machinery.

Isabel Myers-Briggs, Mary H McCaulley, Naomi L Quenk, and Allen L Hammer. 1998. *MBTI manual: a guide to the development and use of the Myers-Briggs Type Indicator*, volume 3. CPP.

Babak Naderi, Tim Polzehl, André Beyer, Tibor Pilz, and Sebastian Möller. 2014. Crowdee: mobile crowdsourcing micro-task platform for celebrating the diversity of languages. In *Fifteenth Annual Conference of the International Speech Communication Association*, pages 1496–1497. International Speech Communication Association.

Clifford Nass and Kwan Min Lee. 2001. Does computer-synthesized speech manifest personality? experimental tests of recognition, similarity-attraction, and consistency-attraction. *Journal of experimental psychology: applied*, 7(3):171.

Ketakee Nimavat and Tushar Champaneria. 2017. Chatbots: An overview. types, architecture, tools and future possibilities. *International Journal for Scientific Research & Development*, 5(7):1019–1024.

Eunil Park, Dallae Jin, and Angel P del Pobil. 2012. The law of attraction in human-robot interaction. *International Journal of Advanced Robotic Systems*, 9(2):35.

James W Pennebaker and Laura A King. 1999. Linguistic styles: language use as an individual difference. *Journal of personality and social psychology*, 77(6):1296.

David J. Pittenger. 1993. The utility of the myers-briggs type indicator. *Review of Educational Research*, 63:467 – 488.

ITUT Rec. 2003. P. 851, 2003. subjective quality evaluation of telephone services based on spoken dialogue systems. *International Telecommunication Union, Geneva*.

Elayne Ruane, Sinead Farrell, and Anthony Ventresque. 2020. User perception of text-based chatbot personality. In *International Workshop on Chatbot Research and Design*, pages 32–47. Springer, Cham.

Tommy Sandbank, Michal Shmueli-Scheuer, Jonathan Herzig, David Konopnicki, John Richards, and David Piorkowski. 2017. Detecting egregious conversations between customers and virtual agents. *arXiv preprint arXiv:1711.05780*, pages 1–9.

Klaus Rainer Scherer. 1979. *Personality markers in speech*. Cambridge University Press.

Ryan M Schuetzler, Mark Grimes, Justin Scott Giboney, and Joesph Buckman. 2014. Facilitating natural conversational agent interactions: lessons from a deception experiment. *Information Systems and Quantitative Analysis Faculty Proceedings and Presentations*, pages 0–16.

Tuva Lunde Smestad. 2018. Personality matters! improving the user experience of chatbot interfaces-personality provides a stable pattern to guide the design and behaviour of conversational agents. Master's thesis, NTNU.

Benedict Tay, Younbo Jung, and Taezoon Park. 2014. When stereotypes meet robots: the double-edge sword of robot gender and personality in human–robot interaction. *Computers in Human Behavior*, 38:75–84.

Diana-Cezara Toader, Grațiela Boca, Rita Toader, Mara Măcelaru, Cezar Toader, Diana Ighian, and Adrian T Rădulescu. 2020. The effect of social presence and chatbot errors on trust. *Sustainability*, 12(1):256.

Jess und Gregory J Feist. 2002. *Theories of Personality*. 7th edition. McGraw-Hill Humanities/Social Sciences/Languages.

Darius Zumstein and Sophie Hundertmark. 2017. Chatbots–an interactive technology for personalized communication, transactions and services. *IADIS International Journal on WWW/Internet*, 15(1):96–109.

Appendix A: Item Setup for Overall Study

No.	Type	Item
PQ1	IE	I consider myself to be energetic rather than relaxed.
PQ2	IE	I would describe myself as a talker rather than a listener.
PQ3	IE	I oftentimes like to stay home rather than going out to town.
PQ4	IE	Speaking in public is more likely to frighten me than to entertain me.
PQ5	IE	I describe myself as a calm person rather than being impulsive.
PQ6	IE	I would describe myself as an open person instead of being guarded.
PQ7	TF	I am more a skeptical person than a believer.
PQ8	TF	I rather strive to have an mechanical mind than striving to let my thoughts run free.
PQ9	TF	I am easily hurt and not emotionally thick-skinned.
PQ10	TF	I prefer to follow my heart rather than my head.
PQ11	TF	I rather value emotions instead of feeling uncomfortable with (expressing) them.
PQ12	TF	I rather use reason over instinct.
Q1	ITU-T	Overall, I was satisfied with the chatbot.
Q2	ITU-T	The chatbot reacted naturally.
Q3	ITU-T	I would advise my friends to also use the chatbot.
Q4	ITU-T	The overall dialogue course was too long.
Q5	Custom	The chatbot was formal.
Q6	Custom	The chatbot was emotional.
Q7	Custom	The chatbot was trustworthy.
Q8	Custom	The messages were too long.
Q9	Custom	The chatbot was appropriate according to my expectations.
C1	Comparison	Do you believe the interaction was adapted to you personally?
C2	Comparison	Which chatbot do you like more?
G1	General	How often do you use chatbots?
G2	General	Please tell us about your age range.
G3	General	Is English your native language?
G4	General	Please tell us about your gender.
G5	General	What is your current profession?

Table 7: Overview of all items used throughout the study.

Towards Socially Intelligent Agents with Mental State Transition and Human Value

Liang Qiu[*1], **Yizhou Zhao**[*1], **Yuan Liang**[2], **Pan Lu**[1], **Weiyan Shi**[3], **Zhou Yu**[3], **Song-Chun Zhu**[1]
[1]UCLA Center for Vision, Cognition, Learning, and Autonomy
[2]University of California, Los Angeles
[3]Columbia University
liangqiu@ucla.edu

Abstract

Building a socially intelligent agent involves many challenges. One of which is to track the agent's mental state transition and teach the agent to make decisions guided by its value like a human. Towards this end, we propose to incorporate mental state simulation and value modeling into dialogue agents. First, we build a hybrid mental state parser that extracts information from both the dialogue and event observations and maintains a graphical representation of the agent's mind; Meanwhile, the transformer-based value model learns human preferences from the human value dataset, VALUENET. Empirical results show that the proposed model attains state-of-the-art performance on the dialogue/action/emotion prediction task in the fantasy text-adventure game dataset, LIGHT. We also show example cases to demonstrate: (*i*) how the proposed mental state parser can assist the agent's decision by grounding on the context like locations and objects, and (*ii*) how the value model can help the agent make decisions based on its personal priorities.

1 Introduction

Recently, there has been remarkable progress in language modeling with large-scale pretrained models (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2019). Such models are used to build either general chatbots (Zhang et al., 2020) or task-oriented dialogue systems (Peng et al., 2020; Acharya et al., 2021; Qiu et al., 2020). While most of these systems have been able to generate fluent sentences, there are two major challenges towards building socially intelligent agents. First, considering dialogues as a "meeting of minds" (Gardenfors, 2014) or achieving some alignment of the interlocutors' mental models (Rumelhart et al., 1986; Stolk et al., 2016), few existing works are explicitly tracking the mental state transition of agents (Adhikari et al., 2020). Endowing current dialogue systems with such capability would allow the agent to condition its utterance on the context, simulate the effect of its actions, and further help understand the extended meaning, implicature, and irony expressed by the user (Grice, 1981, 1989). Second, it remains under-explored to teach agents to make a rational decision guided by its value. From a social and cultural perspective, humans tend to have a common preference described by the utility function related to individual values, common sense, and social awareness. For the example in Figure 1, someone who values personal security prefers staying at home rather than going outside at night.

Figure 1: Socially intelligent agents with mental state simulation and human values.

Our work aims to alleviate the aforementioned problems, based on Embodied Cognitive Linguistics (ECL) (Lakoff and Johnson, 1980; Gardenfors, 2014) and established value theories in sociology (Schwartz, 2012). The ECL states that natural language is inherently executable, driven by mental simulation and metaphoric inference (Lakoff and Johnson, 1980), and learned through embodied interaction (Feldman and Narayanan, 2004; Tamari et al., 2020). Following its tenents, we present a hybrid mental state parser that converts dialogue and

[*]Equal contribution. The work was done prior to Liang joining Amazon Alexa.

Proceedings of the SIGdial 2022 Conference, pages 146–158
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

event observations into a graphical representation of the agents' mind. Initialized with the location and object description, the interpretable representation is updated through the interaction history to track the evolving process of an agent's belief about surroundings and other agents.

In the field of intercultural research, Schwartz (1992); Schwartz et al. (2012) identify basic individual values that are recognized across cultures. Inspired by the theory, we propose to incorporate a value model that learns social common preferences from the human value knowledge base, VALUENET (Qiu et al., 2022). We perform experiments on a large-scale text-based embodied environment LIGHT (Urbanek et al., 2019). Empirical results show that the model with our mental state emulator and value function achieves the highest performance that aligns with human annotation among existing transformer-based models. Moreover, case studies further demonstrate that the mental state provides extra context information, while the value model helps agents make value-driven decisions.

Our contributions are two-fold. First, we propose to rethink the design of current dialogue systems and suggest a new paradigm from the perspective of cognitive science and contemporary sociology. Second, we present a new framework for building socially intelligent agents by incorporating mental state simulation and human value modeling into dialogue generation and decision making. Our methodology can be generalized to a wide range of interactive social situations in dialogue systems (Zhao, 2019), virtual reality (Lai et al., 2019), and human-robot interactions (Yuan and Li, 2017).

2 Related Work

2.1 Text-based Embodied AI

Most recent works in dialogues only study the statistical regularities of language data, without an explicit understanding of the underlying world. Virtual embodiment (Krishnaswamy and Pustejovsky, 2019) was proposed as a strategy for language research by several previous works (Brooks, 1991; Kiela et al., 2016; Gauthier and Mordatch, 2016; Mikolov et al., 2016; Lake et al., 2017). It implies that the best way to acquire human knowledge is to have the agent learn through experience in a situated environment. Urbanek et al. (2019) introduce LIGHT as a research platform for studying grounded dialogue (Grice, 1981, 1989; Stalnaker, 2002), where agents can perceive, emote, and act when conducting dialogues with other agents. Ammanabrolu et al. (2020) extend LIGHT with a dataset of "quests", aiming to create agents that both act and communicate with other agents in pursuit of a goal. Instead of guiding the agent to complete an in-game goal, our work aims to teach agents to speak/act in a socially intelligent way. Besides LIGHT, there are also other text-adventure game frameworks, such as Narasimhan et al. (2015) and TextWorld (Côté et al., 2018), but no human dialogues are incorporated in them. Based on the TextWorld, there are recent works (Yuan et al., 2018; Yin and May, 2019; Adolphs and Hofmann, 2019; Adhikari et al., 2020) on building agents trained with reinforcement learning.

2.2 Mental State Transition

An important hypothesis in the ECL (Lakoff and Johnson, 1980; Feldman and Narayanan, 2004) is that humans understand the meaning of language by mentally simulating its content. Great efforts have been made to model human mental states. For example, Dinan et al. (2019) design a memory network capable of storing knowledge and generating natural responses conditioning on retrieved entries. Adhikari et al. (2020) propose a graph-aided transformer agent (GATA) that infers and updates latent belief graphs during planning to enable effective action selection. However, GATA is designed for capturing game dynamics not dialogues, and our method is more flexible to encode both explicit environmental changes caused by agents' actions and implicit mental state updates triggered by agents' utterances. Such hybrid approaches mixing fixed symbolic states with deep continuous states are studied in recent neural-symbolic research (Sun, 1994; Garcez et al., 2008; Besold et al., 2017; Yi et al., 2018). The result interpretable graphs have two benefits: *(i)* the mental state parsing could be viewed as a form of executable semantic parses (Liang, 2016), so it is easy to write programs to simulate the mind transition. A real-world application leveraging similar approaches is seen in Andreas et al. (2020). *(ii)* the unified graphical representation can be extended to model higher-order mental states, *i.e.,* theory-of-mind (ToM) (Premack and Woodruff, 1978). ToM is defined as the ability to impute mental states to oneself and others. It enables humans to make inferences about what other people believe in a

given situation and predict what they will do (Apperly, 2010; Gordon and Hobbs, 2017; Akula et al., 2019). ToM is thus impossible without the capacity to form "second-order representations" (Dennett, 1978; Pylyshyn, 1978; Ganaie and Mudasir, 2015).

2.3 Human Value

When teaching agents to speak and act in a socially intelligent way, an approach considering values should be adopted. The theory of basic human values, developed by Schwartz (1992, 2012), tries to measure universal values that are recognized throughout major cultures. A set of 10 basic values[1] are identified and serve as the guiding principles in the life of a person or group (Cieciuch and Davidov, 2012), as shown in Figure 2. Simi-

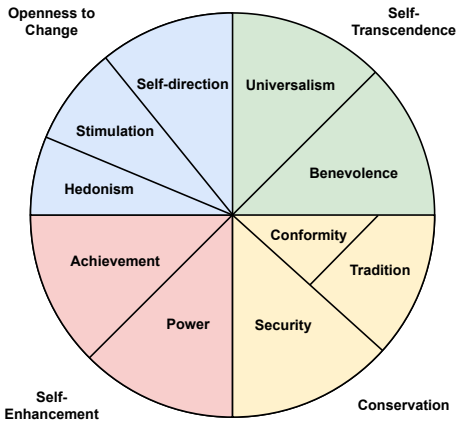


Figure 2: Theory of Basic Human Values (Schwartz, 1992).

larly, in economics and ethics, the concept of utility was developed as a measure of pleasure or satisfaction that drives human activities at all levels. Derived from the rational choice theory (Abella, 2009), utilitarianism states that human decision-making could be viewed as a two-step procedure. First, we select a feasible region based on financial, legal, physical, or emotional restrictions we are facing. Then we make a choice based on the preference order (Allingham, 2002; de Jonge, 2012). In this paper, we learn a transformer-based utility function of human values from the knowledge base VALUENET (Qiu et al., 2022). Inspired by descriptive ethics, VALUENET provides social scenarios and annotated human preference to teach the agent human attitudes to various ethical situations. The dataset is curated from the widely used social commonsense dataset SOCIAL-CHEM-101 (Forbes et al., 2020) and labeled with Amazon Mechanical Turk.

[1] A refinement of the theory (Schwartz et al., 2012), partitions the same continuum into 19 more narrowly defined values that permit more precise explanation and prediction.

3 Problem Formulation

We will first briefly introduce the text-adventure environment LIGHT, followed by the mental state modeling and value utility formulation.

LIGHT (Urbanek et al., 2019) is a large-scale crowd-sourced fantasy text-adventure platform for studying grounded dialogues. Figure 4ⓐ shows a typical local environment setting, including location description, objects (and their affordances), characters, and their personas. Agents can talk to other agents in free-form text, take actions defined by templates, or express certain emotions (Figure 4ⓑ). Given the environmental setting and observation history, our task is to predict the agent's utterance/action/emotion for the next turn. To achieve this goal in a socially intelligent manner, we model the agent's mental state transition and incorporate human values. The mind model is proposed to depict the agent's belief about the underlying states of the text world. Meanwhile, a utility function of human values is designed to describe human preferences in common social situations. We experiment on the text-adventure game for simplicity, but the proposed architecture supports richer environments.

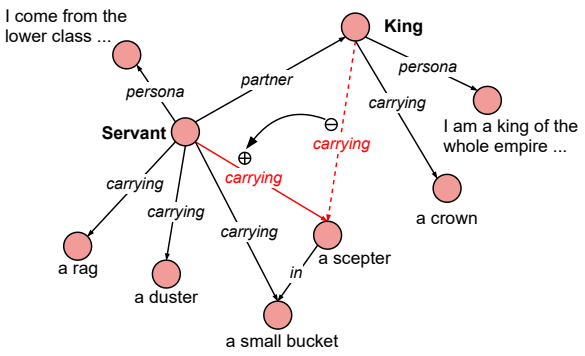


Figure 3: A graphical representation of the agent's mental state. Nodes are attributed with encoded natural language description of agents, objects and the environment. Agents' action trigger explicit topology changes of the graph.

3.1 Mental State Modeling

Our goal is to parse, construct and maintain the mental states in dialogues. With the mental state grounding on the details of the local environment, the agent could simulate and reason the evolution-

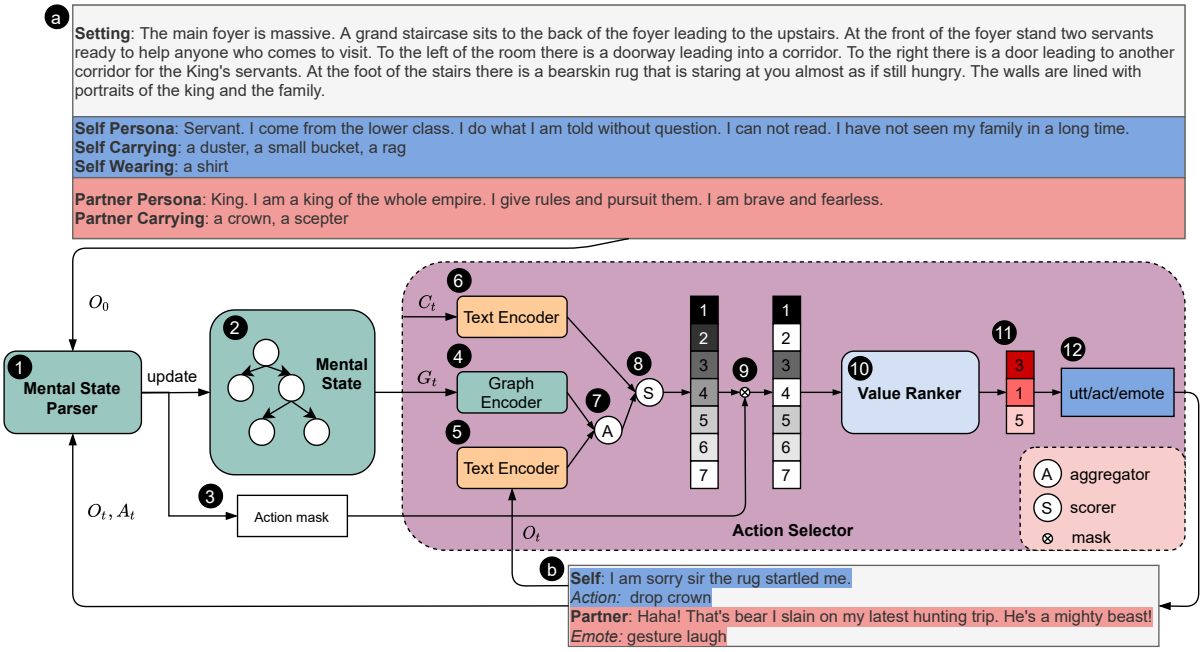


Figure 4: Socially Intelligent Agent Architecture with Mental State Parser and Value Model.

ary status of the world and condition its speaking and actions. A graphical representation of the mental state is proposed, as illustrated in Figure 3. Nodes in the graph represent the involved agents, persona descriptions, objects, objects' descriptions, and setting descriptions, which will change as the game setting switches. The relational edges between these nodes describe the state of mind. The mental state is updated with the observed dialogue history or actions, *e.g.*, *King gives the scepter to the servant* will result in the scepter being moved from the king to the servant.

3.2 Human Value Modeling

We assume that the agent in the fantasy world would make near-optimal choices to maximize the utility of its preferred values. We denote the available alternatives to be a set of n exhaustive and exclusive utterances or actions $A = \{a_1, ..., a_i, ..., a_n\}$. The value function $f_v(\cdot)$ describes the utility score of the alternative from the value dimension $v, v \in V = \{$*achievement, power, security, conformity, tradition, benevolence, universalism, self-direction, stimulation, hedonism*$\}$[2]. For example, if a_i is more preferred than a_j in terms of *security*, then $f_{security}(a_i) > f_{security}(a_j)$. Usually, we cannot find an analytical form of the value function. However, what matters for preference ordering is which of the two options gives the higher expected utility, not the numerical values of those expected utilities.

In LIGHT, the agent's value priority is reflected by its persona description. For the example in Figure 4ⓐ, the servant is a person who values *conformity* and *tradition* and has a lower priority on *self-direction* and *stimulation*. Using the same value function to approximate a value priority parser: $f_v(p)$, where p is the persona description, the utility or the desirability of candidate a_i to person p is the Euclidean distance between its value priority and the candidate's utility score:

$$u(a_i) = \sqrt{\sum_{v \in V} \left(f_v(p) - f_v(a_i)\right)^2}. \quad (1)$$

Since some actions could be impossible physically (*e.g., one cannot drop an object if the agent is not carrying the object*), the decision making process becomes a problem of maximizing the utility score that is subject to some constraints from the mental state, *i.e.*, $u(a|c)$, where c represents the context or constraints.

4 Algorithms

The overall architecture of our proposed framework is illustrated in Figure 4. For each scenario, a setting description (Figure 4ⓐ) is provided by the LIGHT environment, which can include a description of the location, object affordances, agents' personas, and the objects that agents are carrying,

[2]Detailed definition for each dimension is attached in Appendix A.1.

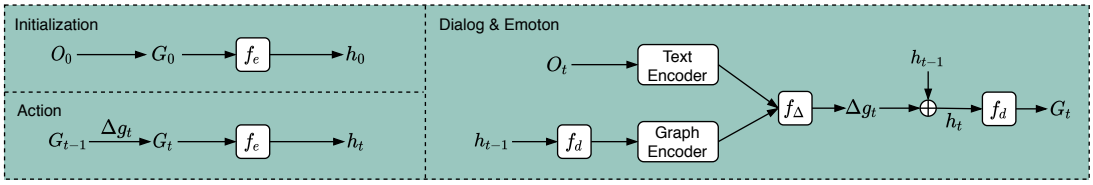


Figure 5: Overall Architecture of the Hybrid Mental State Parser

wearing, or wielding. The free-form conversations, actions, and emotions are logged during the communication as the observation history (Figure 4ⓑ). To begin with, a mental state parser will parse the setting descriptions into graph representation and initialize the agent's mental state (steps ① and ②). Besides the mental state updating, the parser also outputs an action mask that is aimed to rule out actions that are physically or causally impossible to take (step ③). A graph encoder (step ④) and a text encoder (step ⑤) will convert the mental state graph G_t and the dialogue observation O_t into vector representations, respectively. The same text encoder will be used to encode the candidates C_t (step ⑥). In step ⑦, the context vectors are combined by a bi-directional attention aggregator (Yu et al., 2018; Seo et al., 2016), and each candidate is assigned a score with a Multi-Layer Perceptron (MLP) (step ⑧). The action mask is then applied to get the feasible candidates under current mental state constraints (step ⑨). In steps ⑩ and ⑪, the top three candidates from the last step will be fed into the value model and re-ranked. Finally, the selected utterance/action/emotion is executed by the agent (step ⑫) and fed back to the environment. Upon receiving the response from other agents in the environment, the new observation will be again parsed and used to update the agent's state of mind, and the cycle repeats. In the following, we will describe each component in more detail.

4.1 Mental State Modeling (steps ①-②)

Figure 5 describes the architecture of the mental state parser. We define the mental state graph $G \in [-1, 1]^{R \times N \times N}$, where R is the maximum number of relation types and N is the maximum number of entities. The initial mental state graph G_0 is constructed by a ruled-based parser from the setting description O_0. The graph is encoded by function f_e to a hidden state h_0 that is later used for graph update. At game step t, the mental state parser parses relevant information from observation O_t and update the agent's mental state from G_{t-1} to G_t. Considering that observation O_t typically conveys incremental information from step $t-1$ to t, we generate the graph update Δg_t instead of the whole graph at each step

$$G_t = G_{t-1} \oplus \Delta g_t, \tag{2}$$

where $\oplus$ is the graph update operation. The graph update can be either discrete or continuous, and there have been studies on the pros and cons of each updating method (Adhikari et al., 2020). The discrete approach may suffer from an accumulation of errors but benefit from its interpretability. The continuous graph model needs to be trained from data, but it is more robust to possible errors. In this work, we propose a hybrid (discrete-continuous) method for updating the agent's state of mind by considering there exists a mixture of discrete events and continuous information in typical human-machine interactive environments. In the specific example of our tested LIGHT, the actions or events are template-based, it is more appropriate to adopt a discrete method for parsing; meanwhile, since utterances are challenging to be encoded into discrete representations, we apply a continuous update method instead.

4.1.1 Discrete Graph Definition & Update

To update the graph, we define Δg_t as a sequence of update operations of the following two atomic types:

- ADD(`src`, `dst`, `relation`): add a directed edge, named `relation`, from node `src` to node `dst`.
- DEL(`src`, `dst`, `relation`): delete a directed edge, named `relation`, from node `src` to node `dst`.

LIGHT defines various actions including *get*, *drop*, *put*, *give*, *steal*, *wear*, *remove*, *eat*, *drink*, *hug* and *hit*, and each taking either one or two arguments, *e.g.*, *give scepter to servant*. Every action could be parsed as one or a sequence of update operators that act on G_{t-1}. For example, actor performing "*give object to agent*" can be parsed into DEL(*actor*,

object, *carrying*) and ADD(*agent*, *object*, *carrying*). The rule-based parsing of the setting description and the discrete events could also be replaced by a seq2seq decoding process. Since both strings are well-structured in LIGHT, we omit training such a decoder for simplicity. Note that actions in LIGHT could only be executed when constraints are met, so we also generate an action mask according to the current mental state. By checking the adjacency matrix, we rule out action candidates conducted on objects that are inaccessible.

4.1.2 Continuous Graph Definition & Update

Besides the actions taken by the agents, their utterances could also have an implicit impact on the agents' mental states. To handle the continuous dialogue observation, we use a recurrent neural network as the graph update operation $\oplus$.

$$\begin{aligned} \Delta g_t &= f_\Delta(h_{G_{t-1}}, h_{O_t}), \\ h_t &= \text{RNN}(\Delta g_t, h_{t-1}), \\ G_t &= \text{MLP}(h_t). \end{aligned} \tag{3}$$

The function f_Δ aggregates the information from the previous mental state G_{t-1} and observation O_t to generate the graph update Δg_t. $h_{G_{t-1}}$ denotes the representation of G_{t-1} from the graph encoder. h_{O_t} is the output of the text encoder. h_t is a hidden state acting as the memory, from which we decode the new mental state G_t using a MLP. For the recurrent operator, we could either use LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014). More details on the graph encoder and text encoder we applied are presented in the section 4.2.

4.2 Action Selector (steps ④-⑪)

Conditioned on the agent's mental state, the action selector chooses the optimal candidate based on the prediction task (*i.e.*, utterance, action, or emotion). The selector consists of five components: a graph encoder (Fig. 4④) to convert the state-of-mind graph to a hidden state vector; a text encoder (Fig. 4(⑤, ⑥)) to encode the dialogue history and text candidates; an aggregator (Fig. 4⑦) to fuse the two context representations; a general scorer (Fig. 4⑧) to assign a score to each candidate; and a value model (Fig. 4⑩) to re-rank the candidates based on the assigned persona.

1. Graph Encoder. We use relational graph convolutional networks (R-GCNs) (Schlichtkrull et al., 2018) to encode the graph representation of mental states. The R-GCN is adapted from Graph Convolutional Networks (GCNs) so that it could embed the edge attributes (relational text embedding) in the mental state graph.

2. Text Encoder. A BERT-based (Devlin et al., 2019) encoder converts the text-based dialogue history into a vector representation, using the last hidden state corresponding to the `[CLS]` token; We also use the same encoder to encode the text response candidates.

3. Aggregator. A bi-directional attention layer (Yu et al., 2018; Seo et al., 2016) is adopted to fuse the information from the mental state and the contextualized text hidden state. The co-attention allows the agent to focus on the memory part that has been mentioned in the dialogue.

4. Scorer. The full context representation vector is concatenated with each candidate and an MLP layer with softmax activation generates a score for each of them.

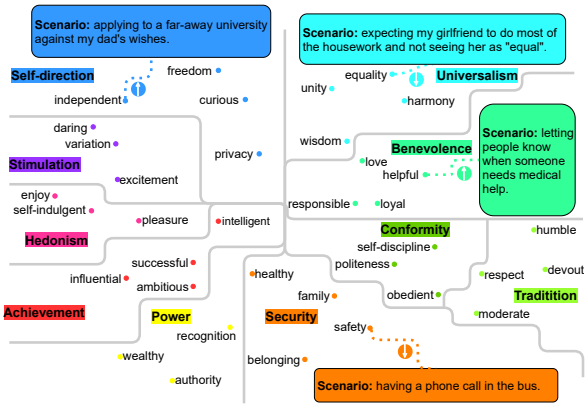


Figure 6: The VALUENET (Qiu et al., 2022) dataset with social scenarios organized by Schwartz values (Schwartz, 2012).

5. Value Ranker. After all the candidates are ranked, we select the top three candidates and then re-rank them according to the proposed value model. The value model is a BERT-based utility scorer trained on the knowledge base VALUENET (Qiu et al., 2022). A custom input format constructed as '`[CLS][$VALUE]s`' is fed into the BERT, *i.e.,*

$$f_v(\texttt{s}) = \text{BERT}(\texttt{[CLS][\$VALUE]s}), \tag{4}$$

where `[CLS]` is the special token for regression, `s` is the scenario, and `[$VALUE]` are special tokens we define to prompt (Li and Liang, 2021; Brown et al., 2020) the transformer the interested value dimension v. A regression head is put on top of the model to get a continuous estimation of the utility in the range of $[-1, 1]$.

Method	*Seen Test* Dialogue R@1/20	Action Acc	Emotion Acc	*Unseen Test* Dialogue R@1/20	Action Acc	Emotion Acc
BERT-based Bi-Ranker	76.5	42.5	25.0	70.5	38.8	25.7
BERT-based Cross-Ranker	74.9	50.7	25.8	69.7	51.8	28.6
discrete mental state	75.8	52.1	25.1	69.9	53.4	25.5
continuous mental state	77.3	49.3	**26.2**	72.1	45.2	29.1
hybrid mental state	78.4	53.5	26.1	72.3	54.3	29.5
hybrid+mask	78.5	54.5	26.1	72.3	55.4	29.4
hybrid+mask+value	**78.8**	**56.4**	26.1	**72.6**	**57.5**	**30.1**
Human Performance*	87.5	62.0	27.0	91.8	71.9	34.4

Table 1: Model performance on the LIGHT *Seen Test* and *Unseen Test*. For dialogue prediction, Recall@1/20 is reported for ranking the ground truth among 19 other randomly chosen candidates. Percentage accuracy is calculated for action and emotion prediction. (*) Human performance is reported by the original paper (Urbanek et al., 2019) on a subset of data.

The VALUENET is organized in 10 dimensions of Schwartz values, as shown in Figure 6. It consists of social scenarios curated from SOCIAL-CHEM-101 (Forbes et al., 2020). And the samples are annotated by Amazon Mechanical Turk workers, who are asked about their attitudes towards provided scenarios. For example, if you are someone who values *benevolence*, will you do or say: "today I buried and mourned a rat"? Their choices (yes, no, unrelated) are then quantified to numerical utilities: +1, -1, 0, respectively.

5 Experiments

We conduct experiments on the LIGHT dataset and compare our model with state-of-the-art methods based on two variants of BERT models. An ablation study is carried out to justify our model design, and a case study is performed to demonstrate how the proposed framework could help the agent ground upon the environment details and make value-driven decisions.

5.1 Experimental Setup and Implementation

The dialogues in LIGHT are split into *train* (8539), *valid* (500), *seen test* (1000), and *unseen test* (739) as the dataset is released. The *unseen test* set consists of dialogues collected on a set of scenarios that have not appeared in the training data. We use the history of dialogues, actions, and emotions to predict the agent's next turn. Note that the original paper manually filters out actions with no affordance leveraging the object annotation, while we provide all candidates to demonstrate our model's capability of reasoning feasible actions automatically from the agent's mental state.

Here we describe the implementation details of the proposed framework. The mental state graph is initialized with a structured setting string including all involved elements in the scenario (an example is attached in Appendix A.2). The setting parser is based on general parsing tools: regular expression and spaCy (Honnibal and Montani, 2017; Clark and Manning, 2016; Honnibal and Johnson, 2015), resulting in the initial mental state graph as shown in Figure 7. For the functions f_e and f_d, we

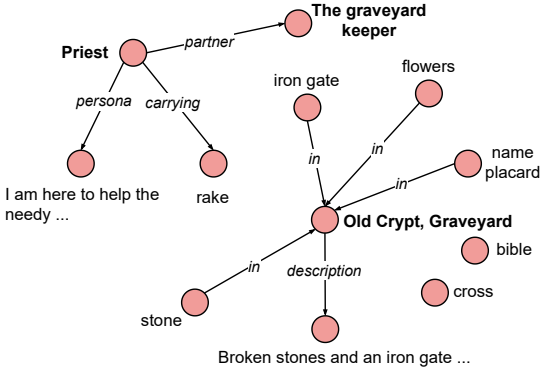


Figure 7: Initial mental state graph parsed from the example setting string in Appendix A.1. The nodes of objects' descriptions are omitted to save space.

use two-layer MLPs with tanh (Karlik and Olgac, 2011) and ReLU (Agarap, 2018) activations. The **Text Encoder** is a pretrained BERT (base-uncased) model (Wolf et al., 2020). The **Graph Encoder** is an R-GCN with six layers and a hidden size of 64. We also adopt the highway connections between consecutive layers for faster convergence and 3-basis decomposition to reduce the parameters and prevent overfitting.

5.2 Baseline Models

Two BERT-based models (Urbanek et al., 2019) are used as strong baselines, which have kept the state-of-the-art performance on this task. **BERT Bi-Ranker** produces a vector representation for the context and each candidate. Each candidate is assigned a score by the dot product between the context embedding and the candidate embedding. **BERT Cross-Ranker** concatenates the context string with each candidate and feeds the string to the BERT model instead. Compared with the bi-ranker, The cross-ranker allows the model to attend to the context when encoding each candidate.

5.3 Results and Analysis

Table 1 shows the results, where our model outperforms the state-of-the-art models by a large margin. To understand the results, we first compare mental state graph designs using discrete, continuous, and the proposed hybrid parser.

The discrete mental state parser uses actions to explicitly update the graph to augment the context representation. In the action prediction task, the discrete parser outperforms the purely continuous method (+2.8% (seen), +8.2% (unseen)), the BERT Bi-Ranker (+9.6% (seen), +14.6% (unseen)), and the BERT Cross-Ranker (+1.4% (seen), +1.6% (unseen)). While the continuous mental state parser misses the hard constraints introduced by less frequent actions, it updates the graph implicitly with the dialogues and shows a better result than the discrete one on dialogue prediction (+1.5% (seen), +2.2% (unseen)) and emotion prediction (+1.1% (seen), +3.6% (unseen)).

The hybrid mental state parser performs the best among the three according to almost all metrics, mainly because it aggregates the soft update from the dense dialogue and the hard constraints from the sparse actions. We also notice that the emotion prediction in LIGHT is a hard task because it is not strictly constrained by the context. Even humans can only achieve 27.0% (seen) and 34.4% (unseen) accuracy. Nevertheless, our model provides a relatively 1.2% (seen) and 3.1% (unseen) performance boost compared to the best BERT baseline.

Then, with the ablation study of our proposed action mask (hybrid mental state *vs.* hybrid+mask), we prove the effectiveness of it for improving action accuracy by ∼1% in action prediction. Figure 8 demonstrates how the mental state could help agent ground on the context. We can see a very

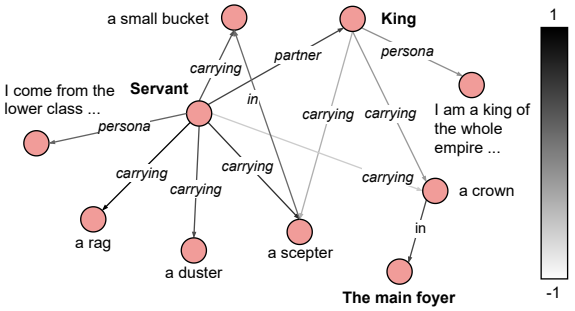


Figure 8: Intermediate mental state for the agent **Servant** in the dialogue example of Figure 4. The adjacency matrix of the mental state graph is visualized and the darkness of the edges represent the relation strength. Only critical relation types between nodes are shown for illustration purpose.

weak relation of the type "*carrying*" between the agent servant and the object crown. Thus the servant should not be able to give the crown to others at this time step. Though our model does not rely on annotated action affordances during action predicting, an action mask can be reasoned from such a mental state, which helps filter out physical or causally impossible actions.

Lastly, we analyze the results after introducing the value model. We first compute the value priority of the agent by applying the value function to its persona description. For example, given the servant's persona description in Figure 4, it shows *conformity, tradition*, and *security* have higher utility scores to the agent than other dimensions. Then we calculate utility scores of the top three candidates based on Equation 1. This teaches the agent to make decisions that align with the assigned role and further improves the overall performance, (+0.3% (seen), +0.3% (unseen)) for dialogue prediction, (+1.9% (seen), +2.1% (unseen)) for action prediction, and +0.7% (unseen) for emotion prediction.

6 Conclusion

This paper proposes to build a socially intelligent agent by incorporating mind simulation and human values. We explore using a hybrid parser to track agents' mental state transition. The value model pretrained on VALUENET brings social preference to help the agent make decisions. The model is proved to have a better performance than the state-of-the-art models on LIGHT. In the future, we have a plan to build a dataset to study the implicature in conversation and model deeper levels in the Theory of Mind based on the proposed representation.

References

Alex Abella. 2009. *Soldiers of reason: The RAND corporation and the rise of the American empire*. Houghton Mifflin Harcourt.

Anish Acharya, Suranjit Adhikari, Sanchit Agarwal, Vincent Auvray, Nehal Belgamwar, Arijit Biswas, Shubhra Chandra, Tagyoung Chung, Maryam Fazel-Zarandi, Raefer Gabriel, et al. 2021. Alexa conversations: An extensible data-driven approach for building task-oriented dialogue systems. *arXiv preprint arXiv:2104.09088*.

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and Will Hamilton. 2020. Learning dynamic belief graphs to generalize on text-based games. *Advances in Neural Information Processing Systems*, 33.

Leonard Adolphs and Thomas Hofmann. 2019. Ledeepchef: Deep reinforcement learning agent for families of text-based games. *arXiv preprint arXiv:1909.01646*.

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Arjun R Akula, Changsong Liu, Sari Saba-Sadiya, Hongjing Lu, Sinisa Todorovic, Joyce Y Chai, and Song-Chun Zhu. 2019. X-tom: Explaining with theory-of-mind for gaining justified human trust. *arXiv preprint arXiv:1909.06907*.

Michael Allingham. 2002. *Choice theory: A very short introduction*. OUP Oxford.

Prithviraj Ammanabrolu, Jack Urbanek, Margaret Li, Arthur Szlam, Tim Rocktäschel, and Jason Weston. 2020. How to motivate your dragon: Teaching goal-driven agents to speak and act in fantasy worlds. *arXiv preprint arXiv:2010.00685*.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Ian Apperly. 2010. *Mindreaders: the cognitive basis of" theory of mind"*. Psychology Press.

Tarek R Besold, Artur d'Avila Garcez, Sebastian Bader, Howard Bowman, Pedro Domingos, Pascal Hitzler, Kai-Uwe Kühnberger, Luis C Lamb, Daniel Lowd, Priscila Machado Vieira Lima, et al. 2017. Neural-symbolic learning and reasoning: A survey and interpretation. *arXiv preprint arXiv:1711.03902*.

Rodney A Brooks. 1991. Intelligence without representation. *Artificial intelligence*, 47(1-3):139–159.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Jan Cieciuch and Eldad Davidov. 2012. A comparison of the invariance properties of the pvq-40 and the pvq-21 to measure human values across german and polish samples. In *Survey Research Methods*, volume 6, pages 37–48.

Kevin Clark and Christopher D. Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262, Austin, Texas. Association for Computational Linguistics.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer.

Jan de Jonge. 2012. Rational and moral action. In *Rethinking Rational Choice Theory*, pages 199–206. Springer.

Daniel C Dennett. 1978. Beliefs about beliefs [p&w, sr&b]. *Behavioral and Brain sciences*, 1(4):568–570.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Jerome Feldman and Srinivas Narayanan. 2004. Embodied meaning in a neural theory of language. *Brain and language*, 89(2):385–392.

Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 653–670, Online. Association for Computational Linguistics.

MY Ganaie and Hafiz Mudasir. 2015. A study of social intelligence & academic achievement of college students of district srinagar, j&k, india. *Journal of American Science*, 11(3):23–27.

Artur SD'Avila Garcez, Luis C Lamb, and Dov M Gabbay. 2008. *Neural-symbolic cognitive reasoning*. Springer Science & Business Media.

Peter Gardenfors. 2014. *The geometry of meaning: Semantics based on conceptual spaces*. MIT press.

Jon Gauthier and Igor Mordatch. 2016. A paradigm for situated and goal-driven language learning. *arXiv preprint arXiv:1610.03585*.

Andrew S Gordon and Jerry R Hobbs. 2017. *A formal theory of commonsense psychology: How people think people think*. Cambridge University Press.

H Paul Grice. 1981. Presupposition and conversational implicature. *Radical pragmatics*, 183.

H Paul Grice. 1989. Indicative conditionals. *Studies in the Way of Words*, pages 58–85.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Bekir Karlik and A Vehbi Olgac. 2011. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122.

Douwe Kiela, Luana Bulat, Anita L Vero, and Stephen Clark. 2016. Virtual embodiment: A scalable long-term strategy for artificial intelligence research. *arXiv preprint arXiv:1610.07432*.

Nikhil Krishnaswamy and James Pustejovsky. 2019. Multimodal continuation-style architectures for human-robot interaction. *arXiv preprint arXiv:1909.08161*.

Hsu-Chao Lai, Hong-Han Shuai, De-Nian Yang, Jiun-Long Huang, Wang-Chien Lee, and Philip S Yu. 2019. Social-aware vr configuration recommendation via multi-feedback coupled tensor factorization. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1773–1782.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

George Lakoff and Mark Johnson. 1980. The metaphorical structure of the human conceptual system. *Cognitive science*, 4(2):195–208.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Percy Liang. 2016. Learning executable semantic parsers for natural language understanding. *Communications of the ACM*, 59(9):68–76.

Tomas Mikolov, Armand Joulin, and Marco Baroni. 2016. A roadmap towards machine intelligence. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 29–61. Springer.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pretrained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526.

Zenon W Pylyshyn. 1978. When is attribution of beliefs justified?[p&w]. *Behavioral and brain sciences*, 1(4):592–593.

Liang Qiu, Yizhou Zhao, Jinchao Li, Pan Lu, Baolin Peng, Jianfeng Gao, and Song-Chun Zhu. 2022. Valuenet: A new dataset for human value driven dialogue system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11183–11191.

Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-Chun Zhu. 2020. Structured attention for unsupervised dialogue structure induction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1889–1899, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

David E Rumelhart, Paul Smolensky, James L McClelland, and G Hinton. 1986. Sequential thought processes in pdp models. *Parallel distributed processing: explorations in the microstructures of cognition*, 2:3–57.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Shalom H Schwartz. 1992. Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries. In *Advances in experimental social psychology*, volume 25, pages 1–65. Elsevier.

Shalom H Schwartz. 2012. An overview of the schwartz theory of basic values. *Online readings in Psychology and Culture*, 2(1):2307–0919.

Shalom H Schwartz, Jan Cieciuch, Michele Vecchione, Eldad Davidov, Ronald Fischer, Constanze Beierlein, Alice Ramos, Markku Verkasalo, Jan-Erik Lönnqvist, Kursad Demirutku, et al. 2012. Refining the theory of basic individual values. *Journal of personality and social psychology*, 103(4):663.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Robert Stalnaker. 2002. Common ground. *Linguistics and philosophy*, 25(5/6):701–721.

Arjen Stolk, Lennart Verhagen, and Ivan Toni. 2016. Conceptual alignment: How brains achieve mutual understanding. *Trends in cognitive sciences*, 20(3):180–191.

Ron Sun. 1994. *Integrating rules and connectionism for robust commonsense reasoning*. John Wiley & Sons, Inc.

Ronen Tamari, Chen Shani, Tom Hope, Miriam R L Petruck, Omri Abend, and Dafna Shahaf. 2020. Language (re)modelling: Towards embodied language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6268–6281, Online. Association for Computational Linguistics.

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 673–683, Hong Kong, China. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kexin Yi, Jiajun Wu, Chuang Gan, Antonio Torralba, Pushmeet Kohli, and Joshua B Tenenbaum. 2018. Neural-symbolic vqa: Disentangling reasoning from vision and language understanding. *arXiv preprint arXiv:1810.02338*.

Xusen Yin and Jonathan May. 2019. Comprehensible context-driven text game playing. In *2019 IEEE Conference on Games (CoG)*, pages 1–8. IEEE.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Wang Yuan and Zhijun Li. 2017. Development of a human-friendly robot for socially aware human-robot interaction. In *2017 2nd International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 76–81. IEEE.

Xingdi Yuan, Marc-Alexandre Côté, Alessandro Sordoni, Romain Laroche, Remi Tachet des Combes, Matthew Hausknecht, and Adam Trischler. 2018. Counting to explore and generalize in text-based games. *arXiv preprint arXiv:1806.11525*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Ran Zhao. 2019. *Socially-Aware Dialogue System*. Ph.D. thesis, Carnegie Mellon University.

A Appendix

A.1 Schwartz Value Definition

Self-Direction Defining goal: independent thought and action–choosing, creating, exploring. Self-direction derives from organismic needs for control and mastery and interactional requirements of autonomy and independence. (creativity, freedom, choosing own goals, curious, independent) [self-respect, intelligent, privacy]

Stimulation Defining goal: excitement, novelty, and challenge in life. Stimulation values derive from the organismic need for variety and stimulation in order to maintain an optimal, positive, rather than threatening, level of activation. This need probably relates to the needs underlying self-direction values. (a varied life, an exciting life, daring)

Hedonism Defining goal: pleasure or sensuous gratification for oneself. Hedonism values derive from organismic needs and the pleasure associated with satisfying them. Theorists from many disciplines mention hedonism. (pleasure, enjoying life, self-indulgent)

Achievement Defining goal: personal success through demonstrating competence according to social standards. Competent performance that generates resources is necessary for individuals to survive and for groups and institutions to reach their objectives. As defined here, achievement values emphasize demonstrating competence in terms of prevailing cultural standards, thereby obtaining social approval. (ambitious, successful, capable, influential) [intelligent, self-respect, social recognition]

Power Defining goal: social status and prestige, control or dominance over people and resources. The functioning of social institutions apparently requires some degree of status differentiation. A dominance/submission dimension emerges in most empirical analyses of interpersonal relations both within and across cultures. To justify this fact of social life and to motivate group members to accept it, groups must treat power as a value. Power values may also be transformations of individual needs for dominance and control. Value analysts have mentioned power values as well. (authority, wealth, social power) [preserving my public image, social recognition]

Both power and achievement values focus on social esteem. However, achievement values (e.g., ambitious) emphasize the active demonstration of successful performance in concrete interaction, whereas power values (e.g., authority, wealth) emphasize the attainment or preservation of a dominant position within the more general social system.

Security Defining goal: safety, harmony, and stability of society, of relationships, and of self. Security values derive from basic individual and group requirements. Some security values serve primarily individual interests (e.g., clean), others wider group interests (e.g., national security). Even the latter, however, express, to a significant degree, the goal of security for self or those with whom one identifies. (social order, family security, national security, clean, reciprocation of favors) [healthy, moderate, sense of belonging]

Conformity Defining goal: restraint of actions, inclinations, and impulses likely to upset or harm others and violate social expectations or norms. Conformity values derive from the requirement that individuals inhibit inclinations that might disrupt and undermine smooth interaction and group functioning. As I define them, conformity values emphasize self-restraint in everyday interaction, usually with close others. (obedient, self-discipline, politeness, honoring parents and elders) [loyal, responsible]

Tradition Defining goal: respect, commitment, and acceptance of the customs and ideas that one's culture or religion provides. Groups everywhere develop practices, symbols, ideas, and beliefs that represent their shared experience and fate. These become sanctioned as valued group customs and traditions. They symbolize the group's solidarity, express its unique worth, and contribute to its survival (Durkheim, 1912/1954; Parsons, 1951). They often take the form of religious rites, beliefs, and norms of behavior. (respect for tradition, humble, devout, accepting my portion in life) [moderate, spiritual life]

Tradition and conformity values are especially close motivationally; they share the goal of subordinating the self to socially imposed expectations. They differ primarily in the objects to which one subordinates the self. Conformity entails subordination to persons with whom one frequently interacts—parents, teachers, and bosses. Tradition entails subordination to more abstract objects—religious and cultural customs and ideas. As a corollary, conformity values exhort responsiveness to current, possibly changing expecta-

tions. Tradition values demand responsiveness to immutable expectations from the past.

Benevolence Defining goal: preserving and enhancing the welfare of those with whom one is in frequent personal contact (the 'in-group'). Benevolence values derive from the basic requirement for smooth group functioning and from the organismic need for affiliation. Most critical are relations within the family and other primary groups. Benevolence values emphasize voluntary concern for others' welfare. (helpful, honest, forgiving, responsible, loyal, true friendship, mature love) [sense of belonging, meaning in life, a spiritual life].

Benevolence and conformity values both promote cooperative and supportive social relations. However, benevolence values provide an internalized motivational base for such behavior. In contrast, conformity values promote cooperation in order to avoid negative outcomes for self. Both values may motivate the same helpful act, separately or together.

Universalism Defining goal: understanding, appreciation, tolerance, and protection for the welfare of all people and for nature. This contrasts with the in-group focus of benevolence values. Universalism values derive from survival needs of individuals and groups. But people do not recognize these needs until they encounter others beyond the extended primary group and until they become aware of the scarcity of natural resources. People may then realize that failure to accept others who are different and treat them justly will lead to life-threatening strife. They may also realize that failure to protect the natural environment will lead to the destruction of the resources on which life depends. Universalism combines two subtypes of concern—for the welfare of those in the larger society and world and for nature (broadminded, social justice, equality, world at peace, world of beauty, unity with nature, wisdom, protecting the environment)[inner harmony, a spiritual life]

A.2 Example Environment Setting

An example setting string for the utterance prediction is:
"**_task_speech**
_setting_name Old Crypt, Graveyard
_setting_desc Broken stones and a iron gate closing the entrance with a name placard that the name is worn off.
_partner_name the graveyard keeper who lives across the yard **_self_name** priest
_self_persona I am here to help the needy. I am well respected in the town. I can not accept lying.
_object_desc a gate : The gate is made out of rusty metal. It squeaks as it swings on its hinges.
_object_desc a flowers : you can see them up close but not afar. when noticed, you realize that they are old.
_object_desc a name placard : The placard is made of wood witha clear name on it.
_object_desc a stone : The stone is chipped from being used as target practice from soldier trainees
_object_desc a placard : A sign used to display names of buildings or notices.
_object_desc an iron gate : The gate is ornate, with complicated iron scrollwork patterns.
_object_desc a Rake : This rake is made of carefully split wood with a sturdy looking handle. Seems useful for keeping the leaves under control.
_object_desc a Cross : The cross is broken and with a few dents in the sides.
_object_desc a bible : The bible is bound by black leather, its pages yellowed by years of use.
_object_in_room a gate
_object_in_room a flowers
_object_in_room a name placard
_object_in_room a stone
_object_in_room a placard
_object_in_room an iron gate
_object_carrying a Rake".
The result mental state graph parsed from this setting is illustrated in Figure 7.

Automatic Verbal Depiction of a Brick Assembly for a Robot Instructing Humans

[1,2]Rami Younes, [1]Gérard Bailly, [2]Damien Pellier, [1]Frédéric Elisei
[1] Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France
{firstname.lastname}@gipsa-lab.grenoble-inp.fr
[2]Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, 38000 Grenoble, France
{firstname.lastname}@univ-grenoble-alpes.fr

Abstract

Verbal and nonverbal communication skills are essential for human-robot interaction, in particular when the agents are involved in a shared task. We address the specific situation where the robot is the only agent knowing about both the plan and the goal of the task, and has to instruct the human partners. The case study is a brick assembly. We here describe a multi-layered verbal depictor whose semantic, syntactic, and lexical settings have been collected and evaluated via crowdsourcing. One crowdsourced experiment involves a robot-instructed pick-and-place task. We show that implicitly referring to achieved subgoals (stairs, pillars, etc) increases the performance of human partners.

1 Introduction

Task-oriented interactions between systems and humans, in order to achieve a common goal, are present in many applications. For instance, receiving directions via GPS is a task-oriented communication with the instructions being delivered visually and verbally (Belvin et al., 2001). Similarly, robots have been used to give directions (Bohus et al., 2014), describe its route experience (Rosenthal et al., 2016; Perera et al., 2016) or instruct students in a tutorial class Gomez et al. (2015). In the later work (see Figure 1), a robot helps two participants to perform a jigsaw assembly task using verbal and non-verbal communication.

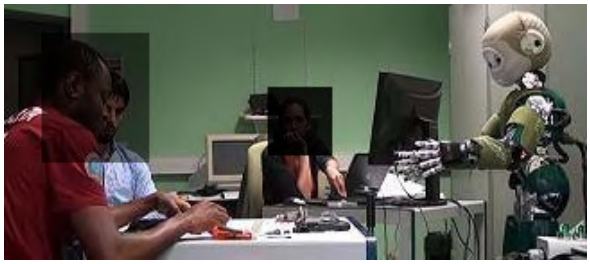


Figure 1: Face-to-face interaction on a Jigsaw reassembly task with an Icub robot (right) acting as the instructor for two students (left). From (Gomez et al., 2015).

The rise of social robots endowed with verbal, co-verbal and non-verbal communication capabilities, now raises the question from the robotic point of view. How a robot and a human can communicate to achieve a common goal and share plans, involving manipulating objects in their common working space?

In this paper, we study this problem by focusing on verbal communication. Indeed, a verbal description of how the task is to be done is a more effective way of communicating objectives than non-verbal descriptions: not only does it improve task performance but also gives rise to more compliance and better mutual adaptation (see Nikolaidis et al., 2017). More precisely, we propose to explore the impact of the verbalization strategy in an extreme case where the robot is the only agent that knows the plan and the goal, and human coworkers are awaiting instructions planned by the robot to achieve the goal. Note that we limit here the number of human partners to one: the opportunistic allocation of tasks between available coworkers will be addressed in a following paper.

When using verbalisation as the main means of communication, an important question is to see how the style, i.e. saying the same thing in different ways, quoted as the *verbalization space* by (Karlgren, 2000), in which the instructions are being delivered, can affect the execution of the task, especially when communicating complex tasks.

In this context, our contribution is threefold:

Styles: we test four different styles on an assembly task (see Figure 2) that offers a large verbalization space, i.e. many stylistic dimensions including choice of geometric relations between bricks, of syntactic and lexical descriptions, etc. One primary style parameter is the use of *context*. By context, we mean implicit referencing to elements of the environment that go beyond the previous and current actions. We compare two AI-generated styles

Proceedings of the SIGdial 2022 Conference, pages 159–171
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

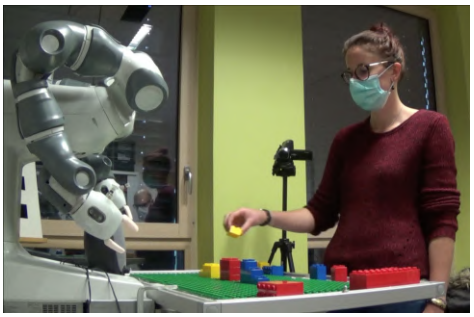


Figure 2: Face-to-face interaction on a LEGO™ assembly task with YUMI acting as the instructor.

(inclusion vs. dismissal of the use of context) with the lowest vs. highest human instructions in terms of: time-to-complete, comprehension/complexity of the instruction, and efficiency/effectiveness of the task completion.

Architecture: we propose a robotic control architecture and its sensorimotor capabilities for task-oriented human interaction. We focus on two key components: (a) the planner and (b) the verbalizer. While the former takes decisions on what to do next, the verbalizer puts each elementary instruction into words for a text-to-speech synthesizer.

Evaluation: we propose an evaluation framework based on a series of three crowdsourced experiments for: (a) collecting and (b) scoring human verbalizations for parameterizing our flexible verbalizer as well as (c) assessing and evaluating the performance of human vs. automatic verbalizations on actual task assembly.

This paper is organized as follows: section 2 contains related work for task-oriented communication and plan description; section 3 describes the overall architecture of our control model, with a closer look on the planner and the verbalizer with its different layers (fig. 6); section (4) introduces the three experiments used to parametrize the verbalizer; section 4.1 presents the results of the first two web-based experiments used for data collection and assessment of human verbalizations; finally, section 4.2 presents the setup and results of the last web-based experiment used to validate the efficiency of the set of rules in our automatic verbalizer and to compare it with human verbalization.

2 Related work

Verbal communication of plans has been used in a large variety of Human-Robot Interaction (HRI) scenarios. They mainly vary along four main categories: (1) task type (e.g. commentating, instructing, navigation). (2) perception capabilities (auditive/visual sensors). (3) style and its use in sentence generation (e.g. information tagging). (4) role (e.g. receptionist, instructor, navigator).

Most HRI tasks require some form of communication. It could be used to describe what happens in the scene: Veloso et al. (2008) presented Rocco, a fully automated RoboCup (Kitano et al., 1997) commentator, aiming at generating real-time summaries of the actions in the games (Voelz et al., 1998). For navigation tasks, Rosenthal et al. (2016); Perera et al. (2016) presented algorithms for generating routing narratives with varying parametrized styles. Belvin et al. (2001) presented a real-time spoken language navigation system able to respond to natural conversational queries. The queries were mainly regarding details of a step in the route. However, responses were generated using simple pre-written "holly sentences" filled with variables extracted from the plan. For assembly tasks, the 'SHRDLU' system (Winograd, 1972, 1974) is quite inspiring: the task focused on manipulating blocks with a robot arm on the basis of the user's textual input. The system translates the user's input into procedures to move the blocks and question the scene. Our work exchanges the roles of the agents: our robot instructs human agents verbally. Fiore et al. (2014) also shares some similarities with our work, i.e. verbalising the actions in the plan for the user as well as explaining which actions should be executed and in what order. However, the task they chose does not require the same level of precision – and their focus was not on verbalization. Finally, the Robert system (Behnke et al., 2020), installed on Bosch equipment, provides its user with a step-by-step instruction (on a screen using text, images, voice and videos) detailing how to complete a given DIY project successfully. Similarly to us, the sequence of instructions (plan) is obtained using HTN planning. They also added a new feature to perceive the scene using connected tools (sensors), enabling the system to check whether the user is performing the project's steps correctly and to provide help in the case of failure.

Zhu et al. (2017) proposed a verbalization system able to generate explanations for navigation as well as grasping and manipulation tasks (pick-and-place kitchen scenario). They used pre-written sentence templates. Canal et al. (2021) proposed PlanVerb, a domain and planner-independent method

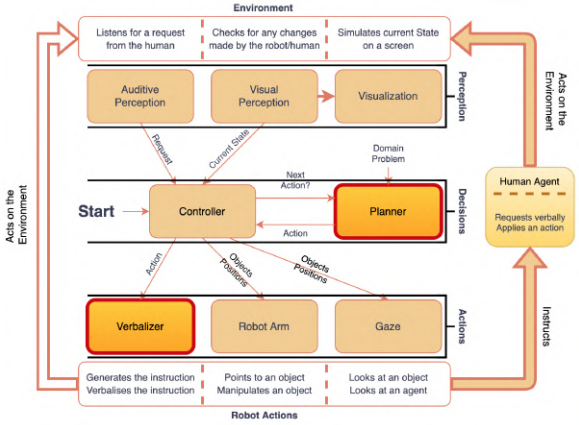


Figure 3: Outer part shows the Human-robot Interaction. The inner part shows the multimodal Architecture. The two components involved in the paper are highlighted .

for the verbalization of task plans based on semantic information tagging of the actions and predicates in the domain (for both PDDL and RDDL). Several works showed the importance of having different styles (Aires et al., 2004; Miehle et al., 2018a; MacFadden et al., 2003). In line with Voelz et al. (1998); Veloso et al. (2008), our verbal generation framework relies on crowdsourcing experiments, for both defining the different styles of the instructions and assessing their efficiency.

Verbal communication has been used in a large variety of situations. Gockley et al. (2005) proposed a robot receptionist with pre-written storylines. Their focus was on long-term interactions with a robot that exhibits personality and character. For their robot bartender, Petrick and Foster (2013) construct plans with tasks, dialogue, and social actions. They advocate for a stronger link between planning and language.

3 The architecture

Figure 3 shows the overall architecture of our HRI system monitoring the interaction between the agents (humans and robot) and the working environment. While no single architecture has proven to be best for all applications, layered architectures have proven to be increasingly popular, due to their flexibility and ability to operate at multiple levels of abstraction simultaneously (Kortenkamp et al., 2016). Similarly to what can be found in (Alami et al., 1998), our robot's architecture can be divided into three levels: perception, decision making and action. With different robots, capabilities change and so do their perception/action modalities. This architecture allows us to add/remove

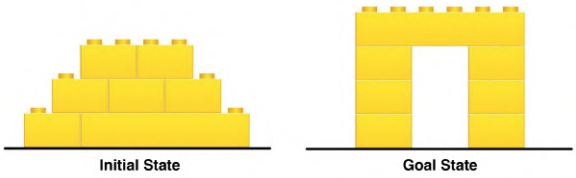


Figure 4: An example where the task is to build a LEGO™ arch – Hierarchical decomposition in fig. 5

modalities, in order to cope with both industrial (e.g. no gaze/head) and humanoid robots (e.g. no grippers).

The perception level is in charge of capturing the current state of the environment as well as the agents acting on it, e.g. analyzing verbal requests coming from the human agent and all changes happening in the working environment. The action level takes charge of all actions towards the environment (e.g robot moving around to pick an object) and agents (e.g. coordinated gaze, speech and pointing to attract partners' attention). The controller is responsible for orchestrating action/perception loops according to the current objective given on request by the planner. In particular, the controller is in charge of monitoring the addressee's activity when processing the robot's instruction, such as on-line attention, task comprehension and correct execution. This includes the chunking or repetition of the instruction if necessary.

It all starts with the controller that receives a "go" signal and requests the first action from the planner. Provided the requested information, the controller (via the action modules) either applies the action or instructs the human agent to apply it. The environment is modified, the controller perceives the updated current state, and the loop continues until the planner deems this task as completed.

The following subsections detail the key modules for the generation of verbal instructions: the planner and the verbalizer.

3.1 Planner

The planner takes as input a domain that contains a logical description of the actions, an initial state obtained from the perception layer and an objective and outputs a sequence of actions (the plan/solution) in order to reach the objective. The initial state and the objective are described as a set of logical propositions.

The first advantage of using a planner is (a) being able to scale to other types of tasks (e.g. assembly,

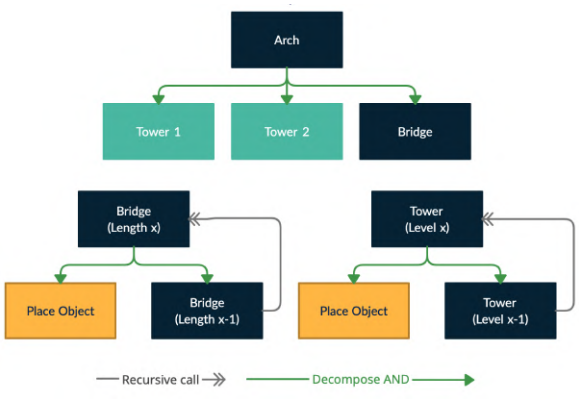


Figure 5: Hierarchical decomposition of an arch into towers/pillars and a bridge/beam – Example in fig. 4

real-world applications, navigation, etc.), by adapting the domain and the problem to the new task; (b) allowing the robot to autonomously adapt and replan when observed actions differ from expected ones. Our planner has two important properties: it performs Hierarchical Planning (Pellier and Fiorino, 2018), i.e. the goal task can be divided into subtasks, and Partial order planning, i.e. some actions can be executed in parallel as long as they satisfy applicability constraints. Hierarchical Planning allows us to specify subtasks, name them, and use these names in the verbalizer. Partial ordering means that the sequence of actions does not need to be fixed and some actions, while satisfying applicability constraints, can be executed in a different order. Partial ordering eases task description and offers more flexibility while executing the plan or verbalizing it. For instance, building an *arch* can be decomposed into building two *pillars* and a *beam*. Each pillar can be constructed by different workers but the beam assembly requires pillars to be finished.

This provides a plan that is almost identical to how a human would plan to build an arch. Thus, enabling the planning system to provide context about the plan as well as some explanation regarding its decisions in the plan. One might argue that context may not be crucial when giving an instruction. However, when dealing with a complex/important task, the addition of hierarchical decomposition into subtasks can help with assigning separate subtasks to different users, or giving a clear explanation to why we are applying a certain action. Our focus for using hierarchy is to give context to help remove ambiguity from instructions, and reduce the number of errors and needed time to complete the task.

The planner takes into account other constraints such as visibility (cannot perform an action if it prohibits you from seeing a later action), applicability (cannot apply what is inapplicable in a given state), and hierarchical constraints (best to finish all actions of a subtask before starting another one). The planner module also provides vital contextual information for the completion of that action.

3.2 Verbalizer

We communicate the instructions via verbalization. The aforementioned verbalizer has multiple parameterized layers (see Figure 6), each shaping one aspect of the message:

The depictor takes charge of all geometric aspects which are vital for completing an action (e.g. 3D position, orientation). This is where business ontologies are hosted (presently, what characterizes a pillar, steps, windows, walls, etc)

The semantic generator focuses on the context, which is in our case giving a hierarchical explanation of where and why we are applying a certain action (e.g. "To finish the red tower").

The syntactic generator focuses on the syntax (i.e. arranging the words and phrases to create well-formed sentences).

The realizer generates the final sentence from the syntactic tree

The text-to-speech system converts the text into audiovisual signals

Style parameters condition each layer so that to be able to adapt communication to the task difficulty and workers' competence, with the objective to improve performance – e.g fewer mistakes and faster completion time of the instruction (Cassell and Bickmore, 2003; Forbes-Riley et al., 2008; Stenchikova and Stent, 2007; Reitter et al., 2006; Mairesse and Walker, 2010; Miehle et al., 2018b) – as well as cognitive load – e.g better recall of the task and processing capacity.

This multi-layer architecture was chosen in order to separate the different skills of the verbalizer, therefore constraining interventions when extending its capabilities. We first discuss each layer separately and spot differences between car navigation vs. assembly task.

3.2.1 Depictor

The depictor handles here agents, objects, and predicates that populate a particular domain: here 3D arrangement of objects. It contains all necessary

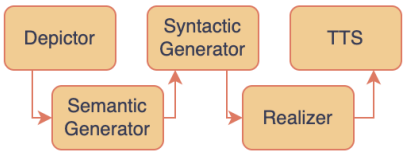


Figure 6: The different layers of the Verbalizer.

properties of these elements such as dimension, relative position, color of objects, sets of objects (e.g. pillars, arches, etc), their relations as well as possible actions (e.g. placing, straddling, sticking, etc). Relative positioning is used by the verbalizer to describe *where to place what, how and why*, e.g. relative to the closest LEGO™ object, or the last placed one. It can be relative to more than one object, or even a structure. For example: it enables the generation of *"To finish the south pillar, put another red brick on top of the previous red brick"*.

It takes as input the action to be performed, the observed scene, as well as the goal, i.e. hierarchical information of the desired final arrangement; and outputs all possible spatial descriptions of the next action using metric, directional and topological operators as seen in (Borrmann and Rank, 2009). For instance, "*Stick* a blue cube, *East of* the previous cube then *move it two slots to the South*". *Stick* translates to the blue cube touching the previous one which is a topological operator. *East of* is a directional operator. Lastly, *move it two slots to the South* is a metric putting forward the distance between the two objects in a certain direction.

3.2.2 Semantic generator

Given all possible actions delivered by the depictor, this layer filters/prioritizes the output list of the depictor according to the style policy: efficiency of the description in terms of positioning (e.g. use of centering, alignments), displacements, use of context, etc. When a chosen description is potentially ambiguous, it may add to the corresponding action extra verification(s).

The context is coming from the hierarchy of the tasks delivered by the planner (e.g. Without context: *"Put a red brick on top of the previous red brick"*. vs. with context: *"To finish the red tower, put a red brick on top of the previous red brick"*). Note that it is also responsible for adding information on the addressee (who should perform the task) and the task (e.g. explaining what it consists of and why this action is triggered, e.g. *"Let's start building an arch starting with its north pile!*).

3.2.3 Syntactic generator

The syntactic generator is responsible for building a syntactic tree with proper verbal constructs, names of objects, etc

Finally, in this layer, we have the option of either including all of the information (verbose) or omitting any redundant information as well as including pronominalisation, all while preserving the unicity of the task. (concise). Verbose: *"Put a red brick on top of the previous red brick"*. Concise: *"Put another one on top"*. The verbose option is straightforward and simply includes everything there is to know about the action. When applying the concise option, in order to remove redundant information, we need to consider what was previously manipulated by the human agent (i.e. LEGO™ type, color, orientation, task).

3.2.4 Realizer

It converts a syntactic tree into sentences. We used the jsRealB (Molins and Lapalme, 2015), that can handle both English and French (fig. 7).

3.2.5 Text To Speech

We currently use the macOS TTS. One issue that we have encountered is problematic mispronunciations in French (in particular handling homographs, liaisons, etc). The current TTS also does not allow to change the intonation in case we decided to manipulate the style of speech (e.g. instructing an order or an astonishment), nor can we include pauses to include coordination with gesture and gaze. Future work will include the use of a different TTS which allows controlling expressivity and rhythm as well as adding emphasis on certain parts of the text.

Going back to the idea of parametrizing the module for another task, a spatial task to be precise, a navigation task for instance. Aside from the necessary updates in the domain and problem files of the planner module corresponding to the new task at hand, some changes need to occur in the first two layers of the verbalizer. The depictor would still generate the semantic depiction and the relative positions between the objects, however, we would need to introduce the newly added different types of objects from the new environment (e.g. immovable obstacles, roads, traffic lights) and actions ("turn", "cross", "look", etc) as well as some information about the role and the link between these objects. The semantic generator would have the same objective as well, however, changes might

Figure 7: English and French realization using jsRealB

be required to accommodate with the type of information that needs to be transmitted. As for the rest of the layers, no particular update is required since it only concerns the styling, generation and utterance of the sentence.

It is important to mention that previous work, such as (Dogan et al., 2020), have shown that including perspective-taking helps reduce time and error. However, their work also mentions that the use of 'in front' and 'behind' did generate some ambiguity and caused more time and errors when applying a task. In our work, the use of left and right could have been easily used instead of East and West since we know the user's position with respect to the environment. However, we decided to use conventional directions (i.e. North East West South) instead of using perspective taking, (1) to ensure the absence of any ambiguity and (2) since this formulation can be used to instruct multiple users having different perspectives.

The following sections include the three web-based experiments that we conducted for finding out which (depiction/verbalization parameters) combination offers the best reduction of errors to complete the assembly.

4 Experiments

The purpose of the first two experiments is to provide us with a ground-truth corpus of verbal descriptions and obtain the highest and lowest ranked human exemplars for giving an instruction. The third experiment introduces an assembly task in order to test the efficiency of AI-generated instructions with reference to the natural ones.

Following the spatial representation — using metric, directional and topological operators as well as using multiple types of object references (point/corner, line/axis . . .) (Borrmann and Rank, 2009) — we chose a set of elementary actions which (1) spans most of these operators and (2) allows the implicit use of the context of that action. Thus, the instructions studied in the following describe the placement of the first brick of a new structure, i.e. giving the semantic generator the possibility to refer (or not) to the just finished one.

We describe below how we use crowdsourcing to gather human descriptions of these placements (mainly to parametrize the semantic and syntactic generator) and compare the efficiency of human vs. automatic descriptions. For this, *we asked subjects to actually perform the actions*. We expect conditions (human vs. AI-generated utterances, effective vs. no use of context) will impact placement error or time-to-complete.

All experiments are performed in French.

4.1 Collection of ground-truth data

We collected and ranked verbal descriptions performed by human subjects in two steps:

Free descriptions provide us with a ground-truth corpus of verbal descriptions of elementary placements performed by human subjects in which they put themselves in the place of a robot instructor.

Ratings of the former verbalizations were then collected by asking subjects to put themselves in the place of human partners and listen to the robot's instructions.

Scene presentation. In both sub-tasks, subjects are presented with bricks laid on a board. The brick just placed by the robot and the one to be placed by the subjects are respectively displayed with back edges vs. 50% transparency. The following *video* shows the screen during the experiment.

Instructions. The proposed vs ranked verbal descriptions should be unambiguous and as short as possible. Before the actual test (24 scenes), we

trained the participants with three examples containing some incorrect propositions. The results are used to validate crowdsourced data. We give the participants additional instructions when describing an action, namely, the use of specific terms (e.g. cube, brick, east-west orientation, north of) and the possibility to refer to the previously laid brick or any overtly constructed structure.

Subjects. The *free descriptions* were performed by the authors and 10 French-speaking participants while 15 participants recruited through Prolific performed the *ranking*.

Analysis of free descriptions. We combined descriptions performed by the authors with the ones suggested by the 10 participants. We manually selected an average of 5 *natural instructions* per scene in order to ensure that there were no duplicates, mistakes or ambiguities while trying to span as closely as possible the variety of styles, in particular syntactic constructs, topological properties of objects, etc.

Feature selection. Then, amongst all the sentences, we gather the following key parts which are essential or helpful for the action description: Hierarchy (Hierarchical Planning) and precedence between actions in the assembly (Short and long term recall when referencing objects/landmarks). The different reference types being *the previously placed element*, *a built structure* (e.g. tour, bridge, staircase etc) and *a part of an element:* (e.g. sides, corners, section of a structure etc). Aside from the reference object, an instructed action can be *decomposed* using multiple sub-actions or it can contain *verification* (i.e. additional information for validating the executed action). We note that the topological features and types of objects that can be found in our suggested scenes are taken from Borrmann and Rank (2009).

Analysis of the ranking. Following this phase, we repeat the same experiment with the same example and test sets along with the updated set of scene descriptions. However, we only ask the participants to choose the best, among the new list of natural instructions (see video). 15 participants are recruited through Prolific. The participants have to be French native speakers. The reason behind this experiment is to check which criteria a human agent would prefer as an instruction (e.g. including, or not, a verification step).

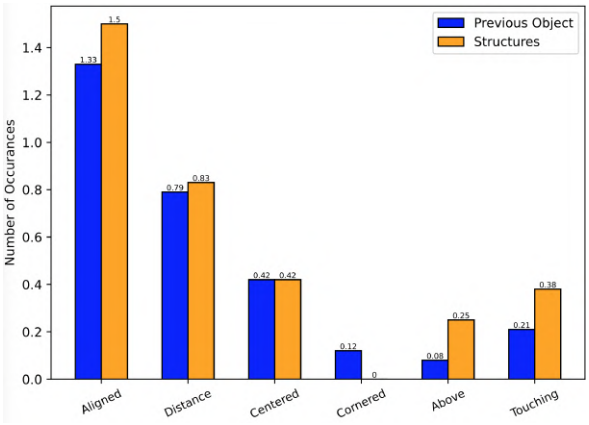


Figure 8: Frequency of appearance of topological operators in the proposed scenes

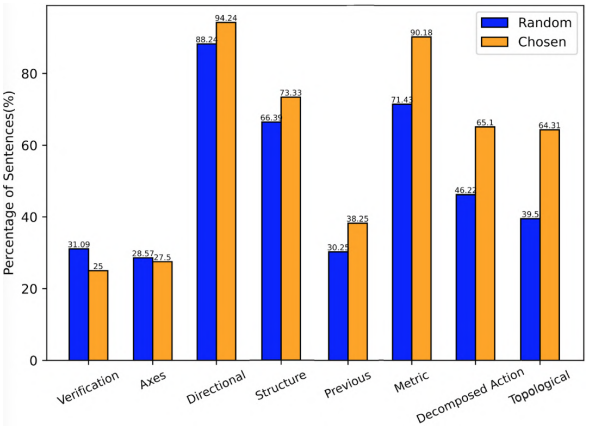


Figure 9: Results of all description criteria (by chance, and chosen by the 15 participants)

4.1.1 Results

As instructed, the participants are to choose a compact description of the action, all while being unambiguous and easy to understand. Amongst all the participants over all the scenes, 58.33% of the chosen sentences are the shortest ones. Suggesting that even with the instruction of 'choosing a compact sentence', longer sentences (usually caused by adding a verification step, or using multiple sub-actions) are also considered by the participants.

Figure 9 compares the percentage of sentences chosen at random with the ones chosen by the participants when a certain criterion is provided. In other words, the more the participants prefer a criterion, the bigger the difference will be between both percentages for that criterion. An instruction is a combination of multiple criteria, however the results still show a difference of preference between some of them. Going from the right, we see that when the feature is provided, the use of 'Decomposition' (decomposing an instruction), 'Metric' (numerical distance), and 'Topological' (touching reference)} were mainly preferred. Then 'Previous'

(mention of previously placed object), 'Structure' (mention of context and structures), and 'Directional' (3D directions) were also preferred but with a smaller percentage. The final two criteria {Verification, Axes} were disliked by the participants. Despite the fact that including a 'Verification' step ensures the correctness and reduces the ambiguity in an instruction, only 25% of the times do the participants choose the option with verification. This might be due to the fact that the participants are steering away from longer sentences containing this verification step. Lastly, we notice that the 'Axes' criterion corresponding to alignment is not largely preferred, which might be caused by the complexity of the positioning compared to other options.

Table 1: Subjective evaluation of different aspects of our verbalisation: 1:Strongly disagree - 5:Strongly agree

Questions	Score
1- Utterances were generated by a computer	4.3
2- Instructions were unambiguous	2.9
3- I prefer instructions referring to structures in place	3.98
4- Utterances were spelled clearly	3.65
5- Syntax was correct	4.15
6- I prefer instructions referring to the brick just placed by the robot	3.55
7- Utterances were generated by humans	3.33
8- The complexity of sentences were well adapted to the task	3.05

Table 2: Subjective evaluation of the participants' mental charge: 1:Strongly disagree - 5:Strongly agree

Questions	Score
1- The task was highly demanding	3.9
2- The pace of the task was too fast	2.9
3- You managed to accomplish what you were told to do	3.23
4- You worked hard to achieve your level of performance	3.9
5- You were insecure and stressed	2.48

Table 3: Different styles of sentences for the scene in fig.10

Type	Sentence
robot	Pour terminer la tour Sud, je mets un cube rouge ici.
worst_NI	Empile une barre bleue orientée Est-Ouest pour recouvrir exactement le haut de l'escalier, et le pilier qui est à l'Ouest de ce dernier.
best_NI	Dépose une barre bleue recouvrant complètement la tour rouge au Nord et le sommet de l'escalier jaune.
without_context	Place une barre bleue orientée Est-Ouest dont le côté Ouest doit être aligné avec celui du cube précédent laissant deux tenons libres vers le Nord.
with_context	Pour faire un pont, place une barre bleue orientée Est-Ouest qui recouvre le sommet jaune de l'escalier et le sommet rouge de la tour Nord.

4.2 Task Assembly

The previous experiments helped us to identify the key features used and preferred by humans for verbally instructing an action. We now test the impact of this verbal instruction on effective action performance.

Figure 10 shows an example scene and table 3 shows the sentences for that scene. The line 'Robot' gives the sentence accompanying the robot's first action. The other four sentences correspond to the

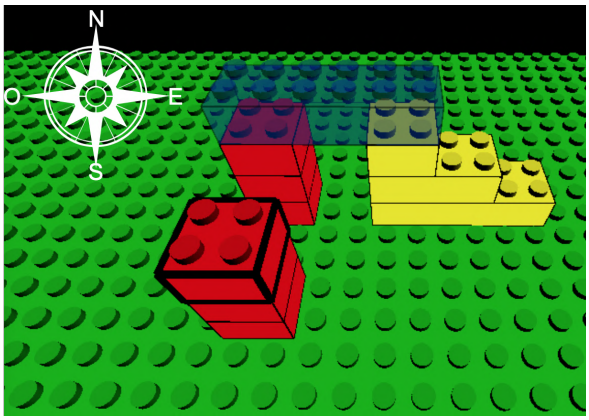


Figure 10: Scene example, having the bold red cube as the previously placed cube by the robot, and the transparent blue bar as the new object to be placed by the human agent

different styles we are comparing when instructing a participant. At first glance, we see in this example some difficulty of giving an instruction without the use of structures/context. This is why both highest and lowest ranked natural sentences use context and structures, suggesting their importance when giving the instruction.

Scene presentation. The participants were asked to observe the robot placing a brick on the game board, and then to continue the assembly according to its verbal instructions (see video). We use the same scenes as before and increase the test set using data augmentation (mirroring along the north/south axis), resulting in 54 scenes in total (3 training scenes & 51 test scenes).

Subjects. 40 French native speakers are recruited through the Prolific platform. 86.79% are right handed, 50% identified as men and 88.67% have already played with LEGO™ before this experiment.

Instructions and conditions. They have to place the right element as instructed, accurately and as fast as possible (see video). We have 4 instruction styles: (a) Lowest preference rate from the data-collection experiments ($worst\_NI$). (b) Highest preference rate from the data-collection experiments ($best\_NI$). (c) AI-generated description without mention of structures ($without\_context$). (d) AI-generated description with mention of structures ($with\_context$). The 4 styles are equally distributed among the 40 participants so that each scene with a given style is exactly performed by 10 participants.

Final questionnaires. We also include a two-part, 5-point Likert scale, questionnaire at the end of the

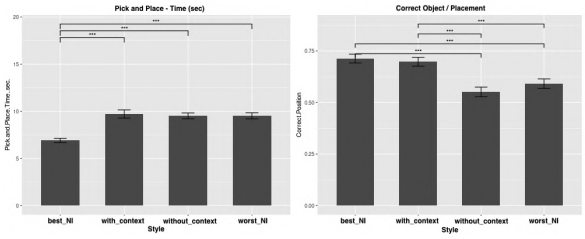


Figure 11: Normalized results over the 4 different styles – scenes with only one structure as available reference

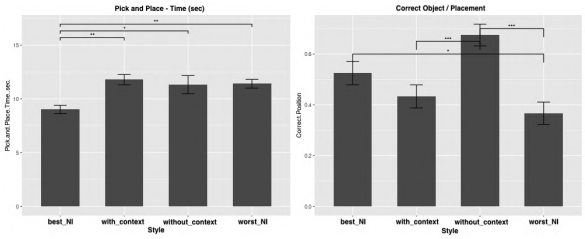


Figure 12: Normalized results over the 4 different styles – scenes with multiple structures as available references

experiment: (a) the first one evaluates the different parts of our verbalizer (table 1); (b) the second one (table 2) uses the NASA Task Load Index (NASA-TLX) (Hart and Staveland, 1988), which is the most common, subjective, multidimensional framework Colligan et al. (2015) to measure the cognitive load.

Results. For each of the 4 styles, we measure 5 cues: (a) the pick_to_place time (i.e. time between the first chosen object and the final release), (b-c) the number of chosen objects and positions to evaluate the participants' hesitation, (d-e) the percentage of correct selections and placements to evaluate performance. Results are given in Figures 11-12.

We select scenes involving more than 2 bricks but less than 2 laid structures (see Figure 11). We fit a linear mixed-effects model (*lmr* from *lme4* R package) with the scene as additional factor and subjects as random effect, and performed a post-hoc Tukey adjustments for pairwise comparisons (*ght* from *multcomp* R package). The highest ranked natural instruction significantly outperforms ($p < 10^{-3}$) the three other styles for time-to-complete and all but best-AI for successful completion. For scenes 21-24 with more than 2 laid structures (see Figure 12), AI-generated descriptions without mention of structures unexpectedly outperforms ($p < 10^{-3}$) all others for successful completion at a large margin: it seems that complex calculations seduce human intelligence but penalize performance. It also mirrors the findings of the data collection: people propose the use of axes and verifications (in experiment 1) but dislike them when asked to choose the best instruction (in experiment 2).

The verbalisation questionnaire (Tab. 1) shows that the verbalizer is working adequately: instructions are syntactically correct and clear, do not contain any major ambiguities and are properly uttered. The participants agree that both the use of hierarchical context and the mentioning of previously placed objects are a plus, while still leaning more towards the use of the former.

The NASA-TLX questionnaire (Tab. 2) shows that the experiment does require effort and cognitive load. It also shows that the participants are fairly satisfied with the rhythm, do not have much to say about their performance and do not express important signs of stress/frustration.

5 Conclusions and Future Work

We evaluate the impact of using hierarchical *context* when giving instructions in an assembly task. We gathered and ranked crowdsourced human instructions. We set up a multi-layer verbalizer that computes AI-generated instructions. We then compared the performance of these verbalization policies on a web-based assembly task. The differences between users' preferences and actual performances claim for an evaluation method in two steps: first, selecting candidate policies by subjective preference but then assess their efficiency by objective performance. We see that referring to hierarchical context improves human performance, compared to refraining from using it, in particular when context is unambiguous. We also show that our AI-generated instructions often outperform the least popular human instructions, validating the efficiency of our verbalizer.

While verbalizing, pointing towards the intended object would improve the understanding of the robot's intention, and reduce the effort in the verbal explanation to ensure task completion. Therefore, future work will include this modality in the action layer of our architecture along with speech-hand-gaze coordination. Incremental monitoring of actions by perception, in particular for on-line comprehension and attention, is a key issue for HRI.

Acknowledgments

This work is funded by MIAI (ANR-19-P3IA-0003).

References

Rachel Virgínia Xavier Aires, Aline M. P. Manfrin, Sandra M. Aluísio, and Diana Santos. 2004. What is my style? using stylistic features of portuguese web texts to classify web pages according to users' needs. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association.

Rachid Alami, Raja Chatila, Sara Fleury, Malik Ghallab, and Félix Ingrand. 1998. An architecture for autonomy. *Int. J. Robotics Res.*, 17(4):315–337.

Gregor Behnke, Pascal Bercher, Matthias Kraus, Marvin R. G. Schiller, Kristof Mickeleit, Timo Häge, Michael Dorna, Michael Dambier, Dietrich Manstetten, Wolfgang Minker, Birte Glimm, and Susanne Biundo. 2020. New developments for robert - assisting novice users even better in DIY projects. In *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*, pages 343–347. AAAI Press.

Robert S. Belvin, Ron Burns, and Cheryl Hein. 2001. Development of the HRL route navigation dialogue system. In *Proceedings of the First International Conference on Human Language Technology Research, HLT 2001, San Diego, California, USA, March 18-21, 2001*. Morgan Kaufmann.

Dan Bohus, Chit W. Saw, and Eric Horvitz. 2014. Directions robot: in-the-wild experiences and lessons learned. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 637–644. IFAAMAS/ACM.

André Borrmann and Ernst Rank. 2009. Topological analysis of 3d building models using a spatial query language. *Adv. Eng. Informatics*, 23(4):370–385.

Gerard Canal, Senka Krivic, Paul Luff, and Andrew Coles. 2021. Task plan verbalizations with causal justifications. In *ICAPS 2021 Workshop on Explainable AI Planning (XAIP)*.

Justine Cassell and Timothy Bickmore. 2003. Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User modeling and user-adapted interaction*, 13(1):89–132.

L. Colligan, H. Potts, Chelsea T. Finn, and R. A. Sinkin. 2015. Cognitive workload changes for nurses transitioning from a legacy system with paper documentation to a commercial electronic health record. *International journal of medical informatics*, 84 7:469–76.

Fethiye Irmak Dogan, Sarah Gillet, Elizabeth J. Carter, and Iolanda Leite. 2020. The impact of adding perspective-taking to spatial referencing during human-robot interaction. *Robotics Auton. Syst.*, 134:103654.

Michelangelo Fiore, Aurélie Clodic, and Rachid Alami. 2014. On planning and task achievement modalities for human-robot collaboration. In *Experimental Robotics - The 14th International Symposium on Experimental Robotics, ISER 2014, June 15-18, 2014, Marrakech and Essaouira, Morocco*, volume 109 of *Springer Tracts in Advanced Robotics*, pages 293–306. Springer.

Kate Forbes-Riley, Diane Litman, and Mihai Rotaru. 2008. Responding to student uncertainty during computer tutoring: An experimental evaluation. In *International Conference on Intelligent Tutoring Systems*, pages 60–69. Springer.

Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek P. Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid G. Simmons, Kevin Snipes, Alan C. Schultz, and Jue Wang. 2005. Designing robots for long-term social interaction. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada, August 2-6, 2005*, pages 1338–1343. IEEE.

Guillermo Gomez, Carole Plasson, Frédéric Elisei, Frédéric Noël, and Gérard Bailly. 2015. Qualitative assessment of an immersive teleoperation environment for collaborative professional activities in a" beaming" experiment. In *EuroVR 2015-European conference for Virtual Reality and Augmented Reality*, pages 8–pages.

Sandra G Hart and Lowell E Staveland. 1988. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier.

Jussi Karlgren. 2000. *Stylistic Experiments for Information Retrieval*. Ph.D. thesis, Royal Institute of Technology, Stockholm, Sweden.

Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. 1997. Robocup: The robot world cup initiative. In *Proceedings of the First International Conference on Autonomous Agents, AGENTS 1997, Marina del Rey, California, USA, February 5-8, 1997*, pages 340–347. ACM.

David Kortenkamp, Reid Simmons, and Davide Brugali. 2016. *Robotic Systems Architectures and Programming*, pages 283–306. Springer International Publishing, Cham.

Alastair MacFadden, Lorin Elias, and Deborah Saucier. 2003. Males and females scan maps similarly, but give directions differently. *Brain and Cognition*, 53(2):297–300.

François Mairesse and Marilyn A. Walker. 2010. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Model. User Adapt. Interact.*, 20(3):227–278.

Juliana Miehle, Wolfgang Minker, and Stefan Ultes. 2018a. What causes the differences in communication styles? A multicultural study on directness and

elaborateness. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Juliana Miehle, Wolfgang Minker, and Stefan Ultes. 2018b. What causes the differences in communication styles? a multicultural study on directness and elaborateness. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Paul Molins and Guy Lapalme. 2015. Jsrealb: A bilingual text realizer for web programming. In *ENLG 2015 - Proceedings of the 15th European Workshop on Natural Language Generation, 10-11 September 2015, University of Brighton, Brighton, UK*, pages 109–111. The Association for Computer Linguistics.

Stefanos Nikolaidis, Minae Kwon, Jodi Forlizzi, and Siddhartha S. Srinivasa. 2017. Planning with verbal communication for human-robot collaboration. *CoRR*, abs/1706.04694.

Damien Pellier and Humbert Fiorino. 2018. PDDL4J: a planning domain description library for java. *J. Exp. Theor. Artif. Intell.*, 30(1):143–176.

Vittorio Perera, Sai P. Selvaraj, Stephanie Rosenthal, and Manuela M. Veloso. 2016. Dynamic generation and refinement of robot verbalization. In *25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016, New York, NY, USA, August 26-31, 2016*, pages 212–218. IEEE.

Ronald Petrick and Mary Ellen Foster. 2013. Planning for social interaction in a robot bartender domain. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 23, pages 389–397.

David Reitter, Frank Keller, and Johanna D. Moore. 2006. Computational modelling of structural priming in dialogue. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 4-9, 2006, New York, New York, USA*. The Association for Computational Linguistics.

Stephanie Rosenthal, Sai P Selvaraj, and Manuela M Veloso. 2016. Verbalization: Narration of autonomous robot experience. In *IJCAI*, volume 16, pages 862–868.

Svetlana Stenchikova and Amanda Stent. 2007. Measuring adaptation between dialogs. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, SIGdial 2007, Antwerp, Belgium, September 1-2, 2007*, pages 166–173. Association for Computational Linguistics.

Manuela M. Veloso, Nicholas Armstrong-Crews, Sonia Chernova, Elisabeth Crawford, Colin McMillen, Maayan Roth, Douglas L. Vail, and Stefan Zickler. 2008. A team of humanoid game commentators. *Int. J. Humanoid Robotics*, 5(3):457–480.

Dirk Voelz, Elisabeth André, Gerd Herzog, and Thomas Rist. 1998. Rocco: A robocup soccer commentator system. In *RoboCup-98: Robot Soccer World Cup II*, volume 1604 of *Lecture Notes in Computer Science*, pages 50–60. Springer.

Terry Winograd. 1972. Shrdlu: A system for dialog.

Terry Winograd. 1974. Five lectures on artificial intelligence. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.

Qingxiaoyang Zhu, Vittorio Perera, Mirko Wächter, Tamim Asfour, and Manuela M. Veloso. 2017. Autonomous narration of humanoid robot kitchen task experience. In *17th IEEE-RAS International Conference on Humanoid Robotics, Humanoids 2017, Birmingham, United Kingdom, November 15-17, 2017*, pages 390–397. IEEE.

A Appendix

A.1 Obtaining natural exemplars

We present here, a screen capture from the crowd-sourcing experiment[1]. The left part of figure 13 shows the fixed working environment. The right part shows a list of proposed naturally written sentences where the participants needs to choose the best action description (first objective).

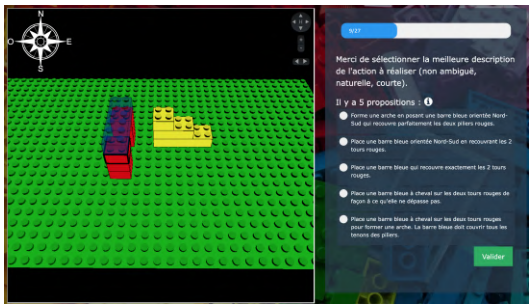


Figure 13: Describing the action

In figure 14, we show the second objective of the task, getting a suggestion of an action description from the participant.

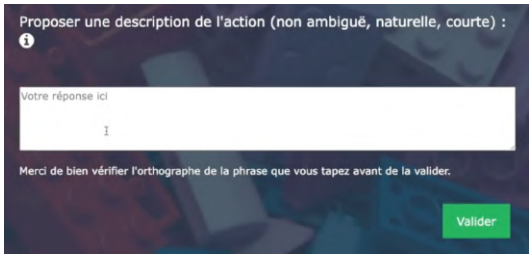


Figure 14: Participant inputs their description of the next action (scene in appendix fig. 15)

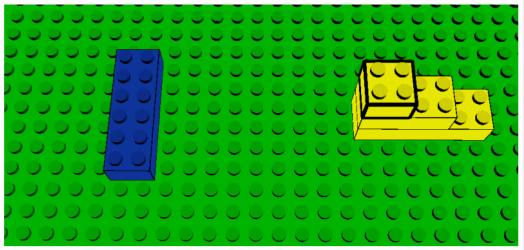


Figure 15: A 3D scene containing, the last added element being the yellow cube on top of a staircase and the next object to be added being the blue brick on the left

[1] https://youtu.be/D5fPfKz8c8Q – video

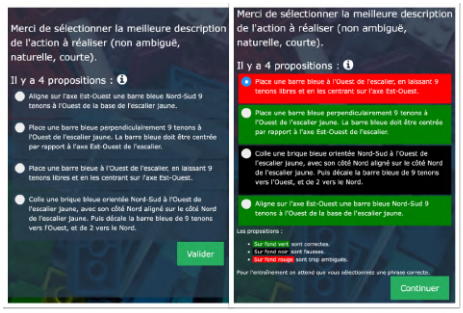


Figure 16: On the left, our proposed sentences. On the right, explanation of the correctness of each choice during the training phase. (scene in appendix fig. 15)

A.2 Data Collection

This section – the second experiment[2]

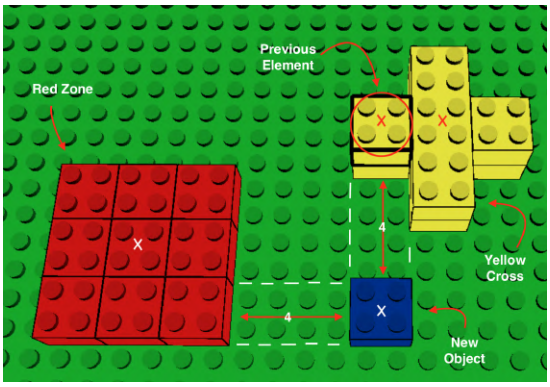


Figure 17: Different criteria (topological) found in a 3D scene

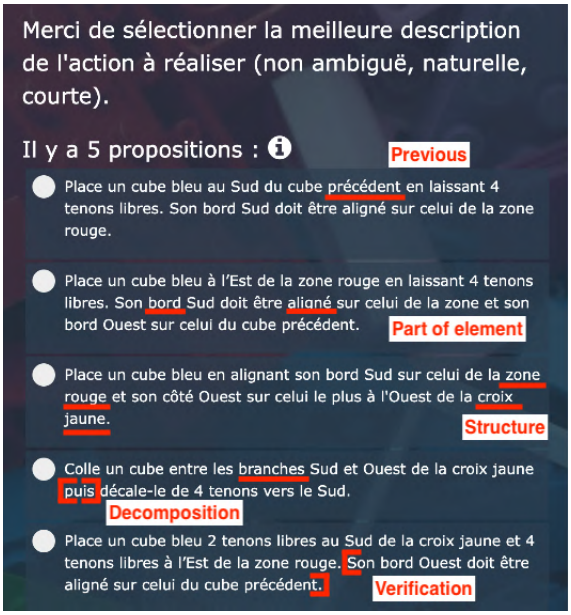


Figure 18: An example of the proposed action descriptions for the scene in fig. 17

[2] https://youtu.be/uS65RWLmpWw – video

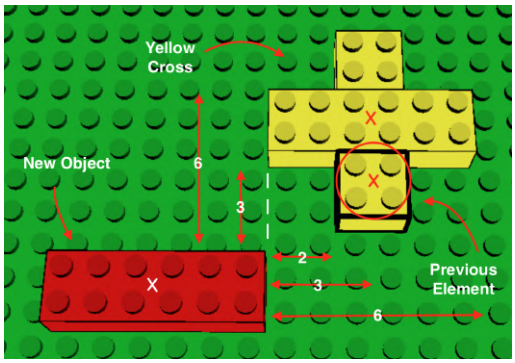


Figure 19: Different criteria found in a 3D scene

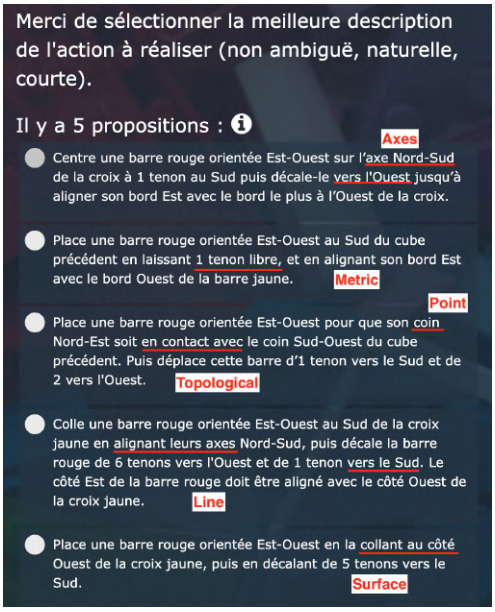


Figure 20: An example of the proposed action descriptions for the scene in fig. 19

A.3 Assembly

We present here, a screen capture from the assembly task experiment[3]. The top part of figure 21 shows the robot placing a yellow cube in order to finish a staircase. The bottom part shows a human participant trying to accomplish a task after receiving a detailed instruction, from the system, on how to do so.

Below, we have a screen capture of the objective evaluation of our assembly task with figure 22 corresponding to the participants' evaluation of the verbalisation and figure 23 corresponding to the participants' evaluation of the concerned mental charge.

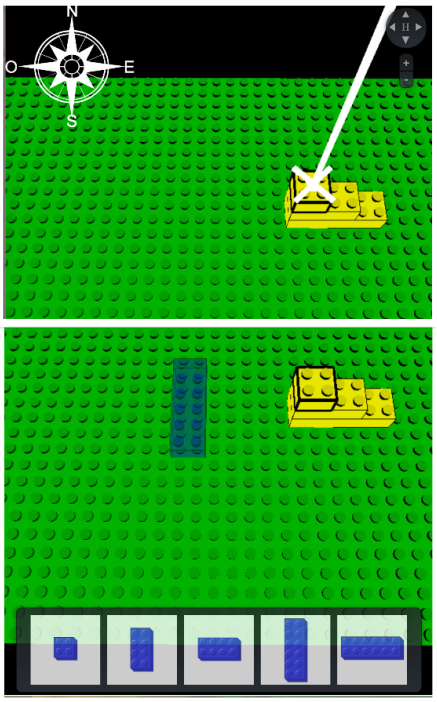


Figure 21: **Top**: Robot's action. **Bottom**: participant's action

Figure 22: Questions on the verbalisation

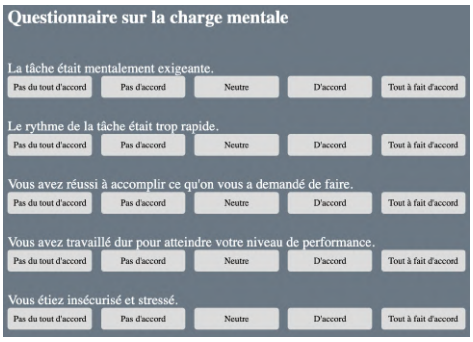


Figure 23: Questions on mental load – taken from NASA-TLX

[3]https://youtu.be/harvF23E_dI – video

Are Interaction Patterns Helpful for Task-Agnostic Dementia Detection? An Empirical Exploration

Shahla Farzana and **Natalie Parde**
Natural Language Processing Laboratory
Department of Computer Science
University of Illinois Chicago
{sfarza3, parde}@uic.edu

Abstract

Dementia often manifests in dialog through specific behaviors such as requesting clarification, communicating repetitive ideas, and stalling, prompting conversational partners to probe or otherwise attempt to elicit information. Dialog act (DA) sequences can have predictive power for dementia detection through their potential to capture these meaningful interaction patterns. However, most existing work in this space relies on content-dependent features, raising questions about their generalizability beyond small reference sets or across different cognitive tasks. In this paper, we adapt an existing DA annotation scheme for two different cognitive tasks present in a popular dementia detection dataset. We show that a DA tagging model leveraging neural sentence embeddings and other information from previous utterances and speaker tags achieves strong performance for both tasks. We also propose content-free interaction features and show that they yield high utility in distinguishing dementia and control subjects across different tasks. Our study provides a step toward better understanding how interaction patterns in spontaneous dialog affect cognitive modeling across different tasks, which carries implications for the design of non-invasive and low-cost cognitive health monitoring tools for use at scale.

1 Introduction

A recent surge of interest in automated assessment of cognitive health within the speech and language processing communities (Zhu et al., 2019; Di Palo and Parde, 2019; Farzana and Parde, 2020; Luz et al., 2020, 2021) has spurred the development of high-performing diagnostic models. These models carry the potential for substantial real-world positive impact, offering an affordable and accessible healthcare screening solution for individuals who may otherwise be under-served (Petti et al., 2020). However, recent cognitive assessment models have generally been constrained to specific tasks, each with their own characteristics and requirements. Although this facilitates the development of models that excel at their target task (e.g., predicting which users have Alzheimer's disease in a picture description task (Luz et al., 2020)), it creates challenges in building generalizable knowledge about the complex relationship between linguistic or verbal behavior and cognitive status. It can be unclear which findings are task-specific, and which may be applicable to different tasks in related settings.

In this work, we set out to provide clarity regarding the generalizability of a category of features that have held promise for task-specific cognitive assessment. Specifically, we examine facets of individuals' interaction patterns, which have been recognized as predictive of Alzheimer's disease or related dementia (AD) in sociolinguistic studies (Orange et al., 1996; Elsey et al., 2015; Hamilton, 1994) and proved informative for automatically detecting AD in task-specific settings (Nasreen et al., 2021; Mirheidari et al., 2019). We do so by adapting an existing dialog act (DA) annotation scheme (Bunt, 2006; Farzana et al., 2020), previously used to analyze dialogs from AD and control participants, to two distinct cognitive tasks and study subjects' interactions across tasks. We also examine the use of interaction features derived from these DA tags in dementia detection models to assess their task-agnostic utility in this domain. Our key contributions are as follows:

- We adapt a DA annotation scheme for two cognitive tasks in a popular dementia detection corpus and present comparative analyses of subjects' interaction patterns across tasks.
- We develop a DA tagging model using this scheme and show that it achieves strong performance (F_1=0.82) when trained on both tasks jointly. The model leverages neural sentence embeddings, part-of-speech (POS) tags, previous utterances, and speaker information

Proceedings of the SIGdial 2022 Conference, pages 172–182
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

to make its predictions.

- We propose a set of content-free interaction features for task-agnostic dementia detection and show that they yield high utility in distinguishing between dementia and control subjects across different tasks.

We describe these contributions further in the remainder of this paper. In §2, we review relevant background to position our work within the broader research landscape. In §3, we present our methods for modeling DA sequences using the selected DA scheme (§3.1) and developing task-agnostic interaction features within this domain (§3.2). We describe our data in §4, our experiments in §5, and our results in §6, before concluding in §7.

2 Background

2.1 Interaction Patterns and AD Detection

Conversation analysis has proved to be effective for detecting dementia and tracking its progression through the study of user intent, clarification and verbal disfluency frequency, and other discourse cues (Mirheidari et al., 2019; Orange et al., 1996; Farzana et al., 2022). Speech-based interaction features like average turn duration, total turn duration, and average number of words per minute have been utilized to model conversation dynamics in the context of AD detection (Luz et al., 2020). However, many of these features are task-specific, and focus only on the participants' part of the dialog. Nonetheless, fine-grained analysis of question-answer ratio has been the focus of several studies showing promising performance on dementia detection (Hamilton, 1994; Varela Suárez, 2018).

Dialog act-based conversation analysis was first introduced by Farzana et al. (2020), capturing the interaction patterns from DementiaBank's (Becker et al., 1994) semi-structured picture description task in terms of different DAs from both the subject and interviewer. Similar corpus analyses on the Carolinas Conversation Collection (CCC) (Pope and Davis, 2011) by Nasreen et al. (2019) observed that interaction patterns like signal non-understanding and clarifying questions are more evident in cognitively challenged subjects than healthy controls, and leveraged DA features to model dementia detection (Nasreen et al., 2021). Speaker turn sequence processing has also previously been used to model intervention patterns (Sarawgi et al., 2020), and leveraging acoustic, linguistic, and fusion features to represent conversations between interviewers and participants has shown promising performance in AD detection (Pérez-Toro et al., 2021). Most of these experiments have been evaluated on task-specific corpora, including semi-structured cognitive screening interviews like the picture description task (Roth, 2011) or more open-ended tasks in which subjects talk about their health (Pope and Davis, 2011)). Modeling task-agnostic linguistic anomalies to detect dementia from casual conversations has been studied very recently (Li et al., 2022), although this work did not extend its study to interaction style.

2.2 DA Tagging and AD Detection

DA recognition is known to be a complex problem, and many approaches ranging from multi-class/multilabel classification to structured prediction have sought to tackle it (Stolcke et al., 2000; Yang et al., 2009). Performing DA classification effectively enables the development of high-quality natural language dialogue systems (Higashinaka et al., 2014). Previously, a context-aware deep neural model leveraging a hierarchical recurrent network and self attention mechanism (Raheja and Tetreault, 2019) achieved state-of-the-art performance in DA tagging on the SWDA corpus (Jurafsky et al., 1997), a standard benchmark for this task.

Most DA tagging corpora are highly imbalanced, so a crucial shortcoming of most high-performing DA tagging models is that in focusing on improving overall performance, they end up performing poorly on rare class DAs. These DA classes can be critical for modeling conversations in cognitive health screening tasks (Farzana et al., 2020; Nasreen et al., 2021). Thus, DA tagging models tailored more specifically for AD detection settings may be needed to facilitate sufficient understanding and analysis of interaction patterns.

3 Methods

3.1 DA Tagging Model

Our initial dialogue act recognition model trained on Farzana et al. (2020)'s *Cookie-Theft DA* dataset is a multi-layer perceptron (MLP) adapted from a model introduced in prior work (Martínek et al., 2021). Each utterance, consisting of a variable number of words, is first encoded into a single pre-trained 1024-dimensional sentence embedding vector using a BERT Large (Reimers and Gurevych,

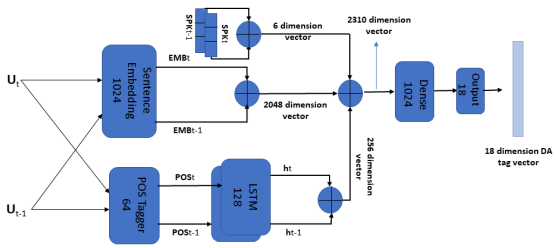


Figure 1: DA Tagging model architecture.

2019) encoder. As shown in Figure 1, our model computes two such vectors, respectively, for the context and current utterances. These vectors are concatenated and passed along as input to the MLP. We also incorporate utterance-wise part-of-speech (POS) tags generated using the pre-trained Stanford CoreNLP parser (Qi et al., 2020). To do so, we feed sequences of POS tags for the current and contextual utterance through an LSTM and concatenate its output with the previously computed semantic representation as shown in Figure 1.

We also add a speaker information vector indicating the speaker for a given utterance. We compute one speaker vector each for the current and context utterances and concatenate them with the previously created representation, ultimately resulting in a concatenation of numerous input vectors (utterances, POS sequences, and speaker tags) that is fed to the dense layer of the MLP, followed by the output layer. Following the training procedures later described in §5.1, we perform DA tagging experiments on two AD detection tasks separately and in a joint setting. In doing so, we seek to investigate the following topics pertaining to DA prediction: (1) the model's ability to generalize when predicting DAs for two different cognitive tasks, provided that the tasks share some common nuances in interaction style, yet differ in linguistic traits and overall objectives; and (2) the extent to (and ways in) which prediction accuracy for rare DAs differs when the model is trained jointly with a class-weighted loss function versus when the model is trained separately on single tasks.

3.2 AD Detection Features

To investigate the effects of interaction patterns on AD detection performance in numerous cognitive tasks, we made use of the DA tags as well as turn-based features, following their earlier success in prior work (Nasreen et al., 2021). We represented local interaction patterns as unigram, bigram, and trigram sequences of DA tags. To avoid sparsity, we filtered these n-grams based on their training set frequency differences between the AD and non-AD classes (i.e., only DA n-grams for which the between-class training set frequency differed by ≥ 5 were retained). To represent turn-taking patterns, we computed the following based on timing signatures from the transcripts:

- **Average Turn Duration:** The average length of a participant's turn, in milliseconds.
- **Total Duration:** The length of the full conversation (in milliseconds) between the participant and interviewer.
- **Normalized Turn Switch:** The average number of turn switches per minute (e.g., a minute of dialog with the turn sequence (*Participant* → *Interviewer* → *Interviewer* → *Participant*) would have two turn switches).
- **Average Words/Minute:** The average number of words spoken in a minute of recorded speech.

These features may provide valuable clues regarding the interaction patterns and approximate flow of communication in a given dialog. All turn-taking features were extracted using timings recorded for each utterance in the transcripts. Finally, we also incorporated ratio-based features to measure other aspects of the interaction patterns. We included the following ratio-based features:

- **Question Ratio:** The number of DAs tagged as *Request:Clarification* or *Question:General* from participants, normalized by all DA tags in the dialog.
- **Answer Ratio:** The number of DAs tagged as *Answer:General*, *Answer:Yes*, *Answer:No*, or *Acknowledgement* by an interviewer, normalized by all DA tags in the dialog.

These features were designed to capture various aspects of global interaction patterns that may have been missed by other feature groups, and have also shown promise in prior work on specific tasks (Nasreen et al., 2021; Khodabakhsh et al., 2015).

4 Data

4.1 Data Sources

We evaluated our DA tagging and AD detection models on two tasks in a subset of *DementiaBank*

	Cookie	Fluency
Gender	M=45, F=52	M=9, F=8
Age	90.29±35.01	66.47±8.41
Education	13.10 ±2.65	12.59±2.99
Onset Age	65.31±8.64	63.88±8.23

Table 1: Demographic information for both tasks. *Cookie* refers to the picture description task, and *Fluency* refers to the verbal fluency task. *Education* is in years. The *onset age* is the age of a participant when first diagnosed with AD.

known as the *Pitt corpus* (Becker et al., 1994). DementiaBank is a large database encompassing corpora pertaining to dementia submitted by numerous contributors around the globe. It includes corpora in multiple languages, spanning multiple cognitive tasks. The Pitt corpus is an English-language subset containing longitudinal dementia and control audiorecordings and associated transcripts for four language tasks, including picture description, verbal fluency, sentence construction, and story recall. We selected the picture description and verbal fluency tasks for our experiments.

The picture description task, known formally as the *Cookie Theft Picture Description Task* (Roth, 2011), is the most commonly studied task in research towards automated dementia detection, serving as the focus of two popular challenges in 2020 (Luz et al., 2020) and 2021 (Luz et al., 2021). It includes semi-structured interviews between an interviewer and a subject belonging to one of two groups (AD or non-AD). The subject is instructed to describe the contents of an eventful picture featuring, among other things, a child stealing a cookie. Previously, Farzana et al. (2020) annotated 100 transcripts from this dataset spanning 1616 utterances with 26 DA tags. The DA classes were adapted from the ISO Standard 24617-2 (Bunt, 2006) DA scheme, with the addition of 8 task-specific DAs.

The second task, designed to assess verbal fluency, features dialog between a participant and an interviewer. In the first segment of the interview, the participant is prompted to utter as many animal names as they can in one minute. In the second segment, they are instructed to utter as many words starting with *f* as they can within one minute.

For AD detection using the DA tags, we filtered the dataset such that each subject had one conversation.[1] We excluded three annotated conversations from Farzana et al. (2020)'s *Cookie-Theft DA* corpus, since two of the conversations belonged to a repeated participant (in different years) and the other's participant overlapped with one also present in the verbal fluency task. Altogether, our final dataset included 97 conversations (*non-AD*=46, *AD*=51) from the picture description task, and 17 (*non-AD*=2, *AD*=15) from the fluency task.[2] The annotations are available for the research community[3] for further followup work, and can be used after separately gaining access to DementiaBank.[4] Table 1 presents demographic statistics and Table 2 presents descriptive statistics for each task.

	Cookie	Fluency
# Conversations	97	17
Total Utterances	1569	760
Average Duration	672.43	147.29
Words/Minute	623.16	300.19

Table 2: Descriptive statistics for both tasks. Duration is in seconds, averaged across the number of conversations in the task.

4.2 Data Annotation

Although we were able to use the existing DA tags from *Cookie-Theft DA* directly, we manually annotated the 17 transcripts from the verbal fluency task with corresponding DAs. We followed the same guidelines established by Farzana et al. (2020), with minor task-specific adjustments. Specifically, we replaced the 8 task-specific DA tags corresponding to core topics in the cookie theft picture (denoted with labels *Answer:t1–Answer:t8* in *Cookie-Theft DA*) with two task-specific DA tags more closely aligned with the verbal fluency task; namely, *Answer:Topic1* and *Answer:Topic2*. *Answer:Topic1* is assigned to utterances in which participants refer to animal names, and *Answer:Topic2* is assigned to utterances in which participants say words beginning with the letter *f*. We distinguished these tags from one another to facilitate easy sepa-

[1] Since the Pitt corpus contains longitudinal data, some subjects have multiple entries for the same task, from initial and follow-up visits.

[2] We annotated 17 transcripts from verbal fluency task in *DementiaBank*, which is an imbalanced corpus with 2 and 239 transcripts from the *non-AD* and *AD* classes respectively, to avoid having a huge class imbalance in our resulting dataset.

[3] https://nlp.lab.uic.edu/resources/

[4] https://dementia.talkbank.org/

DA	Label	Example	Ratio
QUESTION: GENERAL	*qg*	do you know other types?	<0.1
QUESTION: REFLEXIVE	*qr*	a bird?	<0.1
ANSWER: YES	*ay*	yeah that's fine	<0.1
ANSWER: NO	*an*	I don't know	<0.1
ANSWER: GENERAL	*ag*	gosh I can't think of it	<0.1
INSTRUC-TION	*is*	words that begin with f	0.2
SUGGEST.	*sg*	just keep naming them	<0.1
ACK.	*ak*	okay good	0.1
REQUEST: CLAR.	*rc*	did I say facts?	<0.1
FEEDBACK: REFLEXIVE	*fr*	no that's not an animal	<0.1
STALLING	*sl*	oh let's see	<0.1
OTHER	*or*	&=laughs	<0.1
ANSWER: TOPIC	*at*	uh dog, &hm oh a fence	0.5

Table 3: DAs with non-zero frequency in our *Verbal Fluency DA* dataset, with examples. For DA tagging and AD detection, we reduce the task-specific DAs in both tasks to *Answer:Topic*. *Ratio* indicates the specified DA's frequency ratio for the *verbal fluency* task.

ration of tasks in later analyses.

Two graduate students annotated these transcripts adhering to the annotation guidelines published by Farzana et al. (2020), with new amendments added for the task-specific DA classes, after an initial training session with a practice transcript. They achieved strong inter-annotator agreement, as measured using Cohen's kappa (Cohen, 1960) with a score of $\kappa = 0.79$. The annotations were collected using the INCEpTION framework (Klie et al., 2018), a free, user-friendly, web-based annotation interface with built-in support for adjudication and assessment of inter-annotator agreement. Disagreements were forwarded to a third-party, expert adjudicator for final label selection. Table 3 presents example labeled utterances from our new *Verbal Fluency DA* corpus with a variety of DA tags.

5 Experiment

We conducted two core sets of experiments in this work. In the first set (§5.1), we evaluated the performance of our DA tagging model (described in §3.1) at correctly assigning labels from our annotation scheme to utterances in the picture description and verbal fluency tasks. In the second (§5.2), we measured the performance of features designed to capture meaningful interaction patterns using these DAs when leveraged in a dementia detection task.

5.1 DA Tagging

To evaluate the performance of our DA tagging model, we devised a series of experimental conditions featuring different components of interest in our study:

- **NO-CONTEXT:** The current utterance embedding. This was used as our baseline model.
- **n EMB.:** An utterance embedding history of length n is passed to the DA prediction model. For example, when $n = 1$, the current utterance is passed to the model, and when $n = 2$, the previous utterance is used as context along with the current one.
- **n POS:** A POS embedding history of length n is passed to the DA prediction model. For example, when $n = 1$, the POS tag sequence for the current utterance is passed to the model, and when $n = 2$, the POS tags for the previous utterance are used as context along with the current sequence.
- **n SPK.:** A speaker history of length n is passed to the DA prediction model. For example, when $n = 1$, the current speaker tag is passed to the model, and when $n = 2$, the speaker tag for the previous utterance is used as context along with the current speaker tag.

Studying performance under these different conditions allowed us to develop a fuller understanding of the contributions of individual components. To implement our DA tagging model, we used a neural network backbone with the fine-tuned hyperparameters: *learning rate* = 0.001, *Adam* optimizer (Kingma and Ba, 2015), *batch size* = 32, *epoch*

Feature	Details
Unigram	(*I_Instruction*), (*P_Request:Clarification*)
Bigram	(*P_Request:Clarification* + *I_Answer:General*), (*P_Stalling* + *I_Acknowledgment*)
Trigram	(*I_Instruction* + *P_Request:Clarification* + *I_Answer:General*)
Ratio + Turn-Taking	*Question Ratio*, *Answer Ratio*, *Average Turn Duration*, *Normalized Turn Switch*, *Total Duration*, *Average Words/Minute*

Table 4: Interaction pattern features for AD detection. Durations are in milliseconds (ms).

Features	Joint	Cookie	Fluency
1 EMB. (NO CONTEXT)	0.77	0.82	0.71
1 EMB. & 1 POS & 1 SPK.	0.77	0.82	0.73
2 EMB.	0.81	0.84	0.74
2 EMB. & 2 POS & 2 SPK.	0.81	0.83	0.74
1 EMB. & 1 POS	0.78	0.82	0.70
2 EMB. & 2 POS	0.79	0.84	0.75
1 EMB. & 1 SPK.	0.78	0.81	0.74
2 EMB. & 2 SPK.	**0.82**	**0.85**	**0.75**

Table 5: 10-fold cross-validation DA tagging results with micro-averaged F_1 scores on the *picture description* (Cookie), *verbal fluency* (Fluency) and *joint* tasks.

= 300, and early stopping criteria *min* $\delta = 0.0001$. We used a class-weighted categorical cross-entropy loss, since our class labels (for both the *picture description* and *verbal fluency* tasks) are imbalanced. Our utterance embeddings were computed using the *nli-bert-large* (Conneau et al., 2017) model with an embedding dimension of 1024 from the HuggingFace *sentence-transformers* library.

Finally, to capture our DA tagging model's ability to generalize, we also compared three versions of each condition. Specifically, we trained and evaluated the DA tagger on the *picture description* and *verbal fluency* tasks separately, and then we also trained and evaluated a *joint* model using data from both the tasks combined. This allowed us to empirically validate the feasibility of this model in several settings for later use in extracting AD detection features.

5.2 AD Detection

To evaluate the impact of our interaction features on classifying AD status in a task-agnostic setting, we performed experiments considering the following conditions:

- **ALL:** This condition utilizes all features included in Table 4 and described previously.
- **N-GRAM:** This condition includes all features in the rows corresponding to unigrams, bigrams, and trigrams in Table 4.
- **N-GRAM + TURN-TAKING:** This condition includes the union of all n-gram and interaction features listed in Table 4.
- **N-GRAM + RATIO:** This condition includes the union of all n-gram and ratio features listed in Table 4.

We implemented our AD detection model using a random forest classifier (*rfc*) with the following hyperparameters: *number of estimators*=100, *max depth*=10. We selected this model from among a pool of it and two other feature-based classification models (support vector models with polynomial and radial basis functions, respectively) based on preliminary performance validation experiments. Our choice of a feature-based classifier rather than more complex (and potentially higher-performing) neural network alternatives was driven by our need for easy interpretability, to analyze and compare features in task-agnostic settings.

6 Results

6.1 DA Tagging

We summarize the results of our DA prediction experiments in Table 5, comparing all conditions earlier described in §5.1. For the **n EMB.**, **n POS**, and **n SPK.** conditions, we use values of $n \in \{1, 2\}$. We limited our experiment to include only the immediate previous context ($n = 2$) since

	Accuracy	Precision		Recall		F1	
		AD	HC	AD	HC	AD	HC
BASELINE	0.58	1.00	0.00	1.00	0.00	0.73	0.00
ALL	**0.79**	0.80	**.77**	0.85	**0.71**	**0.82**	**0.74**
N-GRAM	0.68	0.72	0.63	0.74	0.60	0.73	0.62
N-GRAM + TURN-TAKING	0.74	0.76	0.70	0.79	0.67	0.78	0.68
N-GRAM + RATIO	0.71	0.76	0.65	0.73	0.69	0.74	0.67

Table 6: Five-fold cross-validation results, for models jointly trained on the *picture description* and *verbal fluency* tasks using gold-annotated DA tags. The baseline model predicted the most frequent class for each instance.

that contributed the strongest performance boost in prior research (Nasreen et al., 2021; Farzana et al., 2020). Our baseline model (NO CONTEXT) yielded micro-averaged F_1 scores of F_1=0.77, F_1=0.82, and F_1=0.71 on the *joint*, *picture description*, and *verbal fluency* training settings, respectively. The performance of F_1=0.82 for the *picture description* setting exceeds that of the highest-performing benchmarking model reported by Farzana et al. (2020).

The results were further improved by adding contextual information from previous utterances (2 EMB.), achieving scores of F_1=0.81, F_1=0.84, and F_1=0.74 on the *joint*, *picture description*, and *verbal fluency* training settings, respectively. Adding the previous utterance's POS sequences and speaker tag (2 EMB. & 2 POS & 2 SPK.) did not offer noticeable advantages beyond this, with nearly equivalent performance. Overall, we observe the strongest performance when contextual embeddings and speaker tags are used *without* contextual part-of-speech sequences (2 EMB. & 2 SPK.), achieving scores of F_1=0.82, F_1=0.85, and F_1=0.75 on the *joint*, *picture description*, and *verbal fluency* training settings, respectively.

When comparing training settings, we observe that the *picture description* setting consistently achieves the highest performance, followed by the *joint* setting and finally the *verbal fluency* setting. This makes sense intuitively. The *verbal fluency* dataset was the smallest, and its size may have interfered with the DA prediction model's ability to derive meaningful information from the feature set. The *joint* dataset was the largest, but it may have struggled to effectively distinguish between class traits that manifested differently in different tasks. The *picture description* task is the most well-studied, and the only one for which benchmarking results were available (Farzana et al., 2020). We note that all of our models exceeded Farzana et al. (2020)'s strongest benchmark (F_1=0.77).

6.2 AD Detection

We summarize the results of our AD detection experiments in Table 6. For these results, we employed the *joint* corpus and used a five-fold cross-validation training and evaluation setting, comparing the different feature combinations outlined in §5.2. We report precision, recall, and F_1 for each class (AD and healthy control participants without AD, referred to as HC), as well as overall accuracy. We observe the highest performance under the ALL condition, with per-class F_1 scores of F_1=0.82 and F_1=0.74 for the AD and HC classes, respectively, and an overall accuracy of 0.79. This provides evidence of meaningful contributions from all interaction features when used in a task-agnostic AD detection setting.

All AD detection models exceeded the baseline condition (predicting the most frequent class, AD, in all cases). When combined with the DA tag unigrams, bigrams, and trigrams, the turn-taking features (accuracy=0.74) outperformed the ratio-based features (accuracy=0.71), although both added utility beyond the DA tag n-grams alone (accuracy=0.68). At a per-class level, performance for the AD class exceeded that of the control class; this was expected given that the dataset included a higher percentage of AD than HC participants.

6.3 Discussion

The results observed from the AD detection experiments clearly suggest that features based on content-free interaction patterns are helpful for dementia detection classifiers in task-agnostic set-

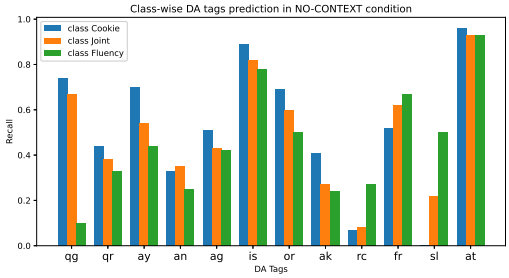


Figure 2: Comparison of class-wise recall of DA tags in the NO CONTEXT condition for all three tasks.

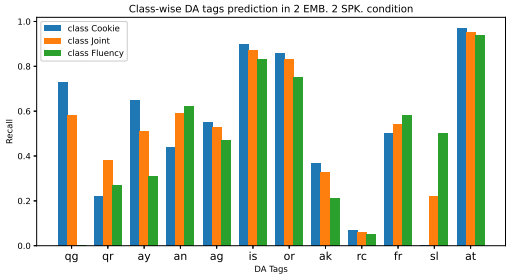


Figure 3: Comparison of class-wise recall of DA tags in the 2 EMB. 2 SPK. condition for all three tasks.

tings. Using only these features, our AD detection model was able to achieve performance comparable to that seen with content-driven, task-specific alternatives (Luz et al., 2020; Di Palo and Parde, 2019). This holds exciting implications for downstream diagnostic or assessment applications, which may be able to leverage these more general features rather than retraining models for new tasks.

To further understand the performance of our DA prediction model and examine the extent to which automated DA tags can support AD detection, we conducted additional error analyses. Specifically, we investigated model outcomes for different DA tags in the baseline (NO CONTEXT) and highest-performing conditions across the *picture description*, *verbal fluency*, and *joint* task settings. We illustrate the findings from these analyses in Figures 2, 3, and 4. Overall, we observed poor recall scores (Figure 2 and 3) for the *Request:Clarification* (*rc*) tag in both models across all tasks. This may be because *rc* utterances can be easily confused with *Question:Reflexive* (*qr*) or *Question:General* (*qg*) tags since they carry similar linguistic and syntactic characteristics (Farzana et al., 2020). Although these question types differ in their intent (*rc* conveys follow-up questions or lack of understanding of specific prior context, whereas *qr* is observed in

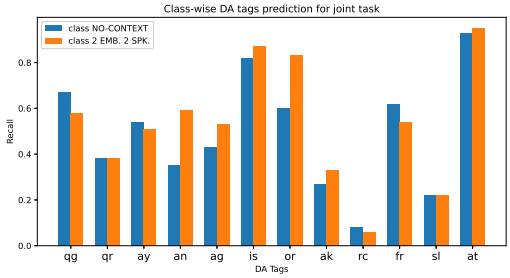


Figure 4: Comparison of class-wise recall of DA tags in NO-CONTEXT vs. 2 EMB. 2 SPK. for the *joint* task.

think-aloud scenarios during which subjects question themselves and *qg* is most commonly seen in out-of-context queries), they ultimately all seek information in some form.

When comparing the NO CONTEXT and highest-performing conditions across all tasks, we also observed that, surprisingly, adding prior context was not always beneficial. In the case of *rc* specifically, performance degrades when the previous speaker tag and utterance embedding are added, primarily in the *verbal fluency* task. The same pattern holds true for *qg* in the *verbal fluency* task, and *qr* in the *picture description* task.

Nonetheless, prior context boosts model performance (or has no negative impact) on a variety of DA classes across tasks. Figure 4 captures the effect of having speaker tags and utterance embeddings both for the current and previous utterance, and shows increases in recall for *Instruction* (*is*), *Other* (*or*), *Acknowledgement* (*ak*), *Answer:No* (*an*), and *Answer:General* (*ag*); we note that these dialog classes in general are associated with utterances that are strongly situated in context. For example, *Instructions* may differ in form depending on how they are received, and *Answer:General* and *Answer:No* are mostly uttered in the context of a question in the previous utterance. Utterances labeled as *Other* are mostly out-of-context statements or non-verbal expressions made by participants, and therefore the inclusion of speaker information is helpful in understanding these utterances.

7 Conclusion

In this paper we studied the extent to which automated analyses of interaction patterns can be leveraged for task-agnostic dementia detection. To do so, we adapted a DA annotation scheme for two different cognitive tasks. We then presented a context-aware DA tagging model that uses transfer learning

from pre-trained sentence embeddings to compute rich representations of utterances, paired with linguistic features (POS sequences) and speaker tags. The model achieved scores of F_1=0.75, F_1=0.85, and F_1=0.82 in a *verbal fluency*, *picture description*, and *joint* task, respectively. We find that although performance is low for some rare-class DAs, adding context information and speaker tags boosts performance in several cases.

To test the utility of interaction patterns as content-free features, we generate features based on these DA tags and other interaction characteristics. We use these to train a random forest classifier for task-agnostic AD detection and achieve strong performance on a joint dataset of *picture description* and *verbal fluency* dialogs. These interpretable interaction features in cognitive health screening tasks show promising performance in AD detection. In the future, we will extend this work to create a more balanced (across tasks and AD vs. non-AD classes) cognitive screening dataset, to further test the boundaries to which these results may generalize. These findings will allow us to outline a guiding principal for designing dialog agents for virtual interviewing in the cognitive health screening domain.

8 Acknowledgment

This work was supported in part by a startup grant from the University of Illinois Chicago. We thank Nitish Dewan for contributing to the data annotation process, and the anonymous reviewers for their helpful feedback.

References

James T. Becker, François Boiler, Oscar L. Lopez, Judith Saxton, and Karen L. McGonigle. 1994. The Natural History of Alzheimer's Disease: Description of Study Cohort and Accuracy of Diagnosis. *Archives of Neurology*, 51(6):585–594.

Harry Bunt. 2006. Dimensions in dialogue act annotation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Flavio Di Palo and Natalie Parde. 2019. Enriching neural models with targeted features for dementia detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 302–308, Florence, Italy. Association for Computational Linguistics.

Christopher Elsey, Paul Drew, Danielle Jones, Daniel Blackburn, Sarah Wakefield, Kirsty Harkness, Annalena Venneri, and Markus Reuber. 2015. Towards diagnostic conversational profiles of patients presenting with dementia or functional memory disorders to memory clinics. *Patient Education and Counseling*, 98(9):1071–1077.

Shahla Farzana, Ashwin Deshpande, and Natalie Parde. 2022. How you say it matters: Measuring the impact of verbal disfluency tags on automated dementia detection. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 37–48, Dublin, Ireland. Association for Computational Linguistics.

Shahla Farzana and Natalie Parde. 2020. Exploring MMSE Score Prediction Using Verbal and Non-Verbal Cues. In *Proc. Interspeech 2020*, pages 2207–2211.

Shahla Farzana, Mina Valizadeh, and Natalie Parde. 2020. Modeling dialogue in conversational cognitive health screening interviews. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1167–1177, Marseille, France. European Language Resources Association.

Heidi Ehernberger Hamilton. 1994. *Conversations with an Alzheimer's Patient: An Interactional Sociolinguistic Study*. Cambridge University Press.

Ryuichiro Higashinaka, Kenji Imamura, Toyomi Meguro, Chiaki Miyazaki, Nozomi Kobayashi, Hiroaki Sugiyama, Toru Hirano, Toshiro Makino, and Yoshihiro Matsuo. 2014. Towards an open-domain conversational system fully based on natural language processing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 928–939, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder Institute of Cognitive Science, Boulder, CO.

Ali Khodabakhsh, Fatih Yesil, Ekrem Guner, and Cenk Demiroglu. 2015. Evaluation of linguistic and prosodic features for detection of alzheimer's disease

in turkish conversational speech. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):9.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Jan-Christoph Klie, Michael Bugert, Beto Boullosa, Richard Eckart de Castilho, and Iryna Gurevych. 2018. The inception platform: Machine-assisted and knowledge-oriented interactive annotation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 5–9. Association for Computational Linguistics. Event Title: The 27th International Conference on Computational Linguistics (COLING 2018).

Changye Li, David Knopman, Weizhe Xu, Trevor Cohen, and Serguei Pakhomov. 2022. GPT-D: Inducing dementia-related linguistic anomalies by deliberate degradation of artificial neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1866–1877, Dublin, Ireland. Association for Computational Linguistics.

Saturnino Luz, Fasih Haider, Sofia de la Fuente, Davida Fromm, and Brian MacWhinney. 2020. Alzheimer's Dementia Recognition Through Spontaneous Speech: The ADReSS Challenge. In *Proc. Interspeech 2020*, pages 2172–2176.

Saturnino Luz, Fasih Haider, Sofia de la Fuente, Davida Fromm, and Brian MacWhinney. 2021. Detecting Cognitive Decline Using Speech Only: The ADReSSo Challenge. In *Proc. Interspeech 2021*, pages 3780–3784.

Jirí Martínek, Christophe Cerisara, Pavel Král, and Ladislav Lenc. 2021. Cross-lingual approaches for task-specific dialogue act recognition. In *AIAI*.

Bahman Mirheidari, Daniel Blackburn, Traci Walker, Markus Reuber, and Heidi Christensen. 2019. Dementia detection using automatic analysis of conversations. *Computer Speech & Language*, 53:65–79.

Shamila Nasreen, Julian Hough, and Matthew Purver. 2021. Rare-class dialogue act tagging for Alzheimer's disease diagnosis. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 290–300, Singapore and Online. Association for Computational Linguistics.

Shamila Nasreen, Matthew Purver, and Julian Hough. 2019. A corpus study on questions, responses and misunderstanding signals in conversations with alzheimer's patients. In *Proceedings of the 23rd Workshop on the Semantics and Pragmatics of Dialogue - Full Papers*, London, United Kingdom. SEMDIAL.

J. B. Orange, Rosemary B. Lubinski, and D. Jeffery Higginbotham. 1996. Conversational repair by individuals with dementia of the alzheimer's type. *Journal of Speech, Language, and Hearing Research*, 39(4):881–895.

Ulla Petti, Simon Baker, and Anna Korhonen. 2020. A systematic literature review of automatic Alzheimer's disease detection from speech and language. *Journal of the American Medical Informatics Association*, 27(11):1784–1797.

Charlene Pope and Boyd H. Davis. 2011. Finding a balance: The carolinas conversation collection. *Corpus Linguistics and Linguistic Theory*, 7(1):143–161.

P.A. Pérez-Toro, S.P. Bayerl, T. Arias-Vergara, J.C. Vásquez-Correa, P. Klumpp, M. Schuster, Elmar Nöth, J.R. Orozco-Arroyave, and K. Riedhammer. 2021. Influence of the Interviewer on the Automatic Assessment of Alzheimer's Disease in the Context of the ADReSSo Challenge. In *Proc. Interspeech 2021*, pages 3785–3789.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Vipul Raheja and Joel Tetreault. 2019. Dialogue Act Classification with Context-Aware Self-Attention. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3727–3733, Minneapolis, Minnesota. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Carole Roth. 2011. Boston diagnostic aphasia examination. In Jeffrey S. Kreutzer, John DeLuca, and Bruce Caplan, editors, *Encyclopedia of Clinical Neuropsychology*, pages 428–430. Springer New York, New York, NY.

Utkarsh Sarawgi, Wazeer Zulfikar, Nouran Soliman, and Pattie Maes. 2020. Multimodal inductive transfer learning for detection of alzheimer's dementia and its severity. In *Proc. Interspeech 2020*, pages 2212–2216.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–374.

Ana Varela Suárez. 2018. The question-answer adjacency pair in dementia discourse. *International Journal of Applied Linguistics*, 28(1):86–101.

Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. 2009. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, page 917–926, New York, NY, USA. Association for Computing Machinery.

Zining Zhu, Jekaterina Novikova, and Frank Rudzicz. 2019. Detecting cognitive impairments by agreeing on interpretations of linguistic features. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1431–1441, Minneapolis, Minnesota. Association for Computational Linguistics.

EDU-AP: Elementary Discourse Unit based Argument Parser

Sougata Saha, Souvik Das, Rohini Srihari
State University of New York at Buffalo
Department of Computer Science and Engineering
{sougatas, souvikda, rohini}@buffalo.edu

Abstract

Neural approaches to end-to-end argument mining (AM) are often formulated as dependency parsing (DP), which relies on token-level sequence labeling and intricate post-processing for extracting argumentative structures from text. Although such methods yield reasonable results, operating solely with tokens increases the possibility of discontinuous and overly segmented structures due to minor inconsistencies in token level predictions. In this paper, we propose EDU-AP, an end-to-end argument parser, that alleviates such problems in dependency-based methods by exploiting the intrinsic relationship between elementary discourse units (EDUs) and argumentative discourse units (ADUs) and operates at both token and EDU level granularity. Further, appropriately using contextual information, along with optimizing a novel objective function during training, EDU-AP achieves significant improvements across all four tasks of AM compared to existing dependency-based methods.

1 Introduction

Considered an integral mode of persuasion, argumentation is prevalent in our daily verbal communication and represents chains of thought patterns and reasoning. An argument constitutes claims and premises, with the claim being the central controversial statement of the argument, and the premise either supporting or attacking the claim by providing the reasoning for the claim (Stab and Gurevych, 2014b). Argument mining (AM) is a recent research field in computational linguistics, that deals with analyzing discourse on the pragmatics level, and finding argumentation structures in natural language texts (Mochales and Moens, 2011; Lippi and Torroni, 2016; Lawrence and Reed, 2019). AM comprises four sub-tasks: (a) *text segmentation:* identifying ADUs from text, by separating argumentative units from non-argumentative units; (b) *component classification:* associating each identified ADU with a tag from a pre-defined labeling scheme (e.g., Claim or Premise); (c) *relation detection:* determining if any relationship exists between pairs of ADUs; (d) *relation classification:* labeling a determined relationship with a tag from a predefined labeling scheme (e.g., attack or support). When performed successfully, AM generally leads to the creation of an *argumentation graph* (AG) (Peldszus and Stede, 2013): a graphical framework for representing arguments, whereby nodes represent claims and premises, and the edges represent diverse relationships (e.g., support, attack) between arguments. Such graphical structures not only help analyze discourse but also aids in the creation of dialogue agents that can leverage the AGs for response generation (Chalaguine and Hunter, 2020; Slonim et al., 2021). Figure 1 illustrates such a graphical relationship, where the premise "biking reduces stress and releases endorphins", supports the claim "biking is good".

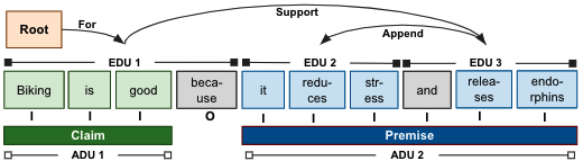


Figure 1: Dependency tree for the argument "Biking is good because it reduces stress, and releases endorphins".

With an increased interest in engendering purposeful and persuasive conversational agents, the need for argument parsers that can automatically and effectively extract, parse and relate argumentative components end-to-end from natural language text is on the rise. In this paper, we address this need by proposing a robust end-to-end argument parser that formulates AM as a dependency parsing (DP) problem. Unlike prior research in DP based argument mining approaches, we exploit the innate relationship between EDUs and ADUs in multi-task learning (MTL) framework and achieve competitive results. We further improve upon our

Proceedings of the SIGdial 2022 Conference, pages 183–192
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

results by utilizing appropriate contextual information and optimizing a novel objective function during training.

2 Related Work

Significant advancements have been made in computational model for AM in recent years. Stab and Gurevych (2014b) implemented a feature engineering based pipelined approach for performing all four sub-tasks of AM, on the Persuasive Essays (PE) corpus (Stab and Gurevych, 2014a), which was further improved by the Integer Linear Programming (ILP) based approach proposed by Persing and Ng (2016). Stab and Gurevych (2017) introduced a larger version of the PE corpus and implemented an ILP constrained pipelined approach for AM. Mirko et al. (2020) improved upon the pipelined approach for AM introduced by Nguyen and Litman (2018), and further implemented a novel graph construction process to create argument graphs. Recently, Bao et al. (2021) proposed a neural transition-based model for component classification and relationship detection, which incrementally builds an argumentation graph by generating a sequence of actions, and can handle both tree and non-tree argumentation structures.

Eger et al. (2017) formulated the tasks of AM as a token level DP, and achieved state-of-the-art performance on the PE dataset, using a neural dependency parser. Inspired by the success of incorporating biaffine classifiers for semantic DP (Dozat and Manning, 2016, 2018), Ye and Teufel (2021) further improved the DP based approach by using biaffine layers, and leveraged pre-trained BERT (Devlin et al., 2018) for richer argument representations. Instead of operating at a word level, Morio et al. (2020) experimented with proposition level AM and used a joint learning framework for jointly performing the tasks of component classification, relation detection and classification.

Considerable work has also been done in trying to establish relationships between ADUs and EDUs. Peldszus (2015); Peldszus and Stede (2016); Musi et al. (2018); Hewett et al. (2019) studied the mapping from discourse structure from Rhetorical Structure Theory (RST) to argumentation structures and showed that discourse relations from RST often correlate with argumentative relations.

3 Proposed Approach

Our work is inspired by the token level dependency parser proposed by Ye and Teufel (2021), and the proposition level parser proposed by Morio et al. (2020). However, unlike previous works, our formulation of dependency representation for arguments unifies all sub-tasks of AM under an EDU level framework and exploits the relationship between EDUs and ADUs. We factorize the sub-tasks of AM as different prediction tasks and train end-to-end in a multi-task learning (MTL) framework. We implement a hierarchical encoding scheme, which enables the use of a larger context, and train using a modified loss function for increasing performance.

3.1 Dependency Representation for Arguments

As illustrated in Figure 1, we structure arguments as a combination of EDUs and define types of relationships that could potentially hold between EDUs. We further enrich each EDU token with segment boundaries, that enable the re-construction of ADU from the EDUs. We list the properties of our dependency representation below:

- An EDU can either partially or fully overlap with an ADU and each token in an EDU is labeled as argumentative or non-argumentative, using the IO tagging scheme. For example in Figure 1, EDU 1 partially overlaps with ADU 1, as the token “because" is tagged as “O", whereas the EDUs 2 and 3 fully overlap with ADU 2, which is indicated by all the tokens in the EDUs labeled as “I".
- Each EDU can only belong to 1 of 4 classes $\in$ [major claim (MC), claim (C), premise (P), non-argument (NA)]. Consecutive EDUs which belong to the same class can be combined using *Append* relationship to yield an ADU. For example in Figure 1, EDU 2 and 3 can be combined using the *Append* relationship to construct ADU 2.
- Claim and premise EDUs can be related using “Support" (Sup) or “Attack" (Att) relationships, with the relationship originating from the last EDU of a claim to the last EDU of a premise. EDUs comprising premises can be related using the “Support" relationship, with the relationship originating from the last EDU of the supported premise to the last EDU of the supporting premise.

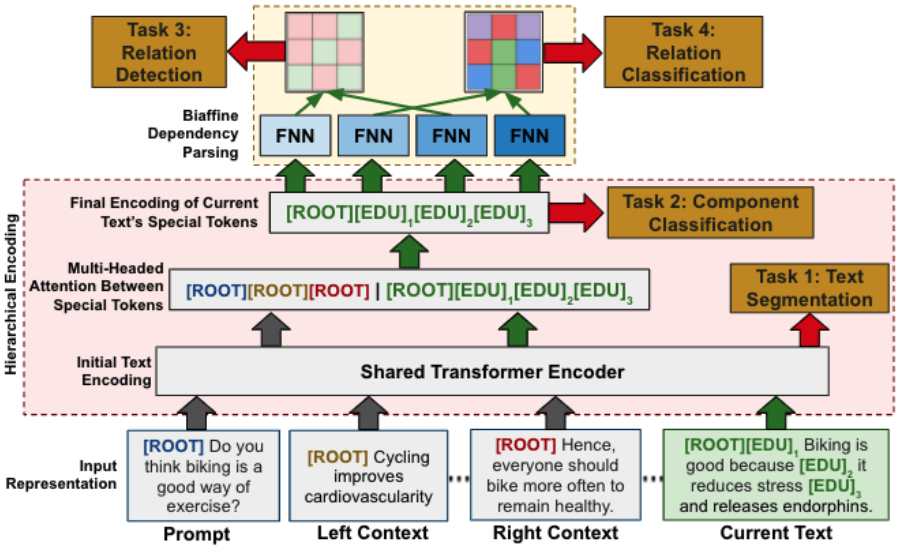


Figure 2: End-to-End Model Architecture.

- A pseudo-token “ROOT" is added to the beginning of each argument, which represents the topic or gist of the entire argument. “ROOT" is always acting as a parent to the highest-level component(s).

- Each claim is parented by the “ROOT", and related using “For" (For) or “Against" (Agn) relationship, signifying the stance of the claim concerning the topic of discussion. In case of the presence of a MC, the “ROOT" parents the MC using a “default" (Def) relationship, which in turn parents the claims using “For" or “Against" relationships.

- An EDU can contain zero or more parents, and the relationships are acyclic.

In contrast to Ye and Teufel (2021), parsing arguments at an EDU level reduces complexity and simplifies all sub-tasks of AM, which we hypothesize should lead to better results. Similar to Morio et al. (2020), we implement DP only for the relationship detection and classification sub-tasks and further incorporate separate classifiers for text segmentation and component classification.

3.2 Multi-Task Learning (MTL) for AM

We train our argument parser in a MTL framework, where all the AM sub-tasks share a common encoding representation, followed by task-specific layers. Figure 2 illustrates our architecture in detail [1].

[1]EDU-AP codebase: https://github.com/sougata-ub/edu-ap.

3.2.1 Model Input Representation

We segment input text into EDUs using the Bi-LSTM-CRF based discourse segmenter by Wang et al. (2018) and add a special *[EDU]* token to the start of each span. The *[EDU]* token acts as a delimiter between EDU spans, and also represents the meaning of the corresponding EDU. Further, a *[ROOT]* token is added to the start of each input, which represents the meaning of the entire text.

3.2.2 Hierarchical Encoding

Depicted in Figure 2, we implement a hierarchical encoder, where we sequentially encode the current paragraph input and context tokens using a shared transformer encoder, and perform multi-headed attention (MHA) between the current input special tokens and the concatenated contextual *[ROOT]* tokens. Equations 1 to 4 defines the encoding process, where $\mathrm{E_{curr}}$ and $\mathrm{E_{ctx}}$ are the encoded representations of the current and context inputs $\mathrm{S_{curr}}$ and $\mathrm{S_{ctx}}$. The representations of the current turn and context special tokens $\mathrm{E^{\mathit{I}}_{curr}}$ and $\mathrm{E^{\mathit{I}}_{ctx}}$ are selected (using Get) from $\mathrm{E_{curr}}$ and $\mathrm{E_{ctx}}$ respectively. The final representation $\mathrm{E^{MHA}_{curr}}$ of the current turn's special tokens is obtained by sum pooling $\mathrm{E^{\mathit{I}}_{curr}}$ and the MHA output followed by a dropout layer.

$$\mathrm{E_{curr}}=\mathrm{Encode}(\mathrm{S_{curr}});\ \mathrm{Get}(\mathrm{X}, \mathit{idx})=\mathrm{X}[\mathit{idx}, :] \quad (1)$$

$$\mathrm{E^{\mathit{I}}_{curr}}=\mathrm{Get}(\mathrm{E_{curr}}, \mathit{idx}_{\mathrm{ROOT,EDU}}) \quad (2)$$

$$\mathrm{E^{\mathit{I}}_{ctx}}=\mathrm{Get}(\mathrm{Encode}(\mathrm{S_{ctx}}), \mathit{idx}_{\mathrm{ROOT}}) \quad (3)$$

$$\mathrm{E^{MHA}_{curr}}=\mathrm{E^{\mathit{I}}_{curr}}+\mathrm{Dropout}(\mathrm{MHA}(\mathrm{E^{\mathit{I}}_{curr}}, \mathrm{E^{\mathit{I}}_{ctx}})) \quad (4)$$

Such a formulation not only encourages the special tokens to better encode its representative span but also ameliorates the length bottleneck (Joshi et al., 2020) in transformer architectures by reducing the sequence length. Thus, enabling the use of a larger context compared to token level parsing.

3.2.3 Task Specific Prediction

Post encoding, we incorporate task-specific layers to perform the final prediction for each task. Depicted in Equations 5 and 6, we use single-layered feed-forward neural networks (FNNs) as the final layer for both text segmentation and component classification. For text segmentation, we use the initial token level encoding ($\mathrm{E_{curr}}$) of the current text as input, whereas for component classification, the final encoding of the current text *[EDU]* tokens ($\mathrm{E^{MHA}_{curr}}$) are used as inputs.

$$sc^{span}=\mathrm{FNN}(\mathrm{E_{curr}});\ sc^{typ}=\mathrm{FNN}(\mathrm{E^{MHA}_{curr}}) \quad (5)$$

$$y^{'span}=\{sc^{span} \geq 0\};\ y^{'typ}=\mathrm{argmax}\ sc^{typ} \quad (6)$$

Biaffine classifiers are generalizations of linear classifiers, which include multiplicative interactions between two vectors. Since relation detection and relation classification require performing inference between argument pairs, we implement biaffine dependency parsing (DP) for both the tasks. Using FNNs, the current text *[ROOT]* and *[EDU]* encodings $\mathrm{E^{MHA}_{curr}}$ are split into two parts with reduced hidden size–a head (parent) and a dependent (child) representation, which in turn are passed through a biaffine classifier (Biaf) for predicting edges and labels between EDUs. Equations 7 to 11 details our biaffine DP formulation, where $\mathrm{H^{e\_p}}$ and $\mathrm{H^{e\_c}}$ denotes the parent and child representations for relation detection, and $\mathrm{H^{l\_p}}$ and $\mathrm{H^{l\_c}}$ denotes the parent and child representations for relation classification. sc^e and sc^l contains the output logits from the biaffine layers, where $sc^e_{i,j}$ and $sc^l_{i,j}$ denotes the logits between the i^{th} and j^{th} EDU for relation detection and classification respectively.

$$\mathrm{Biaf}(x,y)=x^T\mathrm{U}y+\mathrm{W}(x\oplus y)+b \quad (7)$$

$$\mathrm{H^{e\_p}}=\mathrm{FNN}(\mathrm{E^{MHA}_{curr}});\ \mathrm{H^{e\_c}}=\mathrm{FNN}(\mathrm{E^{MHA}_{curr}}) \quad (8)$$

$$\mathrm{H^{l\_p}}=\mathrm{FNN}(\mathrm{E^{MHA}_{curr}});\ \mathrm{H^{l\_c}}=\mathrm{FNN}(\mathrm{E^{MHA}_{curr}}) \quad (9)$$

$$sc^e=\mathrm{Biaf}(\mathrm{H^{e\_p}},\mathrm{H^{e\_c}});\ sc^l=\mathrm{Biaf}(\mathrm{H^{l\_p}},\mathrm{H^{l\_c}}) \quad (10)$$

$$y^{'e}_{i,j}=\{sc^e_{i,j} \geq 0\};\ y^{'l}_{i,j}=\mathrm{argmax}\ sc^l_{i,j} \quad (11)$$

3.2.4 Modified Objective Function

Depicted in Equation 16, we train the model end-to-end by minimizing the aggregated interpolated loss across all four sub-tasks, with an interpolation factor λ. The sub-tasks of text segmentation, component classification and relation classification are trained by minimizing the cross entropy (CE) losses $\mathcal{L}^{span}$, $\mathcal{L}^{typ}_i$, $\mathcal{L}^l_{i,j}$ respectively in Equations 14 and 15, whereas relation prediction is trained by minimizing the binary cross entropy (BCE) loss $\mathcal{L}^e_{i,j}$ (Equation 13).

We further add an extra penalty term δ with an interpolation factor of β to the BCE loss in Equation 13, to increase the recall of predicting relationships between EDUs. Exploiting the symmetry of the final score matrix (logits) in biaffine classifiers, as depicted in Equations 12 and 13, the penalty term for the relationship from the i^{th} to j^{th} EDU is set to be dependent on the logit of its reverse: j^{th} to i^{th}. This results in the loss function being penalized most in case a relationship and its conjugate reverse are both predicted to be present or absent, and least if either is predicted to exist. We hypothesize that such a penalty should increase the relation detection recall, while minimally impacting the precision.

$$\delta(y,\tilde{y})=y\log(1\text{-}\sigma(\tilde{y}))+(1\text{-}y)\log\sigma(\tilde{y}) \quad (12)$$

$$\mathcal{L}^e_{i,j}=\beta\ \mathrm{BCE}(\mathrm{y^e_{i,j}},\mathrm{sc^e_{i,j}})\text{-}(1\text{-}\beta)\delta(y^e_{i,j},sc^e_{j,i}) \quad (13)$$

$$\mathcal{L}^l_{i,j}=\mathrm{CE}(y^l_{i,j},sc^l_{i,j});\ \mathcal{L}^{typ}_i=\mathrm{CE}(y^{typ}_i,sc^{typ}_i) \quad (14)$$

$$\mathcal{L}^{span}=\mathrm{CE}(y^{span},sc^{span}) \quad (15)$$

$$\mathcal{L}=\lambda\mathcal{L}^e+(1\text{ - }\lambda)(\mathcal{L}^l+\mathcal{L}^{span}+\mathcal{L}^{typ}) \quad (16)$$

3.2.5 Post Processing and Graph Construction

During inference, we perform a few post-processing steps to constrain the model output. For relationship prediction, we discard self references and cyclic relationships, and further restrict the argument structures to conform to the ones defined by Stab and Gurevych (2017), i.e premise→premise, premise→claim, claim→major claim/claim→ROOT and major claim→ROOT.

For generating an argument graph, we extract ADUs by concatenating consecutive argumentative EDUs that are predicted to be connected by an "Append" relationship and remove non-argumentative tokens, using the text segmentation prediction. To yield contiguous arguments and prevent unnatural segmentation, we ensure the label of each token within an appended ADU confirms with its neighbours, and re-label to the majority class of its neighbours if needed. Next, we label each ADU by assigning the majority label of the constituent EDUs predicted by the component classifier. Fi-

nally, a graph is formed by connecting the labeled ADUs with the relationships predicted by the relation classifier, only if the relation detector predicts its existence.

4 Experiments

4.1 Dataset

We use the benchmark persuasive essays (PE) dataset by Stab and Gurevych (2017), for all our experiments. This dataset comprises 402 persuasive essays: 322 for training and 80 for testing, randomly selected from an online forum. Barring non-arguments, there are three kinds of argumentative components in the dataset, along with four types of relationships that can hold between the components: *(i) Major claim*: the main claim by an author, which informs their stance. *(ii) Claim*: a statement that is either *For* or *Against* one or more major claims. *(iii) Premise*: a statement that provides evidence to a claim or another premise by a *Support* or *Attack* relationship. Please refer Stab and Gurevych (2017) for a detailed dataset statistics.

4.2 Experiment Setup

We use Roberta (base) (Liu et al., 2019) as the base encoder, and increase its embedding layer to accommodate the special tokens. Two layers comprising four attention heads are used for MHA, where the MHA result in each layer is sum pooled with the residual output while applying dropout with 0.1 probability to the MHA result. The hidden size of the FNNs in the biaffine layer is set to 600. An interpolation factor λ of 0.95 is used for aggregating the losses, and the factor β in the modified BCE loss is set to 0.85. All models are trained with a learning rate of 1e-5 for 15 epochs and optimised using AdamW (Loshchilov and Hutter, 2017), with early stopping if the validation loss doesn't reduce for 2 epochs. We repeat each experiment five times and report the average across all runs.

4.3 Competing Model

We recreate the state-of-the-art BiPAM parser by Ye and Teufel (2021) as an external baseline and compare it against our proposed method. BiPAM implements token level dependency parsing for end-to-end argument mining and had achieved significant improvements over LSTM-Parser and LSTM-ER reported in Eger et al. (2017).

4.4 Evaluation Metrics

All tasks are evaluated using the F1 score. Similar to Persing and Ng (2016), approximate and exact overlap is computed between the golden and predicted ADUs, where a predicted ADU is classified as approximate overlap if at least 50% of its tokens match with the golden ADU, and is classified as exact overlap if all the tokens match with the golden ADU. Each task is evaluated for both the approximate and exact overlapping ADU spans.

4.5 Results and Analysis

We share the experimental results for the sub-tasks of text segmentation and component classification in Table 1, and the sub-tasks of relation detection and relation classification in Table 2. In each table, we calculate and report the F1 score for both approximate and exact overlapping ADUs (approx/exact). We treat EDU-AP–our non-contextual implementation without the δ loss penalty as the internal baseline, and underline the best performing model for each task, in comparison to this baseline. We also compare the results obtained from the BiPAM parser and highlight the best performing result in comparison to this baseline in bold.

As indicated by the top section of Table 1, using a mixture of EDUs and tokens, our baseline EDU-AP outperforms token level BiPAM for text segmentation by a significant margin (61.8/53.2 compared to 38.8/25.6). We reason that operating solely with tokens, BiPAM increases the chances of locally erroneous predictions, yielding discontinuities in ADU spans. Whereas EDU-AP minimizes this by globally identifying EDUs which should be combined as an ADU using the "Append" relationship, and further locally eliminating non-argumentative tokens from each EDU. This is further corroborated by the fact that the average ratio between predicted ADU spans and golden ADU spans per paragraph is 1.1 for the EDU-AP, in comparison to an average ratio of 2.2 in the BiPAM parser, signifying a greater number of short spans predicted by BiPAM.

For component classification (Table 1), EDU-AP outperforms BiPAM across all classes (Major Claim: MC, Claim: C, Premise: P and Non Argument: NA) for the approximately matched ADU spans. However, for the exact matching spans, BiPAM mostly performs better than EDU-AP.

For both the relation detection and classification tasks in the top section of Table 2, EDU-AP largely

	ADU Span	Component Classification			
Model	**F1**	**MC-F1**	**C-F1**	**P-F1**	**NA-F1**
BiPAM †	38.8 / 25.6	81.9 / 87.8	67.8 / **77.0**	88.8 / 88.0	92.5 / **98.3**
EDU-AP *	**61.8** / 53.2	86.2 / 86.8	68.9 / 70.8	90.3 / 90.7	96.4 / 97.6
EDU-AP + prompt	61.4 / 52.5	84.9 / 85.2	69.2 / 71.7	90.5 / 91.1	96.6 / 97.7
EDU-AP + left	61.5 / 51.9	84.2 / 83.9	64.8 / 67.4	89.9 / 90.1	96.2 / 97.6
EDU-AP + all	60.5 / 50.8	83.6 / 84.1	68.2 / 70.2	90.5 / 90.8	95.9 / 97.7
EDU-AP + δ	61.4 / **53.5**	85.7 / 87.3	73.5 / 75.1	91.2 / 91.4	97.2 / 98.1
EDU-AP + δ + prompt	61.2 / 53.2	**87.8** / 88.1	**74.2** / 76.1	**91.9** / **92.2**	**97.3** / 97.9
EDU-AP + δ + left	60.0 / 51.5	87.0 / **88.4**	71.2 / 73.0	91.5 / 91.6	96.6 / 97.8
EDU-AP + δ + all	59.9 / 51.6	86.3 / 86.6	71.9 / 73.7	91.3 / 91.6	96.0 / 97.4

Table 1: Results for the sub-tasks Text Segmentation and Component Classification (**M**ajor **C**laim, **C**laim, **P**remise, **N**on **A**rgument), for both approximate and exact overlapping spans (approx/exact). †and * denotes external and internal baselines respectively.

	Relation Detection	Relation Classification				
Model	**F1**	**Agn-F1**	**Att-F1**	**Def-F1**	**For-F1**	**Sup-F1**
BiPAM †	37.1 / 13.2	16.1 / 5.0	0.0 / 0.0	61.1 / 30.9	51.3 / 24.4	26.1 / 5.7
EDU-AP *	57.0 / 47.3	56.5 / 46.9	31.9 / 12.4	85.5 / 72.9	74.1 / 65.6	57.7 / 47.5
EDU-AP + prompt	62.0 / 51.6	58.5 / 47.7	35.3 / 14.1	85.1 / 71.2	81.9 / 73.2	68.8 / 57.3
EDU-AP + left	57.2 / 46.8	55.9 / 48.3	29.8 / 14.1	86.5 / 71.0	75.4 / 66.9	59.6 / 48.2
EDU-AP + all	57.8 / 47.4	56.4 / 47.2	31.1 / 15.6	86.5 / 71.1	75.9 / 66.9	59.5 / 48.8
EDU-AP + δ	62.6 / 53.8	64.6 / **57.0**	38.8 / 18.7	86.3 / **77.8**	80.3 / 73.2	69.1 / 58.5
EDU-AP + δ + prompt	**64.9** / **55.0**	64.4 / 56.7	**41.1** / **22.6**	**88.4** / 77.0	**82.5** / **74.1**	**74.3** / **62.9**
EDU-AP + δ + left	62.3 / 52.2	59.3 / 50.9	39.0 / 17.3	86.1 / 75.9	77.7 / 69.9	66.8 / 54.8
EDU-AP + δ + all	62.8 / 53.0	64.4 / 57.0	37.7 / 18.1	87.8 / 76.7	81.1 / 72.5	69.3 / 58.0

Table 2: Results for the sub-tasks Relation Detection and Relation Classification (**Ag**a**in**st, **Att**ack, **Def**ault, **For**, **Sup**port), for both approximate and exact overlapping spans (approx/exact). †and * denotes external and internal baselines respectively.

outperforms BiPAM. We reason that since both the tasks demand inference over pairs of argumentative sentences, using our formulation of operating at an EDU level and using representative *[EDU]* tokens better represents and encodes arguments, thus providing better context during scoring, compared to operating at the individual token level. Further, unlike BiPAM, incorporating task-specific layers encourages learning task-specific parameters, which enhances the model's performance.

4.6 Ablation Study

We further perform an ablation study, to determine the effect of adding the δ penalty and utilizing context in all sub-tasks. The bottom section in both Tables 1 and 2 includes the ablation results. We experiment with combinations of adding an essay's prompt as context (+prompt), the prompt along with all the past paragraphs (+left), the prompt along with all other paragraphs (+all), and the loss penalty (+δ).

Overall, we observe that although baseline EDU-AP performs better than BiPAM, incorporating the δ penalty increases the model's efficacy for most sub-tasks, which is further boosted by adding the essay's prompt (+prompt) as context. Most significant improvements are observed for relation detection and classification (Table 2 bottom section), which is intuitively justified, as establishing relationships (like *For*/*Against*) not only require knowledge and understanding of the main theme of discussion, but also cognizance of the established stance towards the topic from prior paragraphs. Table 3 further illustrates the impact of the δ penalty on the precision and recall scores for relation detection. As previously hypothesized, we observe

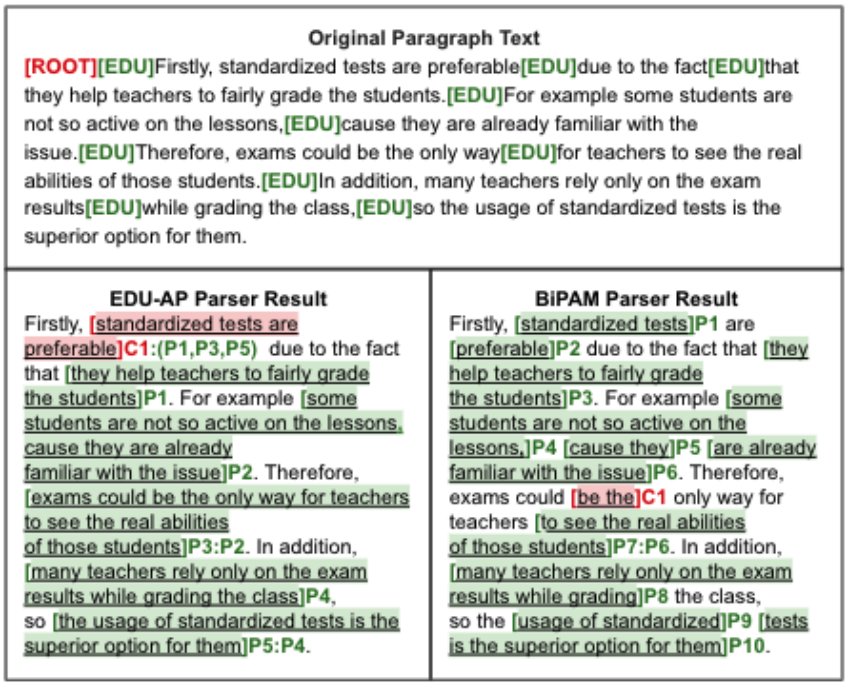


Figure 3: Comparison of a parsed paragraph from test set using the EDU-AP (left) and BiPAM (right) parsers.

that incorporating the δ penalty almost always improves recall, while also boosting the precision in some cases.

Model	F1	Precision	Recall
BiPAM †	37.1 / 13.2	32.3 / 13.9	45.4 / 13.1
EDU-AP *	57.0 / 47.3	71.6 / 66.1	48.1 / 37.6
+ prompt	62.0 / 51.6	68.0 / 62.1	57.1 / 44.2
+ left	57.2 / 46.8	68.6 / 62.4	49.4 / 37.7
+ all	57.8 / 47.4	70.2 / 64.2	50.2 / 38.5
+ δ	62.6 / 53.8	69.2 / 64.5	57.5 / 46.4
+ δ + prompt	**64.9 / 55.0**	67.8 / 62.5	**62.3 / 49.2**
+ δ + left	62.3 / 52.2	**72.8 / 67.7**	55.0 / 42.9
+ δ + all	62.8 / 53.0	69.0 / 63.7	58.1 / 45.7

Table 3: Comparison of F1, Precision and Recall for the Relation Detection subtask, for both approximate and exact overlapping spans (approx/exact). † and * denotes external and internal baselines respectively.

We also observe that text segmentation largely remains unaffected by the addition of context and δ penalty (Table 1 bottom section), which is justified by the nature of the task, which does not depend much on external context, and relies more on linguistic features.

The nature of argumentation is such that the label of the components can be ascertained with a fair probability, from the relationships that exist between components. For example, as described in sub-section 4.1, only a claim can be the child node in a *For*/*Against* relationship, and premises can only be a part of *Support*/*Attack* relationships. Although the δ penalty is not directly applied to component classification, we still observe an increase in performance for classifying components, with the addition of the δ penalty and context (Table 1 bottom section). We attribute this to our multi-task learning framework, which enables learning joint representations that benefit all sub-tasks.

It is also interesting to note that for all the sub-tasks, adding more context (+left, +all) does not always yield superior results, whereas just adding the prompt (+prompt) of the essay yields better results. We attribute this to the fairly small size of the corpus used in the experiments, which does not provide many data points for learning complex interactions from context.

4.7 Discussion

Our results indicate that parsing text at a combination of EDU and token level yields better results, compared to bare token level DP, and can be further improved by appropriately penalizing the loss function, and incorporating contextual information. Figure 3 illustrates and compares a parsed example of a paragraph from the test set, using both the EDU-AP and BiPAM parsers. We underline and enclose the predicted argumentative spans by the models in square brackets, and assign a unique identifier to each component (C1, C2, P1, etc.). Predicted claims are highlighted in red, and their unique iden-

tifier starts with 'C', whereas premises are highlighted in green, with their identifier starting with 'P'. Predicted relationships between components are separated using a colon. Example P5:P4 signifying support relationship from P4 to P5, or C1:(P1, P3, P5) signifying support relationship from P1, P3 and P5 to C1.

We observe that EDU-AP can correctly identify ADU spans by merging EDUs using the "Append" relationship and further eliminating non-argumentative tokens. BiPAM on the other hand yields more fragmented and discontinuous spans. The average ratio of predicted and golden ADU spans per paragraph in the test set is 1.1 for the EDU-AP in comparison to 2.2 for BiPAM parser, signifying a greater number of shorter spans predicted by BiPAM. For example, the span C1 in EDU-AP parsed output (which matches with the golden span) is split into two spans: P1 and P2 by BiPAM. We further observe that EDU-AP is not only able to correctly label the identified ADUs as claim and premise but also able to correctly predict support relationships between the ADUs. Compared to that, BiPAM is not able to correctly identify the claim of the paragraph and fails to predict any relationships between the arguments.

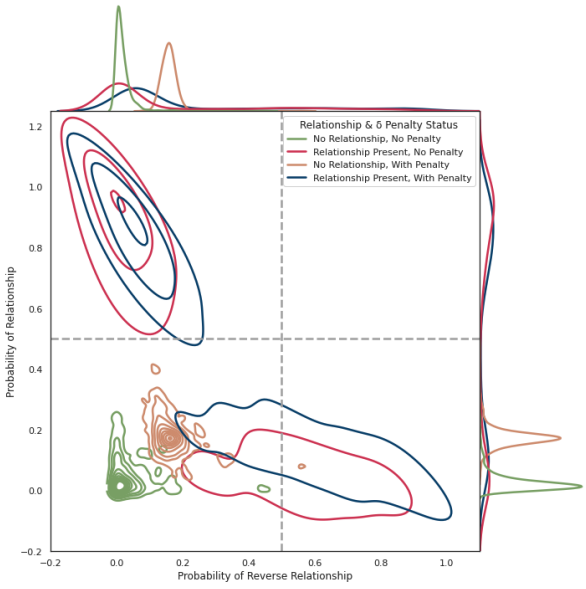


Figure 4: KDE plots comparing the effect of δ penalty on the distribution of relationship probability and its reverse for relationship detection. The dotted line denotes the probability threshold used for the experiments.

To understand the effect of the δ penalty on relation detection, we combine results from all experiments and plot the kernel density of probabilities of predicted relationships and its conjugate reverse relationship in Figure 4. We observe that as expected, overall the model assigns lower probabilities when no relationship exists between a pair of argument components and asymmetrically higher probabilities when a relationship exists, signifying a unidirectional relationship. Adding the δ penalty has the effect of shifting the probability distributions towards more symmetry (i.e for pairs of components, the difference of predicted probability for both directions is reduced), resulting in a recall seeking behaviour.

Although EDU-AP outperforms all baselines, it still fails to attain the human upper bound performance measured by Stab and Gurevych (2017) on the PE corpus. Further, trained only on monological essay data, EDU-AP can't be used for parsing other forms of discourse like dialogue, which we seek to address in our next research steps.

5 Conclusion

In this paper, we present EDU-AP, an end-to-end dependency parsing based argument parser for parsing arguments from molonogical text. Exploiting the innate relationship between EDUs and ADUs, along with the appropriate use of context, and a hierarchical encoding scheme, EDU-AP is trained end-to-end in a multi-task learning setting by minimizing a novel loss function. EDU-AP's efficacy is demonstrated by its superior experimental and ablation results, in comparison to strong internal and external baselines. We believe, with minor adjustments EDU-AP can be purposed for parsing arguments from dialogues.

References

Jianzhu Bao, Chuang Fan, Jipeng Wu, Yixue Dang, Jiachen Du, and Ruifeng Xu. 2021. A neural transition-based model for argumentation mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6354–6364, Online. Association for Computational Linguistics.

Lisa Andreevna Chalaguine and Anthony Hunter. 2020. A persuasive chatbot using a crowd-sourced argument graph and concerns. In *COMMA*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Timothy Dozat and Christopher D. Manning. 2016.

Deep biaffine attention for neural dependency parsing.

Timothy Dozat and Christopher D. Manning. 2018. Simpler but more accurate semantic dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 484–490, Melbourne, Australia. Association for Computational Linguistics.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11–22, Vancouver, Canada. Association for Computational Linguistics.

Freya Hewett, Roshan Prakash Rane, Nina Harlacher, and Manfred Stede. 2019. The utility of discourse parsing features for predicting argumentation structure. In *Proceedings of the 6th Workshop on Argument Mining*, pages 98–103, Florence, Italy. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

John Lawrence and Chris Reed. 2019. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, 16(2).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

LENZ Mirko, Premtim Sahitaj, Sean Kallenberg, Christopher Coors, Lorik Dumani, Ralf Schenkel, and Ralph Bergmann. 2020. Towards an argument mining pipeline transforming texts to argument graphs. *Computational Models of Argument: Proceedings of COMMA 2020*, 326:263.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Gaku Morio, Hiroaki Ozaki, Terufumi Morishita, Yuta Koreeda, and Kohsuke Yanai. 2020. Towards better non-tree argument mining: Proposition-level biaffine parsing with task-specific parameterization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3259–3266, Online. Association for Computational Linguistics.

Elena Musi, Manfred Stede, Leonard Kriese, Smaranda Muresan, and Andrea Rocci. 2018. A multi-layer annotated corpus of argumentative text: From argument schemes to discourse relations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Huy V. Nguyen and Diane J. Litman. 2018. Argument mining for improving the automated scoring of persuasive essays. In *AAAI*.

Andreas Peldszus. 2015. An annotated corpus of argumentative microtexts.

Andreas Peldszus and Manfred Stede. 2013. From argument diagrams to argumentation mining in texts: A survey. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 7(1):1–31.

Andreas Peldszus and Manfred Stede. 2016. Rhetorical structure and argumentation structure in monologue text. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 103–112, Berlin, Germany. Association for Computational Linguistics.

Isaac Persing and Vincent Ng. 2016. End-to-end argumentation mining in student essays. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1384–1394, San Diego, California. Association for Computational Linguistics.

Noam Slonim, Yonatan Bilu, Carlos Alzate, Roy Bar-Haim, Ben Bogin, Francesca Bonin, Leshem Choshen, Edo Cohen-Karlik, Lena Dankin, Lilach Edelstein, et al. 2021. An autonomous debating system. *Nature*, 591(7850):379–384.

Christian Stab and Iryna Gurevych. 2014a. Annotating argument components and relations in persuasive essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1501–1510, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christian Stab and Iryna Gurevych. 2014b. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 962–967, Brussels, Belgium. Association for Computational Linguistics.

Yuxiao Ye and Simone Teufel. 2021. End-to-end argument mining as biaffine dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 669–678, Online. Association for Computational Linguistics.

Using Transition Duration to Improve Turn-taking in Conversational Agents

Charles Threlkeld and **Muhammad Umair** and **JP de Ruiter**
Tufts University
{charles.threlkeld, muhammad.umair, jp.deruiter}@tufts.edu

Abstract

Smooth turn-taking is an important aspect of natural conversation that allows interlocutors to maintain adequate mutual comprehensibility. In human communication, the timing between utterances is normatively constrained, and deviations convey socially relevant paralinguistic information. However, for spoken dialogue systems, smooth turn-taking continues to be a challenge. This motivates the need for spoken dialogue systems to employ a robust model of turn-taking to ensure that messages are exchanged smoothly and without transmitting unintended paralinguistic information. In this paper, we examine dialogue data from natural human interaction to develop an evidence-based model for turn-timing in spoken dialogue systems. First, we use timing between turns to develop two models of turn-taking: a speaker-agnostic model and a speaker-sensitive model. From the latter model, we derive the propensity of listeners to take the next turn given TRP duration. Finally, we outline how this measure may be incorporated into a spoken dialogue system to improve the naturalness of conversation.

1 Introduction

Turn-taking is an important component of many spoken dialogue systems and involves (a) detecting or predicting the end of a turn and (b) accurate timing of the initiation of speech production (Michael, 2020; Kennington et al., 2020). Smooth turn-taking continues to be a challenge for spoken dialogue systems that aim to engage in natural conversation (Hara et al., 2019). Traditionally, most spoken dialogue systems process Inter-Pausal Units (IPUs), which are speech units surrounded by arbitrary fixed length silence thresholds (Skantze, 2021). These pauses of arbitrary duration cause a stilted, unnatural conversation style.

Other systems use incremental approaches, processing smaller units of speech at a time. For example, the incremental dialogue system proposed by Skantze and Schlangen (2009) operates on Incremental Units (IUs) that are processed by Incremental Modules (IMs). These modules may include action, turn, or dialogue management—each of which influences turn-planning and end of turn detection. Although such systems initiate the production of speech when a silence is detected, they do so based on pitch or semantic completeness (Skantze and Hjalmarsson, 2010). Machine learning models of turn-taking operate on previously detected multimodal cues (Bohus and Horvitz, 2010; Skantze, 2021).

The turn-taking techniques used in traditional spoken dialogue systems, such as predicting turn-ends in a time window (Lala et al., 2019), are not fully grounded in current theory of human turn-taking. In natural conversation, people tend to minimize gaps and overlaps while also following the one "one speaker at a time" rule (Sacks et al., 1974). This means that when a turn ends, another speaker may start speaking. Speakers also use the duration of silences to convey social information (de Ruiter, 2019). For example, long and short gaps may communicate hesitance or impatience. Additionally, interlocutors use turn-taking cues (e.g., lexico-syntactic, pragmatic, prosodic etc.) to predict the end of turns and plan responses (Levinson and Torreira, 2015; Liddicoat, 2004). In contrast, turn-taking techniques typically do not explicitly identify points where floor change may occur (Hara et al., 2019), normatively time the duration of silences, or predict turn-ends independent of the occurrence of specific events (e.g, silences) (Skantze and Hjalmarsson, 2010). For spoken dialogue systems, this leads to mistimed responses and a decrease in human engagement (Zhao et al., 2018).

In this paper, we propose an evidence-based model for *when* speech may be produced to facilitate smooth turn-taking, based on the turn-taking model proposed by Sacks et al. (1974). In this model, a speaker's turn consists of one or more

Proceedings of the SIGdial 2022 Conference, pages 193–203
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Turn Construction Unit (TCU), which encompasses sentential, clausal, phrasal, and lexical constructions. Between each TCU are Transition Relevance Places (TRPs), where the current turn may be completed and a floor change may occur (Selting, 2000). We further divide TRPs into continuations (TRPs where the current speaker continues) and switches (TRPs where a speaker-transition occurs). Additionally, in our operationalization, each TRP has a duration—the time between when the previous TCU is complete but before the next TCU begins. We use TCU-level data from transcriptions of the Switchboard corpus (Godfrey and Holliman, 1993) to develop two models of turn-taking based on the duration of TRPs: a speaker-agnostic model and a speaker-sensitive model. Next, we develop an evidence-based function for the propensity of floor-transfer as a function of time after the end of a TCU. Finally, we outline a proposal for implementing this propensity function into the continuous dialogue system architecture formalized by Skantze and Schlangen (2009).

2 Motivation and Related Work

2.1 Conceptual Models of Turn-Taking

Two models of turn-taking have been proposed in the turn-taking literature: Duncan's signal-based model (Duncan, 1972) and Sacks, Schegloff, and Jefferson's "simplest systematics" model (hereafter the Sacks et al. model) (Sacks et al., 1974).

Duncan's model of turn-taking proposes that speakers produce turn-keeping and turn-yielding signals that are picked up by listeners, thereby ensuring smooth floor transfer. Turn-yielding signals include, among others, changing intonation, specific syllable stress patterns, and gesture ending or relaxation.

Previous work has shown that intonational phrases at unit boundaries are signals used in end-of-turn detection (Bögels and Torreira, 2015a). Gravano and Hirschberg (2011) found that the greater the number of turn-end cues present in a phrase, the greater the likelihood of a floor transfer occurring. Similarly, Ford and Thompson (1996) found that syntactic, intonational, and pragmatic completeness are all required for smooth turn transition. One important takeaway from this model is that the *speaker* yields the turn, and is therefore the main decision maker for whether turn transition occurs. Speakers can therefore control whether the listener takes over the turn or not.

In contrast, Sacks et al. (1974) propose a model of turn-taking in which listeners can (but do not have to) take the floor at so-called Transition-Relevance-Places (TRPs). According to Sacks et al., listeners can predict (or "project") ahead of time when the TRP will occur. Once a TRP has been reached, the rules specified by Sacks et al. are that a) the current speaker may select the next speaker, b) if that does not happen, a next speaker may self-select, and c) if no speaker self-selects, the current speaker can continue.

2.2 Conceptual Implications

Interestingly, the differences between these models of turn-taking make different predictions as to the duration of the TRP as a function of whether the same or a different speaker takes the floor. In the Duncan model, the speaker controls the floor transfer using signals, allowing them to keep the rhythm of the conversation steady. In contrast, in the Sacks et al. model, when a speaker arrives at at TRP and has not selected the next speaker, the speaker can only continue their turn after having established that the listener did not self-select. This predicts that if the Sacks et al. model is correct, we will see shorter TRP durations when there is a speaker change than when there is not.

Therefore, the Sacks el al. model predicts that there are two separate distributions of TRP duration: one for TRPs at speaker switches and the other for TRPs at continuations. Since, according to the rules, a listener has the first option for uptake during a TRP, we expect that the TRP duration for speaker switch is faster than for speaker continuation. Of course, the probability distributions are likely to overlap. A speaker may sometimes continue with a small pause or there may be a long pause before a speaker switch. We interpret this model to mean that the speaker switch distribution will be generally faster than the speaker continuation distribution.

There is a third possibility: that speaker continuation is faster than speaker switch. While neither model predicts this, it could happen, for instance, if we assume the Duncan model is correct, and listeners do not detect the turn-yielding cues, or detect them too late. This implies if we find speaker continuations to be faster than speaker switches, it will be evidence for Duncan's model, and against Sacks et al.'s model.

2.3 Application in Spoken Dialogue Systems

Detecting the end of turns and timing speech production is vital for a spoken dialogue system to engage in smooth turn-taking. Accordingly, there are a number of approaches for automated end of turn detection in existing literature (e.g., Masumura et al. (2018, 2019)). A number of approaches have also been proposed for quick turn-transitions in spoken dialogue systems. For example, Gervits et al. (2020) found their incremental model was ready to reply to an utterance 635 ms ($\pm 197ms$) before the end of a turn. Since the mean gap between turns is generally between 0 ms and 200 ms (Heldner and Edlund, 2010; Stivers et al., 2009), this leaves an agent with significant temporal space within which to decide when to start turn production. Our goal, in contrast, is to address the characteristics of *naturalness* in smooth turn-taking timing (Edlund et al., 2008). Therefore, we assume an existing model for end of turn detection and propose a extension module of natural turn taking timing in spoken dialogue systems.

3 Empirical Models

In this section, we fit two Bayesian models of TRP duration: one that assumes a single distribution of all the TRPs in a dialogue, and one that assumes that the distribution is different for speaker switches and speaker continuations.

In what follows, we will first describe the empirical data that forms the basis for our models. Next, we provide a detailed description of the two probabilistic models informed by our conceptual models. We then describe the implications of our findings for turn-taking and speaker selection. Finally, we propose an evidence-based turn-taking propensity function for natural speech production decisions after the end of a TCU.

3.1 Data

We are interested in the duration of TRPs i.e., the timing between TCUs, in natural dialogue. Therefore, our data must consist of dialogue with TCU-level segmentation and highly accurate timing (down to the millisecond). We gathered this information from two different transcriptions of the Switchboard corpus—a corpus of dyadic telephone conversations (Godfrey and Holliman, 1993). The Mississippi State University transcriptions (MSU)[1] provide word-by-word timing, which has been hand-corrected to reduce word error rates to below 1%. The Switchboard Dialogue Act Corpus (SwDA)[2] segmented the Switchboard corpus into TCUs in order to annotate dialogue acts.

The Switchboard corpus is appropriate for this task because participants do not have access to many of the cues of face-to-face interaction (Duncan, 1972). This simplifies the work to only account for spoken language. Although using a corpus of telephone conversations may limit the applicability of our work in face-to-face interaction, it is appropriate for systems where this information is not available (Bosch et al., 2004).

Here, we outline the preprocessing steps applied to the data for the work presented in this paper. First, we merge MSU and SwDA transcripts of the same conversation to create a subset of the Switchboard corpus with transcriptions segmented at the TCU-level and annotated with accurate timing information. From this subset, we selected only conversations where the exact word-level matches were at least 90% of the words in the conversation and the total uncaught word error rate was below 2%. This allowed us to maintain data quality and yielded 75 conversations with acceptable timing information.

Next, we filter our timing data based on the following reasons. Our data consists of the duration of TRPs between TCUs. This duration may be positive or negative for speaker-switch TRPs (i.e., pauses and overlaps). To make a reasonable comparison between values of two different domains, we fit a truncated distribution to the data. For an overlap, we know that a speaker may not overlap with oneself and there is an obvious 'barge-in' when the overlap occurs. Therefore, we set a TRP floor above 0 ms. Further, previous research shows that pauses of one second or longer may be considered trouble sources in conversation (Jefferson, 1983; Roberts et al., 2006). Trouble source detection is out of the scope of this work. Therefore, we removed all TRPs with a duration greater than 1000 ms, which has the additional benefit of removing outliers from the data that might have skewed our models. Finally, we have two datasets: one for speaker switch TRPs and one for speaker continuation TRPs.

We recognize that this subset of data excludes overlaps, which are common in natural dialogue. However, this paper reasons about turn-taking

[1]The MSU corpus is hosted on OpenSLR.

[2]The SwDA corpus is available through Stanford.

through the duration of a silence. Our models do not make claims regarding when speech reasoning may occur, and instead focus on resultant behaviors that are exhibited [3].

The models and analyses below are based on the 4563 TRPs in our filtered dataset. 2686 of these TRPs were followed by speaker switches and 1877 were followed by speaker continuations. All models described are Bayesian models using truncated normal distributions with lower bounds of 0 ms and upper bounds of 1000 ms, in order to conform to the assumptions outlined above. The models were fit using `pymc3` version 3.11.4, a probabilistic programming package for Bayesian modeling. All priors[4] were designed to be weakly informative based on previous research in the field(Stivers et al., 2009; de Ruiter et al., 2006b). Weakly informative priors are also considered best-practice when using Markov-chain Monte Carlo (MCMC) Bayesian updating (Lemoine, 2019).

3.2 Speaker-Agnostic Model

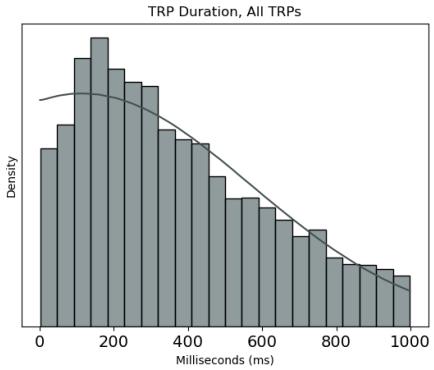


Figure 1: This figure shows a histogram with 50 ms bins of all TRPs with duration between 0 ms and 1000 ms. The best-fit truncated normal curve line is also shown.

The speaker-agnostic probabilistic model assumes that TRPs have a single underlying distribution. The assumption is that all TRPs are a function of the rhythm of the dialogue, controlled by the speaker, and pause durations are not influenced by who was speaking before the pause. Under this model, when there is a pause in the conversation, each participant has the same chance of deciding to continue.

As shown in Figure 1, the estimated mode TRP duration under these assumptions is in the 150–200 ms bin and the mean is 374 ms. The mean TRP duration, according to the posterior predictive model, is 375 ms. Since we are fitting a truncated normal distribution, the mean will be larger than the mode because the distribution is right-skewed. We see this in both the data and the model. The standard deviation of both the data and the posterior predictive model is 250 ms—indicating that the model has a good fit when assessed using the first two statistical moments. It is interesting to note that the mode of our empirical data is in the range 150–200 ms. This is the same mode range that previous work has determined for floor transfer offset, based only on the speaker switch condition (Levinson and Torreira, 2015; Heldner and Edlund, 2010; Stivers et al., 2009). Note that this previous research has focused on floor transfer for entire turns only, which is easier to operationalize since it does not require segmenting turns into TCUs.

	$\mu_{\text{hdi 3\%}}$	μ_{mean}	$\mu_{\text{hdi 97\%}}$	σ_{mean}
μ	56	110	167	30
σ	421	458	495	20

Table 1: This table describes the parameters of the best-fit truncated normal curve for all TRPs greater than 0 ms and no more than 1000 ms. The high-density intervals are given for a 94% high density interval i.e., the models predict only 3% probability that the true values lie above or below these intervals. The σ_{mean} terms are another measure of confidence, though not sensitive to skew.

3.3 Speaker-Sensitive Model

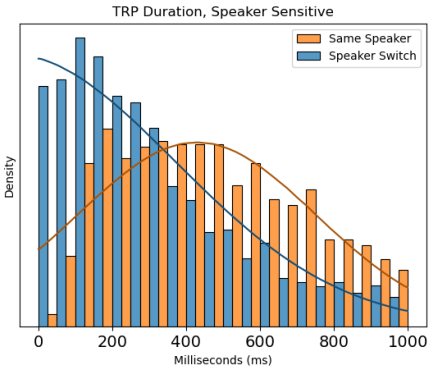


Figure 2: This figure shows a histogram of the empirical TRP data from 0 ms to 1000 ms broken down into 50 ms bins and in two conditions: speaker switch and no speaker switch. The best-fit truncated normal distribution lines for each condition are also shown.

We use the speaker-sensitive probabilistic model

[3] Alternatives to our models are in Appendix A.3.

[4] Prior distributions useds can be found in Appendix A.1.

to test the prediction from the Sacks et al. model. If this model is correct, we expect to see a different TRP distribution for the continuation and switch conditions. This is because the rules specified in their model lead to shorter TRPs when there is a speaker switch than when the same speaker continues, because the speaker first has to wait to see if a speaker self-selects before continuing their turn.

	$\mu_{\text{hdi 3\%}}$	μ_{model}	$\mu_{\text{hdi 97\%}}$	σ_{model}
μ_{switch}	-164	-85	-10	41
σ_{switch}	417	451	488	19
$\mu_{\text{continuation}}$	407	428	447	11
$\sigma_{\text{continuation}}$	300	323	347	13

Table 2: These statistics describe best-fit truncated normal curves for the independently fit curves. $switch$ variables are for cases where a speaker switch occurs at a TRP. The $continuation$ condition has the same speaker before and after the TRP. The high-density intervals are given for a 94% interval i.e., the models predict only 3% probability that the true values lie above or 3% below these intervals. Similarly, the σ_{model} terms are a measure of confidence that is not sensitive to skew.

In our speaker-sensitive model, we fit two distributions: one for speaker switch and one for a speaker continuation. The data contains 2686 TRPs where the speaker switches and 1877 TRPs where the speaker continues for all pauses in conversation from 0 ms to 1000 ms. We expect shorter pauses to be followed by a different speaker while longer pauses are followed by the same previous speaker. A fast speaker switch entails understanding and uptake by the interlocutor. A pause and continuation, in contrast, gives space for the listener to take the floor before the current speaker continues their own turn.

We found distributions that are substantially different for the speaker switch and continuation conditions. The Kolmogorov-Smirnov statistic for the two categories is 0.289 ($p < 0.01$). In the speaker switch condition, the mean of the best-fit posterior predictive is 315 ms, a bit slower than the data mean of 311 ms. The mode of the data shows that the floor transfer pause duration is 100–150 ms when binned into 50 ms segments, which aligns with previous work. This means that there is a preference toward fast responses. As a reminder, we filtered out any overlapping speech since it is outside the scope of this paper, even though we want to note that work on floor transfer offset (e.g., de Ruiter et al. (2006a); Heldner and Edlund (2010); Riest et al. (2015)) shows that overlap is a common phenomenon.

For speaker continuation, the data mean is 462 ms and the posterior predictive model mean is 459 ms. The mode of the data is 150–200 ms, although the values are very close for many other bins from about 100 ms to 600 ms, as shown in Figure 2. These data align closely to the predictions made by Sacks et al.'s model—speaker switch happens quite quickly, and speaker continuation somewhat later.

3.4 Model Selection

In Sections 3.2 and 3.3, we defined and fitted two Bayesian models to test differential predictions of the conceptual models presented in Section 2.1. We showed that each empirical model has a good quantitative and qualitative fit with the empirical data. We now want to know whether the model that incorporates speaker information is better even if we take into account that it has an extra parameter.

We will use linear mixed effects regression models, which are a common tool for differentiating trends based on groups within a population. We created two models, one with a speaker switch included, and one without. To account for different aspects of individual conversations, both models included the conversation identifier as a random factor. We used the `rstanarm` package in R (Goodrich et al., 2020) because it allowed us to use *bridge sampling* (Gronau et al., 2020) to compute Bayes Factors for the purpose of model comparison.

Our analysis shows that the data is 1.43×10^{80} times more likely under the speaker-sensitive model than under the speaker-agnostic model, even when correcting for the higher model complexity of the speaker-sensitive model. This constitutes decisive evidence (Wetzels et al., 2011) for next-speaker being influenced by TRP duration, supporting Sacks et al.'s model of turn-taking. Our results have two implications: (1) when responding, gaps should be minimized so that the speaker does not take the silence as an invitation to continue their own turn, and (2) after speaking, a response should come within the first few hundred milliseconds. Any longer, and the speaker may want to continue their own turn to maintain progressivity (Stivers and Robinson, 2006).

3.5 Turn-taking Propensity Function

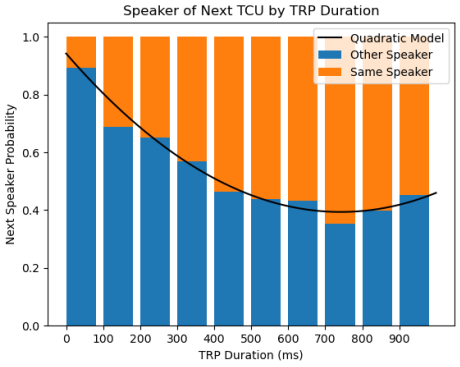


Figure 3: This figure shows the proportion of speaker-switch vs. speaker-continue events for each 100 ms TRP bin between 0 ms and 1000 ms, along with the best-fit quadratic line.

We have described two conceptual models of how turn-taking works, built probabilistic models based on these conceptual models, and established that the speaker-sensitive model inspired by Sacks et al. (1974) fits the data much better. We will now explore how we can use this knowledge to improve *when* conversational agents initiate their turn. To answer this question, we have one missing piece: we need to determine the propensity for speaker switch as a function of TRP duration. Note that the speaker-sensitive model we have formulated can be seen as two separate models: one for speaker switch and one for continuation. As mentioned before, the speaker switch condition was generally more frequent: 2686 TRPs with speaker switch and 1877 with speaker continuation in our dataset. In this section we will explore the relative proportion of speaker switch and continuation as a function of transition time.

Figure 3 shows the proportion of speaker switch and speaker continuations in our data. It shows that, as a silence grows longer, the relative propensity for a speaker to continue initially increases, while the relative propensity for a speaker switch decreases. However, as the silence continues, the share of floor holding decreases. We fit a basic quadratic curve to the floor transfer trend shown in Figure 3. The function below gives the maximum likelihood estimate for probability of speaker-switch as a best-fit quadratic function of the number of milliseconds of silence (t) since the previous turn ended.

$$P_{\text{switch}} = (9.70\times10^{-7})t^2 - (1.48\times10^{-3})t + 0.933$$

It is important to note that our analysis only looks at the first second of silence after a TCU. We did perform a cursory exploration of longer pauses, to check if there were obvious trends. We found that for silences between 1000 ms and 2500 ms, 56% of TCUs were floor-hold (speaker continuation). We caution against over-interpretation of these numbers, as there were 4563 TRPs between 0 ms and 1000 ms, but only 672 between 1000 ms and 2500 ms. Gaps longer than a second in conversations are rare in conversation (Jefferson, 1983), and may have a variety of causes.

A spoken dialogue system can use this formula and our two empirical models above in two ways: (1) timing its own responses and (2) setting response seeking limits. Current techniques allow for extremely fast response rates in spoken dialogue systems. An agent implementing our models can choose times that are acceptable to human dialogue speed. These times do not need to rely on heuristics like the mean FTO or barge-in mechanics, but can keep conversation at a fluid and natural pace on an utterance-by-utterance basis. Our model suggests that an agent should respond to a turn within 394 ms—the point at which each speaker has equal propensity to speak—ideally around 150–200 ms after a turn end, where the probability of a speaker change is still close to maximal.

A spoken dialogue system can also incorporate the propensity function during language generation to make sure that its turn-internal pauses are not too long or too short. If the system knows it wants to continue the turn in a subsequent TCU, it should flow fluidly, rather than give space for the interlocutor to respond.

Finally, if the planned turn is over, an agent could set a maximum listening time, after which it prompts a response or clarifies its previous statement. Our findings show that the agent should aim to do this around 762 ms, the minimum point of our speaker-switch function. Pauses of longer than a second are signs of trouble in a conversation, so continuing a turn is preferable than waiting indefinitely for a response (Jefferson, 1983; Roberts et al., 2006). Adding this functionality to a spoken dialogue system will provide an agent with the ability to ensure that the conversation progresses, and even prompt an interlocutor if they are unresponsive.

The function presented here is meant to be a baseline for turn-taking mechanisms. There are clear paths for extending it, like sensitivity to di-

alogue acts, ellipses, or prosody, but the overall effect should be similar in aggregate since the data here is presented in aggregate.

4 Continuous Module Proposal

In this section, we outline a proposal for operationalising our turn-taking propensity function for timing turn-taking. First, we outline relevant components of the incremental dialogue processing architecture proposed by Schlangen and Skantze (2011). Next, we define the *minimal* module implementing the proposed timing method, as well as possible extensions to incorporate existing turn-taking methods (e.g, Bögels and Torreira (2015b)).

We use the Schlangen and Skantze (2011) architecture for its continuous and incremental properties, which may be useful for comparing different timing methods. However, our proposed method is based simply on timing and does not have a strong dependence on any specific architecture.

4.1 Spoken Dialogue System Architecture

The conceptual model of incremental processing described in Schlangen and Skantze (2011) has two basic components. Incremental Units (IUs) are the basic units of processing and contain payloads (e.g., audio streams, words etc.) that can be processed by Incremental Modules (IMs). Each IM has a Left Buffer (LB) to store incoming IUs and a right buffer to store outgoing IUs. An IM also has a processor that consumes LB IUs and produces RB IUs. IMs communicate with each other by adding IUs to their RB, which is immediately available for LB consumption of connected IMs. Note that the rate of RB IU production does not need to match the rate of LB IU consumption.

Additionally, IUs may be connected to one another using relations, which effectively track the flow of information throughout the system. While there can be many different types of relations, we introduce two—spatial and grounded-in. The spatial relation connects IUs produced by a single IM. For example, for an IM generating turns from words, spatial links may be used to connect words that form the same turn. Second, the grounded-in relation can be used by IMs to connect RB IUs to their corresponding LB IUs. For example, this may allow a word recognized by an Automatic Speech Recognition (ASR) IM to be connected to the corresponding audio signal.

Finally, both IUs and IMs contain specified properties and operators. Each IU contains basic metadata type information that may be used for decision-making in individual IMs. This includes information for an IU to indicate its relations with other IUs, the confidence of the IM in the IU data, whether the IU result is final, and whether the IU has been processed by a specific IM. Similarly, IMs must implement certain methods including a purge method to reset the module's internal state, a new IU update method to update the module state based on incoming information, and a commit method to finalize the IUs in its RB.

4.2 Module Incremental Units

Our proposed module aims to provide more granular turn-taking timing information to the spoken dialogue system compared to existing approaches, which plan and execute entire turns (Jokinen et al., 2013). Therefore, it produces RB IUs whose payloads are waiting times after which the dialogue system should take the next turn. Additionally, the IU includes a confidence value to incorporate our finding that the relative pressure to speak is time-sensitive within the first second of a gap (as shown in Figure 3). Variations in this value indicate the importance of speaking at a specific time.

Finally, a minimal turn-taking timing module would receive IUs where the payload may be an input signal (e.g., the audio signal). Additionally, it requires the ability to determine the elapsed time between turns in the conversation up to that point. Therefore, turn-taking module IUs will use the grounding-in relation to determine the specific IUs the results are based on.

4.3 Turn-Taking Timing Module

The turn-taking module consumes LB IUs to produce RB-IUs, with the invariant $size(RB) \leq size(LB)$, and implements the purge, new IU update, and commit operators. Here, the purge operator is vital in removing all IUs when a connected module, such as the ASR module, indicates that an interlocutor has taken the floor. In this case, any considerations for the time after which the system may start a turn depends only on the following turn and previous results may be discarded. The processor also implements the new IU update method to modify its internal state based exclusively on new LB IUs. It may then produce new turn-timing decisions based on the updated information. The commit method then finalizes the best-guess time

after which a turn may be started by the spoken dialogue system.

4.4 Module Extensions

In this section, we proposed a *minimal* incremental module for timing turn-taking based primarily on TRP duration. However, there are additional sources of information, not considered in this paper, that a spoken dialogue system may use when deciding when to produce a turn after a TCU. For example, intonational and semantic end of turn cues can be used to predict when floor transfer may occur (Lala et al., 2019; de Ruiter, 2019), and non-verbal cues may also be used to time turn-taking (de Ruiter et al., 2006a; Duncan, 1972). These may be modeled as individual IMs in the framework we use and connected to the turn-taking IM to allow information to be integrated when making turn-taking decisions. Additionally, the module may remember wait times proposed across a conversation and adjust for the specific interlocutor. For example, if there is frequent overlap if speech is produced after the proposed wait time, then future wait time estimates may be corrected. Similarly, short gaps by the interlocutor may be mimicked by the spoken dialogue system.

5 Conclusion and Future Work

In this study, we started by comparing two conceptual models of turn-taking—Duncan's "turn-yielding" cue model and Sacks et al.'s "simplest systematics", each of which makes different predictions about the organization of turns in conversation. We used data from the Switchboard corpus to fit two probabilistic models of TRP duration based on these conceptual models: a speaker-agnostic model compatible with Duncan's conceptual model and a speaker-sensitive model inspired by on Sacks' et al.'s conceptual model. Both models have a good quantitative and qualitative fit with the empirical data.

However, when comparing the two models directly, we found that the speaker-sensitive model i.e., Sacks et al.'s model, was decisively better at predicting the data than the speaker-agnostic model. We explored the implications of this finding for turn-taking systems. We showed that the likelihood of a speaker beginning a TCU during a pause in conversation changes as the pause lengthens. For short pauses, it is more probable that the speaker will switch, but as the pause continues, the original speaker becomes more likely to continue their turn.

Our work supports the notion that, for proper turn-taking, detecting and/or anticipating the end of turns is not sufficient. People are sensitive to the pauses and gaps in conversation and organize their speech to take into account this paralinguistic signal. We described the regularities that we found, and outlined implementations for dialogue systems to incorporate our findings. For naturalistic turn-taking adhering to these subtle norms is important, and we described first steps towards implementing this in agents.

In future work, we plan on implementing the spoken dialogue system we have proposed in this paper. While we have established and operationalized normative turn-taking behavior based on human conversations, it is important to investigate whether and to what degree findings from human-human data generalize to communication with spoken dialogue systems. Therefore, evaluating the conversational *naturalness* of our system through human-subject experiments is a relevant next step and will provide insight into the organization of turns in conversation, both for human-human and human-agent communication.

Acknowledgments

This paper was funded in part by a grant from the Data Intensive Studies Center at Tufts University.

References

Dan Bohus and Eric Horvitz. 2010. Computational models for multiparty turn taking. *Technical Report. Microsoft Research Technical Report MSR-TR 2010-115*.

Louis ten Bosch, Nelleke Oostdijk, and Jan P. de Ruiter. 2004. Durational aspects of turn-taking in spontaneous face-to-face and telephone dialogues. In *International Conference on Text, Speech and Dialogue*, pages 563–570. Springer.

Sara Bögels and Francisco Torreira. 2015a. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52:46–57.

Sara Bögels and Francisco Torreira. 2015b. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52:46–57.

Starkey Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.

Jens Edlund, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson. 2008. Towards human-like spoken dialogue systems. *Speech communication*, 50(8-9):630–645.

Cecilia E Ford and Sandra A Thompson. 1996. intonational, and pragmatic resources for the. *Interaction and grammar*, 13:134.

Felix Gervits, Ravenna Thielstrom, Antonio Roque, and Matthias Scheutz. 2020. It's about time: Turn-entry timing for situated human-robot dialogue. In *Proceedings of the Special Interest Group on Discourse and Dialogue*.

John Godfrey and Edward Holliman. 1993. Switchboard-1 release 2 ldc97s62. *Linguistic Data Consortium*.

B. Goodrich, J. Gabry, I Ali, and S. Brilleman. 2020. Rstanarm: Bayesian applied regression modeling via stan r package v. 2.19.2.

Agustín Gravano and Julia Hirschberg. 2011. Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3):601–634.

Quentin F. Gronau, Henrik Singmann, and Eric-Jan Wagenmakers. 2020. bridgesampling: An r package for estimating normalizing constants. *Journal of Statistical Software*, 92(10).

Kohei Hara, Koji Inoue, Katsuya Takanashi, and Tatsuya Kawahara. 2019. Turn-taking prediction based on detection of transition relevance place. In *INTERSPEECH*, pages 4170–4174.

Mattias Heldner and Jens Edlund. 2010. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568.

Gail Jefferson. 1983. Notes on a possible metric which provides for a standard maximum silence of approximately one second in conversation. *Tilburg papers in language and literature*.

Kristiina Jokinen, Hirohisa Furukawa, Masafumi Nishida, and Seiichi Yamamoto. 2013. Gaze and turn-taking behavior in casual conversational interactions. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(2):1–30.

Casey Kennington, Daniele Moro, Lucas Marchand, Jake Carns, and David McNeill. 2020. rrsds: Towards a robot-ready spoken dialogue system. In *Proceedings of the 21th annual meeting of the special interest group on discourse and dialogue*, pages 132–135.

Divesh Lala, Koji Inoue, and Tatsuya Kawahara. 2019. Smooth turn-taking by a robot using an online continuous model to generate turn-taking cues. In *2019 International Conference on Multimodal Interaction*, pages 226–234.

Nathan P Lemoine. 2019. Moving beyond noninformative priors: why and how to choose weakly informative priors in bayesian analyses. *Oikos*, 128(7):912–928.

Stephen C Levinson and Francisco Torreira. 2015. Timing in turn-taking and its implications for processing models of language. *Frontiers in psychology*, 6:731.

Anthony J Liddicoat. 2004. The projectability of turn constructional units and the role of prediction in listening. *Discourse Studies*, 6(4):449–469.

Ryo Masumura, Mana Ihori, Tomohiro Tanaka, Atsushi Ando, Ryo Ishii, Takanobu Oba, and Ryuichiro Higashinaka. 2019. Improving speech-based end-of-turn detection via cross-modal representation learning with punctuated text data. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1062–1069. IEEE.

Ryo Masumura, Tomohiro Tanaka, Atsushi Ando, Ryo Ishii, Ryuichiro Higashinaka, and Yushi Aono. 2018. Neural dialogue context online end-of-turn detection. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 224–228.

Thilo Michael. 2020. Retico: An incremental framework for spoken dialogue systems. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 49–52.

Carina Riest, Annett B Jorschick, and Jan P de Ruiter. 2015. Anticipation in turn-taking: mechanisms and information sources. *Frontiers in psychology*, 6:89.

Felicia Roberts, Alexander L. Francis, and Melanie Morgan. 2006. The interaction of inter-turn silence with prosodic cues in listener perceptions of "trouble" in conversation. *Speech Communication*, 48(9):1079–1093.

Jan P. de Ruiter. 2019. Turn-taking. *The Oxford Handbook of Experimental Semantics and Pragmatics*, page 536–548.

Jan P. de Ruiter, H. Mitterer, and N. J. Enfield. 2006a. Projecting the end of a speaker's turn: A cognitive cornerstone of conversation. *Language*, 82(3):515–535.

Jan P. de Ruiter, Holger Mitterer, and Nick J Enfield. 2006b. Projecting the end of a speaker's turn: A cognitive cornerstone of conversation. *Language*, 82(3):515–535.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.

David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue & Discourse*, 2(1):83–111.

Margret Selting. 2000. The construction of units in conversational talk. *Language in society*, 29(4):477–517.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Gabriel Skantze and Anna Hjalmarsson. 2010. Towards incremental speech generation in dialogue systems. In *Proceedings of the SIGDIAL 2010 Conference*, pages 1–8.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 745–753, Athens, Greece. Association for Computational Linguistics.

Tanya Stivers, Nicholas J Enfield, Penelope Brown, Christina Englert, Makoto Hayashi, Trine Heinemann, Gertie Hoymann, Federico Rossano, Jan Peter De Ruiter, Kyung-Eun Yoon, et al. 2009. Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26):10587–10592.

Tanya Stivers and Jeffrey D Robinson. 2006. A preference for progressivity in interaction. *Language in society*, 35(3):367–392.

Ruud Wetzels, Dora Matzke, Michael D Lee, Jeffrey N Rouder, Geoffrey J Iverson, and Eric-Jan Wagenmakers. 2011. Statistical evidence in experimental psychology: An empirical comparison using 855 t tests. *Perspectives on Psychological Science*, 6(3):291–298.

Ran Zhao, Oscar J Romero, and Alex Rudnicky. 2018. Sogo: a social intelligent negotiation dialogue system. In *Proceedings of the 18th International Conference on intelligent virtual agents*, pages 239–246.

A Appendix

A.1 Statistical Model Priors and Parameters

The statistics in the table describe the priors used to fit the truncated normal distributions for the models described in the turn-taking models Sections 3.2 and 3.3. For each model, identical priors were used so that differences between the models were functions of the data, not of the priors. μ was drawn from a normal distribution with priors shown, and σ was drawn from a Gamma distribution with priors shown. Gamma distributions are typically parameterized with α and β parameters, but `pymc3` allows for parameterization with μ and σ, which is what we chose. All models took 10,000 samples with 6,000 tuning steps and a target acceptance rate of 0.9.

μ_μ	200
μ_σ	75
σ_μ	300
σ_σ	200

Table 3: This table shows the prior parameters used in each of the statistical models.

A.2 Model Comparison Methods

To compare the two models in Section 3.4, while taking into account model complexity, we built two linear mixed effects models using the `stan_glmer` function in the `rstanarm` package for the `R` programming language. This function fits a linear model of the data based on the parameters involved. Both models corrected for the particular conversation as a random effect, and one took into account whether there was a speaker switch at the TRP. Unlike `pymc3`, `Stan` does not require the user to specify priors, but assigns weakly informative default priors based on the data. Using the `Stan` models allowed use the `bayesfactor_models` function of the `bayestestR` package to compare the models and determine if the speaker switch model better explained the data than the no speaker switch model.

A.3 Generalized Models

We recognize that truncated normal models may not be the most robust method of modeling our data - which does not include gaps and overlaps. Additionally, the truncated normal distribution used for the speaker continuation condition is only positive valued. Therefore, we present preliminary analyses on alternative models that may be used to fit our data.

The analyses presented in the paper establish that the speaker switch and continuation conditions are different and provide a justification for creating stochastic models to describe these phenomenon separately. Therefore, we built a Student's t-distribution model for the speaker switch condition to approximate the normal model when ν is large. The dataset used for this model includes all TRPs with duration in range -800 ms to +2500 ms, and only excludes outliers that were likely to be transcription artifacts. The widely-applicable information criterion (WAIC) score for the Student's t is 79,131, while the truncated normal (expanded to the new lower and upper limits) is 79,707, showing some improvement. The Kolmogorov-Smirnov

statistic is also reduced to 0.425 from 0.741 (both $p < 0.001$).

	$\mu_{\text{hdi 3\%}}$	μ_{model}	$\mu_{\text{hdi 97\%}}$	σ_{model}
ν	2.70	2.96	3.23	0.142
μ	95	105	116	5.50
σ	287	297	307	5.63

Table 4: This table describes the posterior model parameters used for the Student's t-distribution model in the speaker switch condition.

Additionally, we built a Gamma model for the speaker continuation condition using TRPs with duration up to 2500 ms. This model describes a variable with a positive domain more elegantly than a truncated normal model. The WAIC score of the Gamma and truncated normal (with adjusted bounds) models is 33,845 and 34,125 respectively, which again shows improvement. The Kolmogorov-Smirnov statistic is also reduced from 0.480 to 0.173 (both $p < 0.001$).

	$\mu_{\text{hdi 3\%}}$	μ_{model}	$\mu_{\text{hdi 97\%}}$	σ_{model}
μ	604	621	638	9.00
σ	419	435	451	8.64
α	1.95	2.06	2.16	5.71e-2
β	3.10e-3	3.29e-3	3.49e-3	1.04e-4

Table 5: This table describes the model parameters used for the Gamma model for the speaker continuation condition.

Each of the models presented show potential next steps for improving modeling our data. Unfortunately, for our purposes, they are not directly comparable.

A.4 Data Description

The truncated normal models described in Section 3 exclude some data – which was necessary for our analysis. Here, we include descriptions of our raw data to provide further information on our analyses.

Duration	Number of TRPs
0 ms	3565
0–1000 ms	1933
$>$ 1000 ms	380

Table 6: This table shows the number of speaker-continuation TRPs in bins of different duration.

The above descriptions show that overlapping speech is extremely common in our dataset, making up about 42% of the speaker switch conditions.

Duration	Number of TRPs
$<$ 0 ms	2217
0–1000 ms	2703
$>$ 1000 ms	296

Table 7: This table shows the number of speaker-switch TRPs in bins of different duration.

Additionally, though we only consider positive values, speaker continuations with pauses of 0 ms are the majority of speaker continuation conditions —61%. The models we have presented model reasoning *through* a silence, and are therefore sound in the assumption that a silence exists. However, any turn taking model that *only* considers turn taking via silences will be incomplete.

DG^2: Data Augmentation Through Document Grounded Dialogue Generation

Qingyang Wu[1] **Song Feng**[3] **Derek Chen**[2] **Sachindra Joshi**[3] **Luis A. Lastras**[3] **Zhou Yu**[1]
[1]Columbia University [2]ASAPP [3]IBM Research AI
{qw2345, zy2461}@columbia.edu, dchen@asapp.com
{ sfeng@us, jsachind@in, lastrasl@us}.ibm.com

Abstract

Collecting data for training dialog systems can be extremely expensive due to the involvement of human participants and the need for extensive annotation. Especially in document-grounded dialog systems, human experts need to carefully read the unstructured documents to answer the users' questions. As a result, existing document-grounded dialog datasets are relatively small-scale and obstruct the effective training of dialogue systems. In this paper, we propose an automatic data augmentation technique grounded on documents through a generative dialogue model. The dialogue model consists of a user bot and agent bot that can synthesize diverse dialogues given an input document, which are then used to train a downstream model. When supplementing the original dataset, our method achieves significant improvement over traditional data augmentation methods. We also achieve competitive performance in the low-resource setting.

1 Introduction

Most of human knowledge is stored in the form of documents, ranging from answering factoid questions (Reddy et al., 2019) to providing how-tos on millions of tasks (Zhang et al., 2020a). How to comprehend and retrieve relevant knowledge from documents given a user query is a challenging research problem. Inspired by real-world applications, there have been more works (Rajpurkar et al., 2016a, 2018; Kwiatkowski et al., 2019; Yang et al., 2015) that aims to tackle this challenge. In this work, we focus on the task of conversational information seeking based on the associated documents, which are often referred to as document-grounded dialogue systems (Ma et al., 2020).

Recent works have introduced various datasets for building document-grounded conversational question answering and dialogue systems. Some work such as QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2019) first explored the direction of conversational question answering. Then, ShARC (Saeidi et al., 2018) added follow-up questions by agents. Later, Doc2Dial (Feng et al., 2020a) further included the dialogue actions and domains, which aims to simulate more kinds of real-life scenarios. However, such dataset is typically hard to scale up to new domains, as it requires carefully crafted dialogue flows and expensive human annotations.

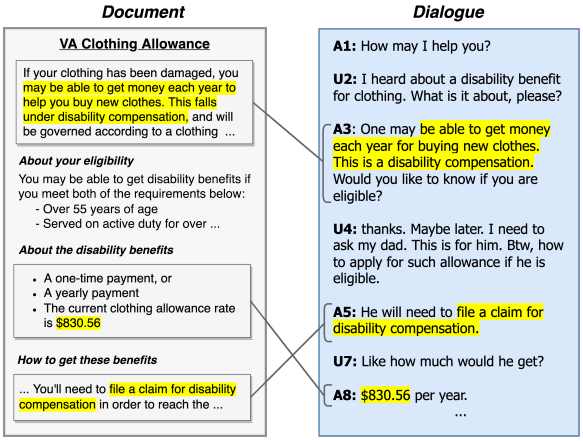


Figure 1: An example from Doc2Dial of dialogue conversation produced from grounding to an associated document. The agent must select the correct spans and engage in a fluent manner to generate a proper response.

However, as the relations between conversations and documents become more complex, the cost of collecting large-scale datasets also becomes more expensive. As a consequence, one main obstacle for developing scalable and effective document grounded dialog systems is the lack of sufficient data. In chit-chat scenarios, recent works such as DialoGPT (Zhang et al., 2020b), Meena (Adiwardana et al., 2020), and Blender (Roller et al., 2021) have achieved high performance by taking the advantage of training on a large-scale corpus. Similarly, task-oriented dialog systems such as ARDM (Wu et al., 2021) and SimpleTOD (Hosseini-Asl et al., 2020) have also utilized large-scale corpora or pre-trained models to achieve

Proceedings of the SIGdial 2022 Conference, pages 204–216
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

good performance. The aforementioned models were trained with millions of samples, while the current document-grounded dialogue datasets like Doc2Dial (Feng et al., 2020a) only contain thousands of conversations. Training on such a small-scale dataset constrains the performance of neural network models. Therefore, augmenting existing datasets can help build a more effective document-grounded dialogue system.

One popular approach to augmenting datasets is to paraphrase existing seed data. The most straightforward form of paraphrasing is to directly use a model trained to generate paraphrase pairs (Gao et al., 2020). Back-translation serves as another type of paraphrasing, which first translates a sentence into another language and then back again (Chadha and Sood, 2019; Bornea et al., 2021). Back-translation ensures the quality and correctness of the augmented data and often shows improvement in downstream models. Both methods aim to provide variety to the training data without greatly altering the semantics of the original sentences. However, these methods only operate on the existing dialogue data and fail to take advantage of the available document for augmentation.

Another direction for data augmentation is to generate examples from scratch by grounding to auxiliary documentation. Lewis et al. (2021) generate question-answer pairs with a model pre-trained on available training data. This often requires additional filtering or denoising measures to ensure correctness of generated data. Also, these models are built for the purposes of single-turn question answering, rather than multi-turn dialogues.

Inspired by Alberti et al. (2019), we propose an automatic document-grounded dialogue generation (DG^2) method that augments the amount of data available for training a dialogue system. The model consists of a user bot and an agent bot that alternately generates utterances to complete a conversation. The user bot includes a span extraction model that can first select a passage and then predict the rationale start and end positions inside a passage. The agent bot has a denoising mechanism to filter out generated rationales irrelevant to the conversation. The user bot begins by selecting a passage from the document that is most relevant to the current context. It then selects a rationale span from this passage and generates the user utterance. The agent bot takes the selected span from the user bot, and then checks if it can find the correct rationale span, and finally generates the agent response. This process repeats until an entire dialogue is generated.

We evaluate our model on a representative document-grounded dialog dataset Doc2Dial (Feng et al., 2020a). We test and generate additional dialogs with both the seen documents and unseen documents. We augment the original dataset and train it on a downstream model. The results show that our method improves the performance of the downstream model after augmentation. We also test scenarios of low-resource settings. We train and evaluate the generative models with only $25\%, 50\%, 75\%$ data. Experimental results show that our method perform well even when training data is scarce.

2 Related Work

2.1 Document Grounded Dialogue Systems

Document Grounded Dialogue System (DGDS) is the type of dialogue systems that the dialogues are grounded on the given documents. It helps humans to better retrieve information they want as most of human knowledge is stored in the form of documents. The study of DGDS can greatly impact the future way of interacting with knowledge.

Recently, there are many document grounded dialogue datasets proposed. Doc2Dial (Feng et al., 2020b) is a representative document grounded dialogue dataset which involved human-to-human conversations and focused on real scenarios under social welfare domains. Previous datasets such as CoQA (Reddy et al., 2019) and QuAC (Choi et al., 2018) focused on machine reading comprehension. SharC (Saeidi et al., 2018) is close to Doc2Dial. Its conversations are grounded to short text snippets, and contains follow-up questions. ABCD (Chen et al., 2021) supports customer service interactions by providing Agent Guidelines as additional documentation to aid in task-oriented conversations.

An example of DGDS from Doc2Dial is shown in Figure 1. For each turn, the agent needs to look at the specific paragraph inside the document to be capable of answering the user's questions. Moreover, the agent can also ask follow-up questions. For A3, the agent asks "Would you like to know if you are eligible?". In this way, the agent guides the user to center more on the details in the document. Due to the complexity of Doc2Dial, simulating such dialogues is highly nontrivial.

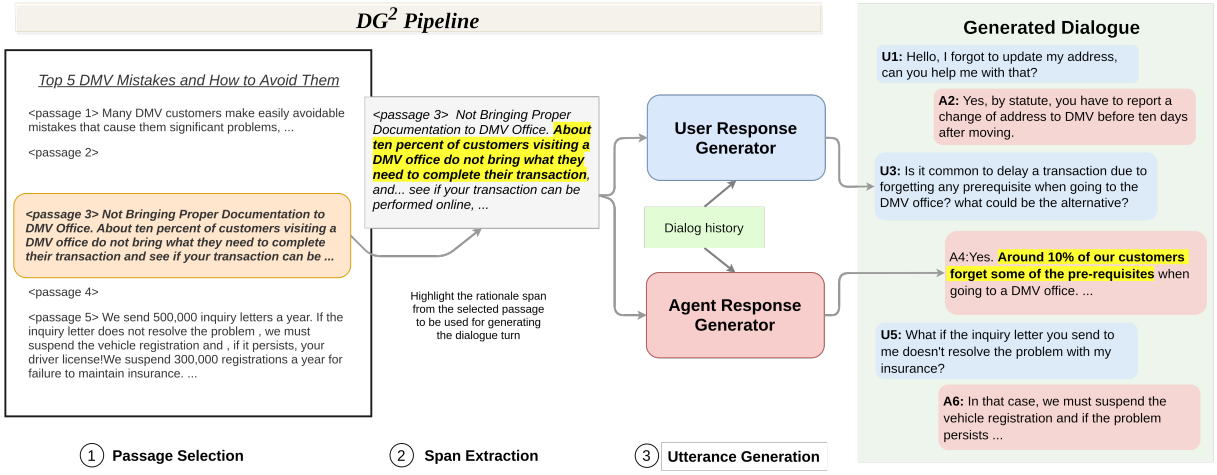


Figure 2: Overall pipeline of DG^2. Given a document and the dialogue history, DG^2 iteratively performs (1) passage selection, (2) span extraction, and (3) utterance generation to produce a completed dialogue.

2.2 Data Augmentation

Data augmentation for question answering and dialogue systems has been well-studied in the past. There are two major directions: paraphrasing existing QA pairs from seed data or generating new QA pairs from scratch.

Paraphrasing is a simple and effective technique to augment natural language datasets. It has been widely used in many NLP tasks including natural language understanding, question answering, and task-oriented dialog systems (Gao et al., 2020) to improve the downstream models' performance. In question answering, paraphrasing with back-translation (Chadha and Sood, 2019; Bornea et al., 2021) is well-studied for datasets such as SQUAD (Rajpurkar et al., 2016b).

Another approach is generating new question-answer pairs. Early question-answer generation models used rule-based methods (Rajpurkar et al., 2016b). More recently, there have been studies of neural network-based question-answer pair generation models. PAQ (Lewis et al., 2021) generated 65 million question-answer pairs based on Wikipedia and trained a retriever with the generated data.

However, existing approaches have not explored applications for conversational question answering yet, especially for document grounded dialog systems. Compared to single-turn question answering datasets like SQUAD (Rajpurkar et al., 2016b), it involves additional complexity of modeling dialog flow and interconnection naturalness. Also, instead of only providing an answer span, datasets like Doc2Dial (Feng et al., 2020b) have free-form agent responses. The agent needs to produce natural utterances conditional to the selected rationale.

Also, existing conversational question generation models (Gu et al., 2021) only focused on the quality of generations but did not address the improvement on downstream models. We design a specific dialog augmentation approach for document-grounded dialog systems. Our work can synthesize the entire conversation, and can be used to improve down-stream task's performance.

3 Document-Grounded Dialogue Setup

A dialogue can be thought of as a series of turns between two interlocutors. Within goal-oriented dialogues, we refer to the first speaker as the user, and the second speaker as the agent, whom we model as $d = [(u_1, a_1), (u_2, a_2), ...(u_t, a_t)]$. In a document-grounded setting, the conversation revolves around the topics and entities mentioned in the associated document. A document is composed of a series of text passages, which are themselves broken down further into spans.

Dialogue success is determined by following the typical success metrics for any given task, where the only difference is that the outcome of the conversation is likely to depend on the ability to reason about the contents of the document. While sophisticated architectures are certainly capable of improving document-grounding, we take a data-centric approach instead by generating new dialogues from the documents to serve as additional training data for the downstream model.

4 Data Augmentation via DG^2

We propose **D**ocument-**G**rounded **D**ialogue **G**eneration (DG^2) as a method of data augmentation. We aim to generate a complete and coherent dialogue given a document by building two bots talking to each other.

Given a document C, we can model a dialog d between the user and the agent with:

$$p(d|C) = \prod_{i=1}^{t} p(u_i, a_i|c_i \in C) \quad (1)$$

where u_i is the user turn utterance, a_i is the agent turn utterance, and c_i is the selected passage at i-th turn.

We further decompose the model into three parts: passage selection, rationale extraction, and utterance generation. We also apply a filtering model to ensure the quality of generated utterances.

4.1 Passage Selection

A document can often be very long, so it must be divided into smaller passages first. Then, we need to rank the passages, and select a relevant passage given the dialogue context. We can maximize the passage probability for c_t with contrastive loss where the positive passages are from ground truth, and the negative passages are from the same document.

$$p(c_t|\{u_i, a_i\}_{i<t}, C) \quad (2)$$

During generation, we sample from the probability distribution to select the passage. We choose to sample rather than perform greedy selection since this allows for choosing different passages given the same dialogue context, thereby increasing the diversity of the augmentation.

4.2 Rationale Extraction

Next, we further extract a rationale span from the selected passage.

$$p(r_t|\{u_i, a_i\}_{i<t}, c_t)$$

Span extraction systems typically model the start and end position of a span independently as $p(r_{\text{start}}|c) \times (r_{\text{end}}|c)$. This settings works well when the span is short, as is often the case for standard question answering tasks. However, the spans encountered in some document-grounded dialog datasets are much longer causing problems in traditional approaches. As an alternative, we propose an autoregressive method that samples the start and end position in sequentially with:

$$p(r_t) = p(r_{\text{start}}|c) \times p(r_{\text{end}}|r_{\text{start}}, c) \quad (3)$$

To ensure that the autoregressive property holds, we add the predicted start position's hidden state H_{start} and each position's hidden state H_i, and then we project the combined hidden state with a learnable function f_r to get the final predicted end position. Thus, the training objective becomes to maximize

$$r_{\text{end}} = \arg\max_i f_r(H_{\text{start}} + H_i) \quad (4)$$

When extracting a rationale, we first sample a start position from top-k options. Conditioned on this start index, we then sample the end position. This allows us to extract different rationales given the same context, which greatly improves the diversity of generated dialogues compared to using the same rationale.

4.3 Utterance Generation

Given the selected passage and the extracted rationale, we can now start to generate the user utterance and the agent utterance.

User Utterance As seen in Figure 2, user model generates a user utterance conditioned on the dialog history and the extracted rationale. Instead of only using the rationale to generate utterances, we provide the context passage along with the rationale for better performance. To tell the model where the rationale is in the passage, we highlight the rationale span by wrapping its text in the input with "[" and "]". The new passage with the rationale span information is defined as c'_t.

We then model the user utterance with a encoder-decoder where the input is the dialogue history and the passage c'_t, and the output is the user utterance.

$$p(u_t) = p(u_t|\{u_i, a_i\}_{i<t}, c'_t) \quad (5)$$

Agent Utterance Similar to user utterance generation, we model the agent utterance with a encoder-decoder.

$$p(a_t) = p(a_t|\{a_i, u_i\}_{i<t}, c'_t) \quad (6)$$

The difference is that the dialogue history now includes the previous generated user utterance. The rationale position information in the passage is processed similarly as in user utterance generation. We can repeat the user utterance and agent utterance generation process to generate the entire dialogue.

4.4 Filtering the Augmented Data

Roundtrip consistency checking (Alberti et al., 2019; Zhong et al., 2020) has previously been used to improve the correctness of generated augmentation data. It utilizes a model to double-check whether the answer span is the same as the span used to generate the question. Based on this insight, rather than tuning a sampling temperature to trade-off against noise and diversity, we instead greedily pick the rationale span and use consistency checking to filter for quality. For our purposes, we expect the extracted rationale to be aligned with the dialogue context as well as the user utterance.

We build a new passage selector and rationale extraction model such that:

$$p(\hat{c}_t|\{u_i, a_i\}_{i<t}, , u_t, C) \quad (7)$$

$$p(\hat{r}_t|\{u_i, a_i\}_{i<t}, u_t, \hat{c}_t) \quad (8)$$

where $\hat{c}_t$ is the predicted passage from the document C with the dialogue context and the generated user utterance, and $\hat{r}_t$ is the prediction rationale within $\hat{c}_t$. When the predicted $\hat{r}_t$ contradicts the previous r_t, we filter out the utterance u_t. Because rationale spans can be long and not unique, filtering based on exact match will be too strict. Instead, we use F1 word overlap for filtering.

4.5 Document Positional Information

When a document is divided into passages, it loses positional information between different passages. As a dialogue progresses, we can expect to focus more on the later part of a document, which involves more details of a topic. Therefore, it is important to incorporate the turn information and the passage position information into the model.

We use a simple yet effective method to combine the dialogue turn positional information and passage positional information. For the speaker positions we use a prompt "user{num}:" or "agent{num}:", where "num" is replaced with the number of turns so far. This allows the model to track how many turns have passed, leading to a more coherent dialog structure. For the passage positions, we embed a passage index to indicate the location of the passage within the document. Combining the two flows together, the model is able to have conversations focused on the beginning of the document at the first, and naturally shift towards the end of document later.

5 Experiments

We first introduce the datasets evaluated with our method, then the baselines for comparisons, and in the end our method's implementation details.

5.1 Datasets

	Dialogue Level				Document Level	
	#dial	#turns	#tok	%span	#doc	#tok
train	3,474	11.8	15.0	26.5	415	834
valid	661	12.1	15.3	25.8	273	821
test	661	12.0	14.9	24.5	273	809
DG^2	3,474	12.0	14.2	42.2	415	834

Table 2: Doc2Dial dataset statistics. The following abbreviations are made: 'dial' is short for dialogue, 'tok' is short for tokens, and 'doc' is short for documents. '%span' means the percentage of spans as reference.

Doc2Dial consists of two subtasks around identifying relevant spans based on dialogue context and producing cohesive responses based on extracted rationales (Feng et al., 2020a). Formulated as a span selection task, user utterance understanding requires an agent to interpret user queries in the context of the dialogue history and then select the relevant span from the associated document. Predicted spans are graded based on Exact match (EM) and F1-score. Exact match is when the predicted span exactly lines up with the actual span. F1-score balances the recall and precision of the predicted uni-grams compared to the gold span.

The second subtask is agent response prediction, which requires an agent to generate a natural language response to the user query given the dialogue context and the document. Response quality is measured by SacreBLEU metric (Post, 2018) which aims to capture how closely the predicted response lines up with the gold response. Table 2 shows Doc2Dial's dialogue-level statistics and document-level statistics.

5.2 Baselines

We compare against a number of baselines typically used to augment natural language data. In contrast to our technique, these methods all operate on the existing dialogues, whereas our method generates new dialogues from scratch from the associated document.

Easy Data Augmentation Wei and Zou (2019) propose to augment data through a series of surface

Model	Validation			Test			Span Coverage
	EM	F1	BLEU	EM	F1	BLEU	
Original data	58.13	72.61	37.08	58.34	73.25	36.89	48.27
+ EDA	**60.40**	74.30	37.72	59.71	73.62	37.63	48.27*
+ Back-translation	60.15	73.74	36.68	60.17	73.35	37.32	48.27*
+ Paraphrase	59.97	73.92	37.76	57.98	72.71	38.40	48.27*
+ DG^2	60.30	**74.34**	**38.07**	**60.92**	**74.53**	**38.57**	**57.65**

Table 1: Experimental results on the Doc2Dial dataset. EM stands for Exact Match. **Bold** means the best score. Underline means the second best. *EDA, Back-translation, and Paraphrase do not modify span information and thus are unable to increase span coverage in relation to the original data.

form alterations. In particular, Easy Data Augmentation (EDA) consists of inserting new tokens, deleting random tokens, swapping pairs of tokens, or replacing tokens with their synonyms.

Back-translation Back-translation is another strong augmentation method which first translates some text into a separate language and then back-translates to the original language. We follow BERT-QA (Chadha and Sood, 2019), in translating all user utterances to French and then back to English to augment the original dialogues.

Paraphrase Paraphrasing can be achieved by training a sequence-to-sequence model on parallel paraphrase pairs corpora. In particular, we train a BART-base model (Lewis et al., 2020a) on the MRPC (Dolan and Brockett, 2005), QQP (Iyer et al., 2017) and PAWS (Zhang et al., 2019) datasets.

5.3 Coverage Metric

Any section within a document could potentially contain possible rationale spans. A model trained on dialogues that cover larger portions of given documents should therefore perform better. Consequently, a strong data augmentation method should aim to generate dialogues that cover as much of the document as possible. We formalize this intuition with the span coverage metric, which we calculate as:

$$\text{Coverage} = \frac{\sum_{\text{span}} |\bigcup_{d \in \text{doc}_i} \bigcup_{s \in d} s|}{|\text{document}_i|}$$

where s refers to spans within a document and doc refers to the number of documents in the corpus.

5.4 Implementation Details

For passage ranker, and rationale extraction model, we fine-tuned RoBERTa-base (Liu et al., 2019) on the downstream training datasets. For utterance generators, we fine-tuned BART-base (Lewis et al., 2020b). We set total input length of 512-tokens which is 128 tokens for dialogue followed by 360 tokens for the document, with some room left over for special tokens. The augmented data is generated with sampling beam search with beam size 4, top-p 0.9, and temperature 0.9. When utilizing the augmented data, we pre-trained the downstream model on the augmented data for one epoch before fine-tuning (Alberti et al., 2019). The default F1 threshold is set to 0.9, which we determined by validating against the dev set. For fine-tuning, we train for five epochs, and use the same optimizer of AdamW (Loshchilov and Hutter, 2019) and learning rate of $3e{-}5$ for all experiments.

6 Results and Analysis

This section shows the results for the full dataset and low-resource settings. We also conduct human evaluation on the generated dialogues. Afterwards, we discuss the results by analyzing generated examples.

6.1 Main Results

As shown in Table 1, DG^2 achieves the overall best performance compared to other baselines that only augment the original human-annotated data. Other baselines all show some improvements over the downstream model only trained using the original data. EDA has very high EM and F1 scores for the rationale extraction task, but suffers at producing coherent dialogues as measured by BLEU. Paraphrase has relatively lower EM and F1 scores, but it achieves better BLEU scores than EDA and Back-translation. We suspect that this is because Paraphrase contains more diverse utterances as the inputs than other baselines.

When evaluating the augmented dialogues with

Model	25%			50%			75%		
	EM	F1	BLEU	EM	F1	BLEU	EM	F1	BLEU
Baseline	43.08	64.01	32.76	41.61	62.25	34.35	58.03	72.61	36.48
+ EDA	46.68	64.68	**33.97**	**56.09**	70.51	**35.84**	**59.84**	**73.40**	36.24
+ Back-translation	**47.48**	65.18	33.00	54.44	69.52	35.30	58.66	72.75	36.08
+ DG^2	46.48	**65.58**	32.90	54.51	**71.40**	35.74	58.89	73.38	**37.01**

Table 3: Experimental results on low-resource settings on validation set. **Bold** means the best score. Underline means the second best.

the original training set's documents, we find that DG^2 achieves higher span coverage. Unlike the other methods, DG^2 is able to generate novel rationales to increase the diversity of the augmented data, which we believe plays a large factor in improving downstream metrics.

Filtering	#Spans	EM	F1
None	-	57.78	73.27
F1 < 0.5	top-1	57.73	73.01
F1 < 0.9	top-10	58.23	73.05
F1 < 0.9	top-1	**60.80**	**74.38**
F1 < 0.95	top-1	59.21	74.00
F1 < 0.98	top-1	59.26	73.84

Table 4: We test different quality thresholds to determine the optimal level of filtering. A higher F1 score means that more samples are filtered.

6.2 Low Resource Setting

To further illustrate the performance of DG^2, we train all the models with only $25\%, 50\%, 75\%$ of the original training data. We generate the dialogues based on the documents in the knowledge base. In this limited data setting, our model generally outperformed Back-translation. However, compared to EDA, there is still some performance gap. We suspect that this is because when training with less data, the generative models' performance degenerates faster than the downstream model. We hope to overcome these issues with further improvements on data quality filtering.

6.3 Different Filtering Thresholds

Prior works in data augmentation have shown that filtering the synthetically generated examples can provide a meaningful boost in the data quality (Chen and Yu, 2021). As a result, we tune against different F1-score thresholds and span counts on the validation set. When the generated dialogue produces a higher F1-score, then this example is more likely to also produce better results during testing. The span count determines how many examples we consider when calculating this score. While raising the F1-score threshold increases the potential quality of the data, it comes as the expense of keeping fewer of the generated examples. Based on Table 4, we observe a sweet spot at 0.9, where a stricter filtering process would remove too many examples while a looser filtering process would lower the quality too much.

6.4 Human Evaluation

We conduct human evaluation on the human dialogues and the generated dialogues. We randomly sample 50 dialogues from each class. We shuffled the sampled dialogues and ask annotators to rate the dialogues with a score 1-5 in four different aspects: consistency, fluency, naturalness, and overall quality.

From the evaluation results, the generated dialogues show better fluency than original human dialogues with $p < 0.05$. We observed that some human dialogues contain typos and grammar errors, while the generated dialogues are more grammatically correct, which explains the score difference. In terms of other human evaluation metrics, the generated dialogues are still worse than the original human dialogues.

6.5 Qualitative Analysis

We now compare and contrast two examples generated by our procedure. Table 6 shows a good example from the document-grounded dialogue dataset. In the first four turns, the agent guides the user's focus by asking relevant questions. When the user wants to know more details, the agent then switches to provide the relevant knowledge retrieved from the rationale. This behavior is different from traditional question answering datasets where the agent simply reacts to user requests rather than exhibiting proactive behavior. On the flip side, one major

Speaker	Utterance
User	I'm thinking about getting some retirement benefits.
Agent	Are you wondering what age to start your benefits? *Rationale: What Is The Best Age To Start Your Benefits?*
User	Yes, I want to know what age I should start my benefits.
Agent	Are you planning to apply for Medicare? *Rationale: Should I apply for Medicare?*
User	Yes, I am.
Agent	Then you should know that Medicare usually starts when you reach age 65. *Rationale: Remember, Medicare usually starts when you reach age 65. If you decide to delay starting your benefits, be sure to contact Social Security about 3 months before you turn age 65 to check about applying for Medicare. Even if you have health insurance through a ...*
User	What if I need to enroll but don't?
Agent	Your Medicare coverage may be delayed and cost more. *Rationale: your Medicare coverage may be delayed and cost more. Please read the general and special enrollment period information on page 12 of our Medicare booklet to find out what may happen if you delay.*
User	What age is the best age to begin benefits?
Agent	The answer is that there is no one " best age" for everyone and, ultimately, it is your choice. You should make an informed decision about when to apply for benefits based on your individual and family circumstances. *Rationale: The answer is that there is no one " best age " for everyone and, ultimately, it is your choice. You should make an informed decision about when to apply for benefits based on your individual and family circumstances.*

Table 6: Generated dialogue showing typical agent and user interaction. Unlike question answering datasets, the agent can also ask questions to the user to guide the direction of the conversation.

	Consistency	Fluency	Naturalness	Overall
Human	3.80	3.96*	3.56	3.70
DG^2	3.60	4.18*	2.98	3.38

Table 7: Human evaluation results on the generated dialogues. * Comparison is made $p < 0.05$.

problem of the current approach is repetition. The user continues to ask about forgetting to update their address despite attempts by the agent to answer their query. Although the surface form of the user utterances are different, the semantic meaning remains the same. This repetition confuses the agent who then extracts irrelevant rationales, further exacerbating the situation.

7 Ethical Consideration

The models and approaches introduced in our work involve using synthetic data as an enhancement to existing datasets for modeling document-grounded dialogue. For the existing datasets, they are often dialogue simulation data generated by human workers based on their understanding of the associated document content and dialogue context. There are potential biases or toxic content introduced in the existing simulation during data collection. We can address such concerns by making efforts to improve the quality of the generated data that has shown its effectiveness in the downstream task. Therefore, our method can add an extra layer of safety and privacy if we only use generated data for training downstream models. Future work can explore how data augmentation can help to build a more private and safe dataset.

8 Conclusion

To address the problem of limited data in document-grounded dialogue systems, we propose DG^2 to perform data augmentation via dialogue generation. Our technique generates diverse utterances grounded on the given document while filtering the utterances to ensure quality and correctness when training on the downstream model. We demonstrated the effectiveness of our pipeline by showing the improvement over the previous data augmentation methods. We additionally show competitive results in the low-resource setting when a limited

amount of human annotated data is available for training. Future work will explore more techniques of filtering to improve data quality. We hope this spurs further research into document-grounded augmentation techniques for dialogue systems.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6168–6173. Association for Computational Linguistics.

Mihaela A. Bornea, Lin Pan, Sara Rosenthal, Radu Florian, and Avirup Sil. 2021. Multilingual transfer learning for QA using translation as data augmentation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 12583–12591. AAAI Press.

Ankit Chadha and Rewa Sood. 2019. BERTQA - attention on steroids. *CoRR*, abs/1912.10435.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 3002–3017. Association for Computational Linguistics.

Derek Chen and Zhou Yu. 2021. GOLD: improving out-of-scope detection in dialogues using data augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 429–442. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*. Asian Federation of Natural Language Processing.

Song Feng, Kshitij P. Fadnis, Q. Vera Liao, and Luis A. Lastras. 2020a. Doc2dial: A framework for dialogue composition grounded in documents. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 13604–13605. AAAI Press.

Song Feng, Hui Wan, R. Chulaka Gunasekara, Siva Sankalp Patel, Sachindra Joshi, and Luis A. Lastras. 2020b. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 8118–8128. Association for Computational Linguistics.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 639–649. Association for Computational Linguistics.

Jing Gu, Mostafa Mirshekari, Zhou Yu, and Aaron Sisto. 2021. Chaincqg: Flow-aware conversational question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 2061–2070. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. 2017. First quora dataset release: Question pairs. *Kaggle Competition*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020b. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Patrick S. H. Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *CoRR*, abs/2102.07033.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Longxuan Ma, Wei-Nan Zhang, Mingda Li, and Ting Liu. 2020. A survey of document grounded dialogue systems (DGDS). *CoRR*, abs/2004.13818.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 784–789. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering

challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 300–325. Association for Computational Linguistics.

Marzieh Saeidi, Max Bartolo, Patrick S. H. Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2087–2097. Association for Computational Linguistics.

Jason W. Wei and Kai Zou. 2019. EDA: easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 6381–6387. Association for Computational Linguistics.

Qingyang Wu, Yichi Zhang, Yu Li, and Zhou Yu. 2021. Alternating recurrent dialog model with large-scale pre-trained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 1292–1301. Association for Computational Linguistics.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2013–2018. The Association for Computational Linguistics.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020a. Intent detection with WikiHow. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 328–333, Suzhou, China. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 270–278. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1298–1308. Association for Computational Linguistics.

Victor Zhong, Mike Lewis, Sida I. Wang, and Luke Zettlemoyer. 2020. Grounded adaptation for zero-shot executable semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6869–6882. Association for Computational Linguistics.

A Training Details

	Hyper-parameters
batch size	16
warm-up steps	1,000
learning rate	3e-5
dropout	0.1
weight decay	0.01
max gradient norm	1.0
training epochs	5

Table 8: Training Details

We trained our model on and A4000 and A6000 GPUs. The detailed hyper-parameters can be found in the table above.

B Bad Generation Examples

We show a bad generation example here to illustrate one of the generation problem.

Speaker	Utterance
User	I was planning to move and forgot to update my address. What do I do?
Agent	By statute, you must report a change of address to DMV within ten days of moving. That is the case for the address associated with your license, as well as all the addresses associated with each registered vehicle, which may differ *Rationale: By statute, you must report a change of address to DMV within ten days of moving. That is the case for the address associated with your license, as well as all the addresses associated with each registered vehicle, which may differ.*
User	Am I able to do anything if I forget to update my address?
Agent	It is not sufficient to only: write your new address on the back of your old license; tell the United States Postal Service; or inform the police officer writing you a ticket. *Rationale: It is not sufficient to only: write your new address on the back of your old license; tell the United States Postal Service; or inform the police officer writing you a ticket.*
User	What happens if I forget to update my address?
Agent	You will miss a suspension order and may be charged with operating an unregistered vehicle and/or aggravated unlicensed operation, both misdemeanors. This really happens, but the good news is this is a problem that is easily avoidable. *Rationale: you will miss a suspension order and may be charged with operating an unregistered vehicle and/or aggravated unlicensed operation, both misdemeanors. This really happens, but the good news is this is a problem that is easily avoidable. Learn more ...*

Table 9: A bad generated dialogue here shows repetition. Although the user utterances' surface form are different, their semantic meaning is the same.

When can I Speak? Predicting initiation points for spoken dialogue agents

Siyan Li **Ashwin Paranjape** **Christopher D. Manning**
Stanford University
{siyanli, ashwinp, manning}@cs.stanford.edu

Abstract

Current spoken dialogue systems initiate their turns after a long period of silence (700-1000ms), which leads to little real-time feedback, sluggish responses, and an overall stilted conversational flow. Humans typically respond within 200ms and successfully predicting initiation points in advance would allow spoken dialogue agents to do the same. In this work, we predict the lead-time to initiation using prosodic features from a pre-trained speech representation model (wav2vec 1.0) operating on user audio and word features from a pre-trained language model (GPT-2) operating on incremental transcriptions. To evaluate errors, we propose two metrics w.r.t. predicted and true lead times. We train and evaluate the models on the Switchboard Corpus and find that our method outperforms features from prior work on both metrics and vastly outperforms the common approach of waiting for 700ms of silence.

1 Introduction

Spoken dialogue agents have exploded in popular use (e.g., Alexa, Siri, and Google Home). However, they only support explicit turn-taking mechanisms: they detect user initiation and barge-ins using wake-words and identify end of user turns based on a silence period (typically between 700–1000ms). Turn-taking feels unnatural under such mechanisms, leading to less "conversational" interactions (Woodruff and Aoki, 2003). This is particularly damaging for open-ended social conversations where thoughtful silences get wrongly interrupted (Chi et al., 2021). To fix this issue, we predict initiation opportunities for spoken dialogue agents for both turn-taking and backchanneling.

Prior work predicting initiation points uses prosodic features like pitch and frequency variation with bag-of-embeddings to predict backchannels (Ruede et al., 2017a) and turn-completion (Skantze, 2017), and more recently, Ekstedt and Skantze (2021) finetuned GPT-2 on dialogue datasets to predict turn-completion using only word features. However, they either predict a binary label indicating initiation in a wide event horizon, which is imprecise; or they predict a binary label for an initiation to happen at a set offset in the future, in which case a single incorrect prediction leads to a missed initiation.

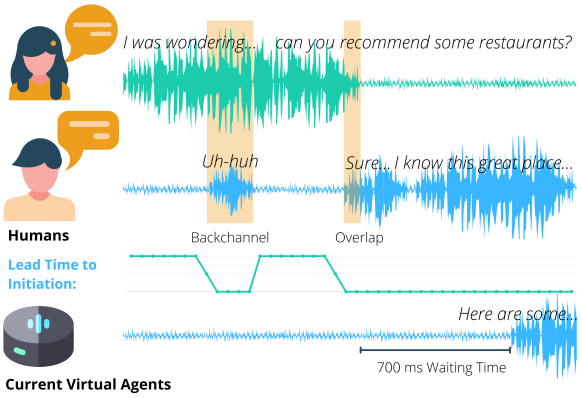


Figure 1: Humans produce overlapping speech with small gaps. By predicting lead to initiation, virtual agents can respond without long waiting periods

As a robust generalization of previous approaches, we predict the lead time to initiation as a continuous value. We model initiation (next utterance from a different speaker) directly and not end-of-turn because there is a variable (and possibly negative) gap between the two (Skantze, 2021). In this work, we combine two models: wav2vec 1.0 (Schneider et al., 2019) for representing prosodic features and finetuned GPT-2 (Radford et al., 2019) for word features. We model the task with a Gaussian Mixture Model (GMM) to account for inherent uncertainty. We train and evaluate our models on Switchboard (Godfrey et al., 1992) and find that the combination of the pretrained models performs the best, vastly outperforming a silence-based baseline that waits for 700ms of silence and baselines using features from prior work.

Proceedings of the SIGdial 2022 Conference, pages 217–224
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

2 Related Work

Prior work for dialogue turn-taking either uses silent gaps as cues or predicts future events repeatedly. A key issue with systems that use silent gaps as initiation cues (Huang et al., 2011; Cohen et al., 2004; Witt, 2015) is the difficulty of adjusting the silence thresholds to accommodate dialogue states (Skantze, 2021). When predicting turn-taking repeatedly, i.e. predicting future actions at every timestep, acoustic features such as pitch and frequency are often used, with additional linguistic features including part-of-speech or word embeddings (Ruede et al., 2017a,b; Skantze, 2017; Ward et al., 2018; Roddy et al., 2018). More recently, Ekstedt and Skantze (2021) implement a spoken dialogue system for travel conversations using TurnGPT (Ekstedt and Skantze, 2020). However, a short silence threshold is still used to determine initiation of agent responses.

Outside of dialogue, Neumann et al. (2019) propose probabilistic models for predicting events in videos, Lei et al. (2020) forecast frames and Vondrick et al. (2016) forecast actions. Time-to-event analysis in the medical domain involves modeling patient status as a function of time (Meira-Machado et al., 2009; Soleimani et al., 2017).

3 Methods

3.1 Setup

$\mathbf{I}^{\mathbf{k}}_{\text{spkr}}$ is the time of k-th initiation (both backchannels and transitions) by a speaker. We use the **current speaker**'s audio and transcript information to predict the the **lead time to initiation**, $\hat{\tau}_t$, of the **target speaker**. When the current speaker is speaking, we consider an **event horizon** δ_{max} to narrow the prediction range and at time t, define the true **lead time to initiation** as $\tau_t = \min(\delta_{max}, I^k_{\text{tgr}} - t)$. When the target speaker is speaking, we set $\tau_t = 0$, to ensure a well-balanced distribution.

3.2 Models

We make two novel contributions. First, we fuse rich contextual prosodic features from a pretrained wav2vec model with contextual word representations from a pretrained GPT-2 model. Prior work has not used such rich contextual prosodic features nor their combination with word representations. Second, prior work does not model the inherent uncertainty of initiations. Inspired by the video event prediction literature (Neumann et al., 2019), we do this using a Gaussian mixture model and maximize model likelihood under the data distribution.

3.2.1 Features

Features are extracted from the current speaker's voice channel and transcript. We suffix model names with abbreviated versions of the features they use.

Wav2vec Embeddings (W): Raw audio is fed into Wav2vec 1.0 (Schneider et al., 2019) to obtain convolutional embeddings. We choose Wav2vec 1.0 because of its unidirectional nature, which enables handling efficient incremental processing of audio. We keep the model weights frozen.

GPT-2 Embeddings (G): This is the GPT-2 Small (Radford et al., 2019) embedding of the last salient word from the target speaker after feeding in prior utterances. The embedding is updated incrementally as more utterances are transcribed. We fine-tune the GPT-2 model during training.

RMSE (R): We select the Root Mean Square Energy (RMSE) of the raw waveform to signal current speaker silence. It simulates audio energy and power in features from prior work.

Additional Prosodic Features (A): Previous work explores pre-neural prosodic features (Ruede et al., 2017a,b; Skantze, 2017); to compare our approach with previous approaches, we include pitch and frequency, both represented as a number for each frame. The prosodic features, including RMSE, are calculated with a frame shift of 50 ms and a window length of 100 ms. Additional details for feature implementation are in Appendix A.1.

Wav2vec features are subsampled to 50 ms by selecting embeddings at every 50ms and for other audio features by adjusting the frame shift. Audio features are concatenated and input to an LSTM network. When GPT-2 embeddings are used, they are concatenated with the LSTM's final hidden state. This is fed into a linear head. More training details are presented in Appendix A.2.

3.2.2 Gaussian Mixture Model

There is an inherent uncertainty in the precise location of an initiation (e.g., it can occur a few milliseconds before or after the prediction) and a single Gaussian is sufficiently powerful to model it because the uncertainty is localized. However, a speaker can initiate at many points in time that are far apart, for e.g., at the completions of grammatical clauses that can happen hundreds of milliseconds apart. We use a Gaussian mixture

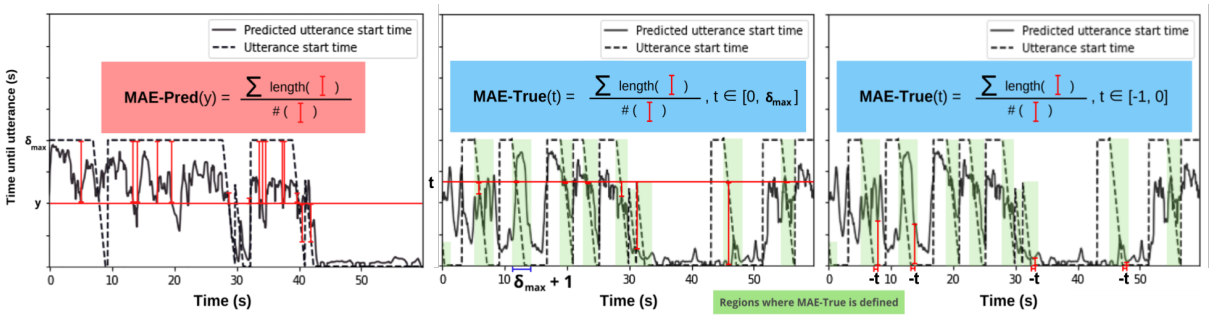


Figure 2: An explanation of our metrics. The red vertical intervals correspond to $|\tau_x - \hat{\tau}_x|$ in the equations. As illustrated, MAE-Pred(y) evaluates the expected error when a model predicts value y. For MAE-True(t), we highlight the regions where MAE-True can be calculated in green; depending on how long the current speaker's next utterance is, the region has a maximum length of $\delta_{max} + 1$.

model (GMM) to capture this multimodal prediction space.

At every time step, we predict the parameters: mean, variance and weights, for T Gaussian distributions $\{\mu, \sigma, h\}_{[1..T]}$. The training objective is to maximize the log of the summed likelihood of τ_t:

$$\log\Big(\sum_{i=1}^{T} h_i \cdot \frac{1}{\sigma_i\sqrt{2\pi}} \cdot \exp - \frac{(\tau_t - \mu_i)^2}{2\sigma_i^2}\Big)$$

At inference, we use the mean of the Gaussians.

3.2.3 Baselines

Silence Baseline: We compare our models with an RMSE-based non-neural baseline. We detect voice activity based on whether RMSE is above a certain threshold (0.01 for this work). If there is a gap of more than 700ms in voice-activity, the baseline predicts an initiation $\tau_t = 0$ at the current time, otherwise predicts δ_{max}.

GMM-AG: We use this baseline as a proxy for Ruede et al. (2017a), where pitch, power, and FFV are used as the prosodic features, and word2vec embedding of the most recent salient word is the linguistic feature. We simulate these features using RMSE, pitch and frequency (the prosodic features), and GPT-2 embeddings.

GMM-G: Ekstedt and Skantze (2020) use GPT-2 to emulate possible continuations of the current conversation in order to decide turn-relevant places. Although we do not use the same algorithm, we still use GPT-2 embedding as a feature. We train a GMM on last-salient-word GPT-2 embeddings only, and use this as a representative baseline for Ekstedt and Skantze (2020).

GMM-WGR-1: We train a Gaussian mixture model with $T = 1$ Gaussian to examine whether using multiple Gaussian models to capture different factors for utterance timing is necessary. This model is trained on the same data as our GMM-WGR model, with Wav2vec, GPT-2, and RMS features.

3.3 Training and Evaluation Data

For training, we randomly sample 60 second audio segments that have its first target speaker initiation in the first 5 to 10 seconds. This is to make sure that there is at least one initiation with enough context. We backpropagate losses only in a limited range around each initiation I^i_{tgt}, $[I^i_{\text{tgt}} - 2\delta_{max}, I^i_{\text{tgt}} + 1]$ This is to ensure a balanced distribution of τ_t. For evaluation and testing, we instead cover entire dialogues by collecting 60-second segments every 20 seconds. We randomly choose the target speaker for each segment.

3.4 Metrics

To measure the performance of our models that produce continuous values, previous work's classification-based metrics are insufficient to differentiate between a prediction error of 0.2 versus 2 seconds. Additionally, we want to differentiate between how precise model predictions are and how well they cover the initiations observed in the dataset. We improve upon Time-to-event error from Neumann et al. (2019), and propose Mean Absolute Error w.r.t. Predicted Lead Time (MAE-Pred) and Mean Absolute Error w.r.t. True Lead Time (MAE-True) as analogues of precision and recall that improve existing metrics (Skantze, 2017). If a practitioner needs l seconds to generate a response, MAE-Pred(l) gives the expected error when the model predicts l (precision) and MAE-True(l) gives the expected error with the true lead

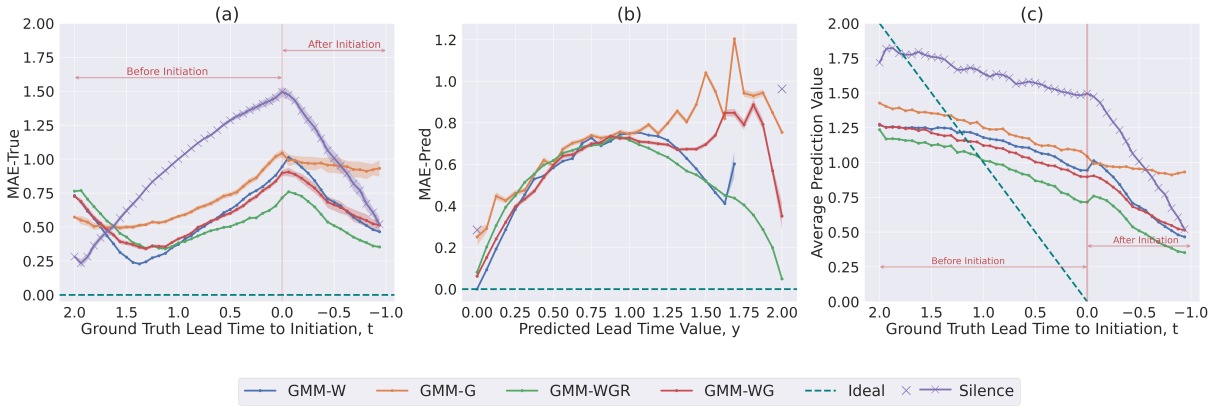


Figure 3: (a) MAE-True, (b) MAE-Pred, and (c) average predicted lead time values for representative neural models and the silence baseline. 95% C.I. are represented by the lightly shaded regions. A perfect model would achieve the "ideal" (dashed) lines. In (b), because the silence based model only predicts 0 or δ_{max}, only these two points are defined in plot (b) for the silence based baseline. The corresponding MAE-Pred values for the silence baseline are indicated as crosses in plot(b). All of our models, including the best performing GMM-WGR, significantly outperform the silence-based model that waits for 700 ms.

time is l (recall). With the set S representing the timesteps included in the calculations, both metrics can be represented as

$$\sum_{x \in S} |\tau_x - \hat{\tau}_x| / |S|$$

Specifically, for **MAE-Pred**(y):

$$S = \{x | \hat{\tau}_x = y\}, y \in [0, \delta_{max}]$$

For **MAE-True**(t):

$$S = \{I^i_{\text{tgt}} - t\}, t \in [-1, \delta_{max}] \cap [I^i_{\text{tgt}} - I^{j+1}_{\text{cur}}, I^i_{\text{tgt}} - I^j_{\text{cur}}]$$

for all target-speaker initiations I^i_{tgt}, limiting to intervals between two consecutive initiations by the current speaker. When $t \leq 0$, the initiation has already occurred and $\tau_t = 0$. We quantize both true and predicted values into 16 buckets per second.

As an aggregated metric, we propose Macro-MAE (MMAE). We define MMAE-X$(a, b) = \sum_{v \in S_{ab}}$MAE-X$(v)/|S_{ab}|$, where S_{ab} is the set of bucket values between a and b for a given set S. We define 1 second before and $0.5s$ after initiation as the interval of interest for MMAE-True, and similarly predicted values between 0 and 1 for MMAE-Pred. We compute MMAE = MMAE-True$(-0.5, 1)$ + MMAE-Pred$(0, 1)$ as a single number quantifying model performance.

4 Experiments

For training and evaluation, we use audio conversations from Switchboard (Godfrey et al., 1992). We select a random set of 200 training, 20 validation, and 20 test dialogues out of a total of 1000 dialogues due to computational constraints. We use the validation set to select the best performing checkpoint based on MMAE scores and report the numbers on the test set. For the GMM models, we experimented with $T = 1, 5, 10, 15, 20$, and found $T = 15$ to be the best-performing. [1]

We plot the MAE-Pred and MAE-True values in Figure 3 and the show the MMAE values in Table 1. A perfect model would have 0 error. As a diagnostic tool, we also plot the average prediction for each t used in MAE-True (Figure 3 (c)). Here, we expect a perfect model to be a line with a slope of -1 passing through the origin before flattening out at 0. We see that for all models MAE-True peaks (roughly) at initiation (Figure 3 (b))). Despite all the cues leading up to an initiation in the data, it is still highly optional and the models aren't able to predict it perfectly. Soon afterward, as the target speaker stays silent the models predict smaller lead times to initiation (steeper downward slope in Figure 3 (c)) and the MAE-True reduces. On the other hand, for all trained models (GMM-*), we see that MAE-Pred reduces for smaller values of y (Figure 3 (c)) indicating that the trained models are very precise when they predict near-term initiations.

Our models outperform the silence baseline by a large margin in most time windows prior to and

[1] Our code for the models and for training is available at https://github.com/siyan-sylvia-li/icarus_final

	Eval			Test
Model	MT	MP	MMAE	MMAE
GMM-AG	0.90	0.63	1.53	1.51
GMM-G	0.90	0.60	1.50	1.42
GMM-WGR-1	0.67	0.59	1.26	1.30
Silence*	1.33	0.60	1.93	1.88
GMM-W	0.70	**0.49**	1.19	1.22
GMM-WG	0.67	0.51	1.18	1.19
GMM-WGR	**0.63**	0.52	**1.15**	**1.11**

Table 1: Performance of different models on the evaluation and the test dialogues, as measured Macro-MAE values. MT = MMAE-True$(-0.5, 1)$, MP = MMAE-Pred$(0, 1)$. * Since only 0 and δ_{max} are valid predictions for Silence Baseline, we use (MAE-Pred(0) + MAE-Pred(δ_{max}))/2 as MMAE-Pred$(0, 1)$.

after initiations (Figure 3 and Table 1). GMM-WGR outperforms prior work baselines: GMM-G (TurnGPT) and GMM-AG (Ruede et al. (2017a)).

Comparing GMM-WG vs. GMM-G, Wav2vec features reduce MAE-True after initiation and stabilizes MAE-Pred for small predicted lead times; GMM-G's predictions stay constant after initiations, because it can only access the transcript from the current speaker. Comparing GMM-WG vs. GMM-W, GPT-2 features reduce MAE-True near initiations, possibly because they provide the model with word cues. GMM-WGR has a lower MMAE-True(-0.5, 1) compared to GMM-WG, indicating that Wav2vec doesn't capture silences as well as RMSE. GMM-WGR-1, our baseline with one Gaussian, performs poorly compared to GMM-WGR, highlighting the importance of the Gaussian mixture.

5 Conclusion

We present the task of lead time to initiation prediction as a continuous-valued problem, collapsing transition and backchannel timing problems into one. We additionally propose metrics to capture precision and coverage in these predictions. Our models trained on pretrained prosodic and verbal embeddings consistently outperform the commonly-used silence baseline. We believe our work will build a foundation for more naturalistic virtual agents with human-like conversational behaviors.

References

Ethan A Chi, Caleb Chiam, Trenton Chang, Swee Kiat Lim, Chetanya Rastogi, Alexander Iyabor, Yutong He, Hari Sowrirajan, Avanika Narayan, Jillian Tang, et al. 2021. Neural, neural everywhere: Controlled generation meets scaffolded, structured dialogue. *Alexa Prize Proceedings*.

Michael H Cohen, Michael Harris Cohen, James P Giangola, and Jennifer Balogh. 2004. *Voice user interface design*. Addison-Wesley Professional.

Erik Ekstedt and Gabriel Skantze. 2020. TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2981–2990, Online. Association for Computational Linguistics.

Erik Ekstedt and Gabriel Skantze. 2021. Projection of turn completion in incremental spoken dialogue systems. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 431–437, Singapore and Online. Association for Computational Linguistics.

J.J. Godfrey, E.C. Holliman, and J. McDaniel. 1992. Switchboard: telephone speech corpus for research and development. In *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 517–520 vol.1.

Lixing Huang, Louis-Philippe Morency, and Jonathan Gratch. 2011. Virtual rapport 2.0. In *International workshop on intelligent virtual agents*, pages 68–79. Springer.

Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. 2020. What is more likely to happen next? video-and-language future event prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8769–8784, Online. Association for Computational Linguistics.

Luís Meira-Machado, Jacobo de Uña-Álvarez, Carmen Cadarso-Suárez, and Per K Andersen. 2009. Multi-state models for the analysis of time-to-event data. *Statistical Methods in Medical Research*, 18(2):195–222. PMID: 18562394.

Lukáš Neumann, Andrew Zisserman, and Andrea Vedaldi. 2019. Future event prediction: If and when. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2935–2943.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Matthew Roddy, Gabriel Skantze, and Naomi Harte. 2018. Investigating speech features for continuous turn-taking prediction using LSTMs. *arXiv preprint arXiv:1806.11461*.

Robin Ruede, Markus Müller, Sebastian Stüker, and Alex Waibel. 2017a. Enhancing backchannel prediction using word embeddings. In *Interspeech*, pages 879–883.

Robin Ruede, Markus Müller, Sebastian Stüker, and Alex Waibel. 2017b. Yeah, right, uh-huh: A deep learning backchannel predictor. *CoRR*, abs/1706.01340.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Gabriel Skantze. 2017. Towards a general, continuous model of turn-taking in spoken dialogue using LSTM recurrent neural networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 220–230, Saarbrücken, Germany. Association for Computational Linguistics.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: a review. *Computer Speech & Language*, 67:101178.

Hossein Soleimani, James Hensman, and Suchi Saria. 2017. Scalable joint models for reliable uncertainty-aware event prediction. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1948–1963.

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Anticipating visual representations from unlabeled video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 98–106.

Nigel G Ward, Diego Aguirre, Gerardo Cervantes, and Olac Fuentes. 2018. Turn-taking predictions across languages and genres using an LSTM recurrent neural network. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 831–837. IEEE.

Silke Witt. 2015. Modeling user response timings in spoken dialog systems. *International Journal of Speech Technology*, 18(2):231–243.

Allison Woodruff and Paul M Aoki. 2003. How push-to-talk makes talk less pushy. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 170–179.

A Appendix

A.1 Feature Implementation

1. Pitch: https://pytorch.org/audio/main/functional.html#compute-kaldi-pitch

2. Frequency: https://librosa.org/doc/main/generated/librosa.yin.html

3. Root Mean Square Energy: https://librosa.org/doc/main/generated/librosa.feature.rms.html

A.2 Training Details

The models are trained on one A100 GPU. All model LSTM's have two layers with 128 hidden units. Each epoch approximately last 1000 seconds, and we train each neural model for 7 epochs, at which point overfitting would have definitely occurred. We train all models with dropout 0.1, Adam optimizer, and a weight decay of 0.0001. We include a comprehensive list of our models and their training details in Table 4.

A.3 Additional Model: Heuristic Heatmap

We have tried training another probabilistic model from Neumann et al. (2019), Heuristic Heatmap. We did not find this model to significantly outperform our GMM-Full model, although it does exhibit interesting qualities.

Heuristic Heatmap (Histogram-based Density Estimator): This model captures temporal shifts in the probability distribution of lead time; as the current speaker keeps speaking, the likelihood of an imminent initiation increases for the target speaker, shifting the probability mass from higher to lower lead time values. At every time step, the model produces a probability distribution with $2\delta_{max}r$ ($r = 16$, the resolution of our estimates) bucket values $h_i = P(\tau_t = \frac{2\delta_{max}i}{2\delta_{max}r})$. Training minimizes the difference between the predicted distribution and a Gaussian centered at τ_t. During inference, the prediction bucket with the highest probability is returned.

Model	W	G	Ac	R
GMM-AG		✓	✓	✓
GMM-G		✓		
GMM-W	✓			
GMM-WG	✓	✓		
GMM-WGR	✓	✓		✓
Heatmap-WGR	✓	✓		✓
GMM-WGR-1	✓	✓		✓

Table 2: The trained models and their features. **W** represents Wav2vec features, **G** GPT-2 embeddings, **Ac** the set of acoustic features (pitch and frequency), **R** the RMSE of the current speaker waveform.

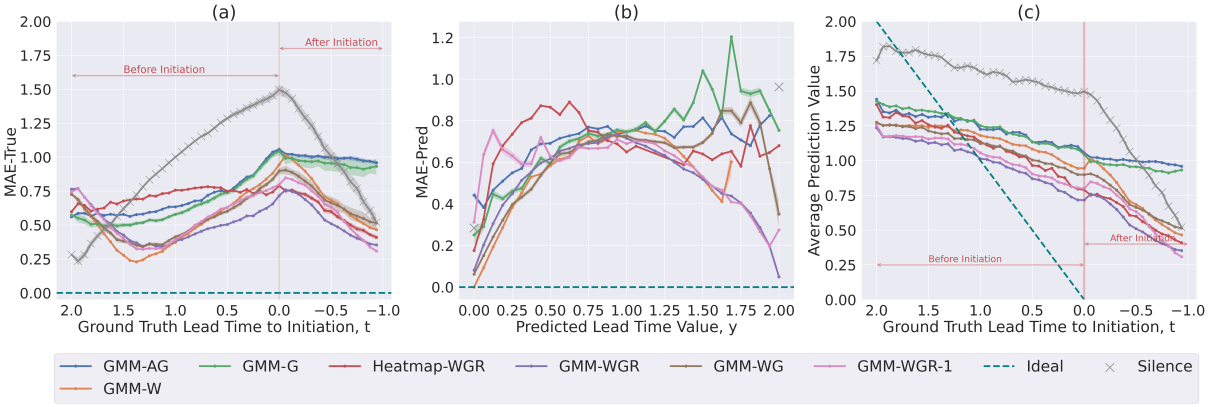


Figure 4: MAE-True, MAE-Pred graphs for all trained models. We also include the graph of average predicted lead time values given true lead time to initiation.

Model	$\mathbf{MT}_{\mathbf{Eval}}$	$\mathbf{MP}_{\mathbf{Eval}}$	$\sum_{\mathbf{Eval}}$	$\sum_{\mathbf{Test}}$
GMM-AG	0.90	0.63	1.53	1.51
GMM-G	0.84	0.58	1.50	1.42
GMM-W	0.70	0.49	1.19	1.22
GMM-WG	0.67	0.51	1.18	1.19
GMM-WGR	0.63	0.52	1.15	1.11
Heatmap-WGR	0.80	0.68	1.48	1.44
GMM-WGR-1	0.67	0.59	1.26	1.30
Silence*	1.33	0.60	1.93	1.88

Table 3: Performance of different models on the evaluation and the test dialogues, as measured by the sum of (1) the average MAE-True(t) on $t \in [1, -0.5]$ ($\mathbf{MT}_{\mathbf{Eval}}$ and $\mathbf{MT}_{\mathbf{Test}}$) and (2) the average MAE-Pred(y) on $y \in [0, 1]$ ($\mathbf{MP}_{\mathbf{Eval}}$ and $\mathbf{MP}_{\mathbf{Test}}$). * For the Silence baseline, since only 0 and δ_{max} are valid prediction values, we calculate the average of MAE-Pred(0) and MAE-Pred(δ_{max}) as $\mathbf{MP}_{\mathbf{Eval}}$ and $\mathbf{MP}_{\mathbf{Test}}$.

A.4 MAE-True and MAE-Pred on All Models

We also include the graphs for MAE-True, MAE-Pred, and average predictions per ground truth time to initiation values for all of our models. They are presented in Figure 4.

Model	Features	Learning Rate	Batch Size
GMM-AG	Acoustic features, GPT-2	1e−4	16
GMM-G	GPT-2 embedding	1e−4	16
GMM-W	Wav2vec representations	1e−4	32
GMM-WG	Wav2vec and GPT-2	1e−5	16
GMM-WGR	Wav2vec, GPT-2, and RMSE	1e−5	32
GMM-WGR-1	Wav2vec, GPT-2, and RMSE	1e−5	15
Heatmap-WGR	Wav2vec, GPT-2, and RMSE	1e−4	32

Table 4: Set of trained models.

Using Interaction Style Dimensions to Characterize Spoken Dialog Corpora

Nigel G. Ward
Computer Science, University of Texas at El Paso
500 West University Avenue, El Paso, Texas 79968, USA
nigelward@acm.org

Abstract

The construction of spoken dialog systems today relies heavily on appropriate corpora, but corpus selection is more an art than a science. As interaction style properties govern many aspects of dialog, they have the potential to be useful for relating and comparing corpora. This paper overviews a recently-developed model of interaction styles and shows how it can be used to identify relevant corpus differences, estimate corpus similarity, and flag likely outlier dialogs.

1 Motivation

Today the process of selecting corpora for dialog systems training or tuning is rarely systematic. This is a problem because dialog systems developers rely heavily on machine learning from corpora to acquire the various knowledge and parameters needed for effective systems. Models for predicting likely corpus suitability would therefore be very useful, but existing methods for corpus comparison rely mostly on lexical and topic overlap, e.g. (Pavlick and Nenkova, 2015), making it hard to predict how well other types of knowledge will transfer.

The scientific investigation of dialog behaviors is similarly impeded by corpus choice issues. Different research teams choose corpora to study for all sorts of reasons, leading to a healthy diversity, but also to many contradictory findings (Egger et al., 2014; Wright et al., 2019; Levitan, 2020). Methods for systematically describing corpus properties could help resolve these, potentially enabling the field of computational pragmatics to clearly describe the realm of validity of each generalization.

This paper focuses on interaction style, as this is an essential issue in providing high quality user experiences (Marge et al., 2022). This is, moreover, no longer a distant goal, as core speech components have advanced to the point where it is becoming possible to implement situation-appropriate turn taking, politeness behaviors, rapport building strategies, and so on (Metcalf et al., 2019). Because our fundamental knowledge in these areas are still spotty, developers rely on discovery or learning from corpora. Indeed, it is still common for a new development project to start with the collection of a new corpus, specific to the task, domain, user demographic, system persona and so on. Instead, we would like to be able to better exploit existing resources (Kashyap et al., 2021). One recent success was a socially well-behaved recommendation system for movies, created by discovering behaviors from a suitable subset of Switchboard data (Pecune et al., 2019). Selection of this subset was easy because Switchboard was designed around topics, and in particular the "movies" tag was available. However, we would like to be able to more precisely delineate relevant corpus subsets, and to do so even when annotations are lacking.

This paper introduces three ways to characterize spoken dialog corpora and their subsets.

2 Precursor Work

Biber, in his landmark contribution to style description, investigated what he termed "conversation text types" (Biber, 2004). Using transcripts from various corpora as data and a text-based feature set, he used Principal Component Analysis to derive three dimensions of variation, and showed how different conversations could be automatically located in this space.

This method has been very influential in the comparison of diverse text corpora, and also occasionally for speech corpora (Shen and Kikuchi, 2014). However these models generally seem to have low explanatory power; for example, Biber's three dimensions accounted for only 36% of the variance. Further, although acoustic-prosodic features potentially provide much more information than text, these have been used in corpus selection so far only by Siegert *et al.* (2018), who demonstrated their

Proceedings of the SIGdial 2022 Conference, pages 225–230
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

value, but only for the narrow problem of training emotion recognizers. Overall, work in this tradition appears not to have found practical use.

In contrast to text-based models (Troiano et al., 2021, submitted), styles in spoken dialog, and in particular interaction styles, have been less studied. Much work in this area has built on Tannen's seminal observations on "conversational styles" (Tannen, 1989, 1980). Importantly, these are not fixed properties of speakers, and frequently vary even in the course of a conversation (Dingemanse and Liesenfeld, 2022).

More recently, computational models have been developed to study style in dialog (Grothendieck et al., 2011; Laskowski, 2016; Yamamoto et al., 2020; Ward, 2021a). These works have variously used features of turn-taking and prosodic and other behaviors to derive models of style. However these models have previously been applied only to questions of how individuals vary in style, not to corpus characterization.

3 Model Properties

The explorations reported in this paper build on our own model of interaction style variation (Ward, 2021a; Ward and Avlia, 2022, submitted), because it is the most comprehensive and because the code is available. The purpose of this section is only to explain the model briefly while clarifying the aspects not clear in (Ward, 2021a) but relevant for the current exploration.

For current purposes, the model serves to take as input one or more 30-second fragments of American English conversation, and to output a representation of its style as a vector of length 8: that is, it maps dialogs into a vector space representation of interaction styles. While for current purposes this is used as a black box, it may be worth overviewing the steps of the process.

1. Low-level (frame level) prosodic features are computed, specifically the raw pitch, intensity, and cepstral coefficients.
2. These are normalized by track.
3. Filters and aggregation processes are applied to obtain mid-level features over various temporal spans, including estimates of intensity, speaking rate, phoneme lengthening, creakiness, enunciation or reduction, and the extent to which the pitch is high or low, or wide or narrow.
4. These mid-level features are normalized using parameters that brought each to mean 0 and standard deviation 1 on the training data.
5. The match of these normalized features to 12 meaningful temporal configurations is computed every 20 milliseconds. These meaningful temporal configurations represent specific American English prosodic constructions, which mark activities such as turn switch, topic closing, enthusiasm, positive assessment, empathizing, and contrasting (Ward, 2019). These cover a wide range of dialog states, activities, behaviors and interactive events.
6. The match values are binned and pooled across each 30-second fragment. There are 7 bins per configuration, thus there are bins for when a speaker is expressing a strong, mild, or weak contrast, or managing an ambiguous, clear, or strong turn switch, and so on.
7. The resulting 84 values are rotated, using Principal Component Analysis, to a representation where the top dimensions capture most of the variance.
8. The top 8 dimensions are retained. (This is because these 8 already explain 52% of the variance, because the lower dimensions lacked clear interpretations, and because including more dimensions did not significantly change the qualitative picture presented below.)

Further, each of the eight dimensions can be given an interpretation, as summarized in Table 1. Those for Dimensions 4 and 7 differ from those given by Ward (2021a), for reasons explained in (Ward and Avlia, 2022, submitted); for all, we here provide clearer descriptions. While the interpretations are not needed for most purposes, they help to understand how and whether the model is working, so the rest of this section elaborates. Evidence and further discussion appears at the companion website (Ward, 2021b).

Dimension 1 relates simply to the amount of shared engagement. Dimension 2 is very high or low when one speaker *versus* the other is taking an active speaking role and the other an active listening role. Dimension 3 involves expressing positive assessment, for example when talking about the speaker's dog, a good fishing day, or a favorite football team, *versus* expressing negative feelings,

1	13%	both participants engaged	...	lack of shared engagement
2	11%	focal speaker mostly talking	...	focal speaker listening actively
3	8%	positive assessment	...	negative feelings
4	5%	focal speaker speaks knowledgeably	...	nonfocal speaker speaks knowledgeably
5	5%	factual	...	thoughtful
6	4%	accepting things beyond individual control	...	envisioning positive change
7	3%	making points	...	referencing shared experiences
8	3%	unfussed	...	emphatic

Table 1: Inferred functions of the top 8 dimensions of interaction style. The second column shows the amount of variance explained by each dimension.

for example about underprepared students or immoral politicians. Dimension 4 is very high or low when one speaker *versus* the other is being confident and/or dominant as they talk about something they know well, while the other is acknowledging the other as an expert on the topic. For Dimension 5 the positive pole involves a thoughtful style and the negative pole a factual style, characterized, among other things, by long regions of low pitch expressing a stance of calm rationality, as the speaker describes something they know well, such as how a network is set up or how security cameras work. Dimension 6 relates to a resigned attitude, for example when taking about high rents or working in a job where there is no opportunity to meet the customers, *versus* a positive, change-oriented outlook, for example when discussing new exercise regimens, changes in women's roles, or medical research advances. Dimension 7 relates to stating and justifying opinions, for example general ideas about dealing with people or situations, *versus* finding common ground, for example when talking about similar experiences with catalog shopping, making hamburger, or drug testing. Dimension 8 involves the continuum between talk about remote or currently unimportant and half-understood or half-remembered ideas or events *versus* expressing strong opinions, for example regarding people or practices that are strongly disliked or strongly admired.

4 Use 1: Corpus Characterization

This model supports visualization of corpus differences. As an example, if we view Switchboard (Godfrey et al., 1992) as a collection of subcorpora, one per topic, we can map them out, for example by plotting the average interaction style of all fragments within that topic. Figure 1 shows this for Dimensions 1 and 3. (Projections onto other dimensions are available at the companion website.) To avoid clutter, the figure show only topics for which there was ample data (225 minutes or more) or which were among the most distinctive topics, in terms of distance on these two dimensions from the global average style. Table 2 shows the values for all 8 dimensions for the topics discussed below.

The positions of the topics in the figure suggest that the model is at least picking up something meaningful. It is informative to consider further some of the topics that appear, at first glance, to be misplaced. For example, it may seem strange that the model characterizes conversations on the topic of "metric system" as positive in style, but listening to examples shows that these conversations are mostly by engineers, who indeed discussed it positively. It may also seem strange that "woodworking" and "painting" are placed differently, as both can be at-home hobbies and projects. According to the model, their interaction styles are very different, as seen in Table 2. In particular, these suggest that dialogs about woodworking exhibited less shared engagement and were more positive and thoughtful in tone (Dimensions 1, 3, and 5, respectively). Listening confirmed that these differences were real, and likely attributable to the tendencies for woodworking to be discussed fondly by dedicated hobbyists, and painting to be discussed by novices talking about difficulties. Thus the model captures much more than simple topic similarity.

In general, diagrams like these may help researchers and developers understand the diversity within and between corpora.

5 Use 2: Similarity Estimation

This model also supports similarity estimation (Kilgarriff and Rose, 1998), for now by simply us-

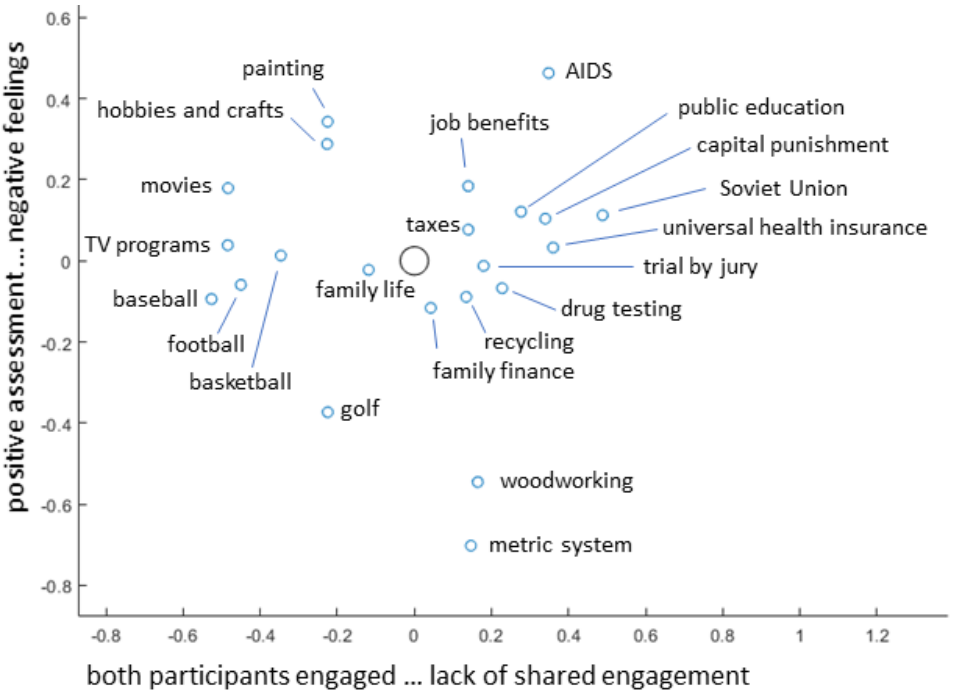


Figure 1: Average Interaction Styles of Some Topics in Switchboard, Projected to Interaction Style Dimensions 1 and 3. The large circle marks (0,0), the global average style. The axis units are standard deviations computed over all conversation fragments. The topic names shown are just mnemonics for the sentence-length prompts given to the participants.

	dimension							
	1	2	3	4	5	6	7	8
woodworking	0.6	2.2	–1.4	1.4	1.1	–0.4	0.6	0.5
painting	–0.8	3.0	0.9	1.8	–0.6	0.1	–0.6	0.5
politics	1.0	2.6	0.1	1.6	0.4	0.1	0.1	–0.2
capital punishment	1.1	2.7	0.3	1.6	0.5	0.0	0.0	–0.0
movies	–1.6	–0.0	0.5	0.0	–0.7	–0.5	0.3	–0.1

Table 2: Average interaction style for selected topics from Switchboard on the 8 dimensions.

ing the Euclidean distance in the 8-dimensional space. For example, considering Switchboard's 20 topics most distant from the global average, the closest pair was "politics" and "capital punishment," as seen in Table 2 respectively. The other most similar pairs were "baseball" and "football," "weather/climate" and "vacation spots," and "movies" and "TV programs."

Such similarity estimates could be used to support targeted data augmentation. Considering again the scenario of seeking data to train a movie recommendation system, the subcorpora closest to "movies" were "TV programs," "clothing and dress," "football," and "baseball," indicating that these would be likely be most compatible as supplementary data.

6 Use 3: Identifying Outliers

The similarity metric could also be used in support of data cleaning. For example, many conversations in Switchboard have the "movies" tag, but not all fragments are good exemplars of the typical style for talking about movies. The model can help identify these, as fragments distant from the average interaction style for this topic. For the movies topic, examination of the five most distant fragments revealed that these were indeed mostly atypical — two involved strong moral judgments, and one was mostly about audience behavior — and would be good candidates for exclusion from the training set

for a movie recommending system with a normal, upbeat style.

7 Prospects

Spoken data is fundamentally richer than text data, and recent work is exploiting this to create more informative models of corpus similarities and differences. This brief report has proposed new ways to exploit one such model, involving interaction style.

Eventually, direct quantitative evaluation of this method should be done. One way would be to examine the correspondences to human judgments of interaction styles and style similarities. This would be a long-term project, but potentially of great benefit for systematizing the scientific study of dialog phenomena.

In the short term, we think the value of these methods will instead be shown by their practical value: their ability to support the creation of better-tailored dialog systems, and to reduce the data-collection efforts required to develop them. More specifically, in addition to the three ways illustrated above, we conjecture that the model will be useful in at least three other ways. 1) It could support quality control and consistency control during corpus collection. 2) It could support attempts to collect corpora with a sweet-spot style that is simultaneously natural for humans and implementable with current technology (Budzianowski et al., 2018; Byrne et al., 2019), by identifying the dimensions in which such corpora most resemble both human-human dialogs and technically-realizable dialogs. 3) It could support the development of widely useful pretrained models by supporting the selection of truly diverse sets of dialog corpora.

To support such uses, the code is available at (Ward, 2021c).

Acknowledgments: I thank Jonathan E. Avila for helping refine the dimension interpretations.

References

Douglas Biber. 2004. Conversation text types: A multi-dimensional analysis. In *Le poids des mots: Proceedings of the 7th International Conference on the Statistical Analysis of Textual Data*, pages 15–34. Presses Universitaires de Louvain.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz: a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Empirical Methods in Natural Language Processing*, pages 5016 – 5012.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Empirical Methods in Natural Language Processing*, page 4515–452.

Mark Dingemanse and Andreas Liesenfeld. 2022. From text to talk: Harnessing conversational corpora for humane and diversity-aware language technology. In *ACL*, pages 5614–5633.

Sebastian Egger, Peter Reichl, and Katrin Schoenenberg. 2014. Quality of experience and interactivity. In Sebastian Moeller and Alexander Raake, editors, *Quality of Experience*, pages 149–161. Springer.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of ICASSP*, pages 517–520.

John Grothendieck, Allen L. Gorin, and Nash M. Borges. 2011. Social correlates of turn-taking style. *Computer Speech and Language*, 25:789–801.

Abhinav Ramesh Kashyap, Devamanyu Hazarika, Min-Yen Kan, and Roger Zimmermann. 2021. Domain divergences: a survey and empirical analysis. In *NAACL*, pages 1830–1849.

Adam Kilgarriff and Tony Rose. 1998. Measures for corpus similarity and homogeneity. In *Proceedings of the Third Conference on Empirical Methods for Natural Language Processing*, pages 46–52.

Kornel Laskowski. 2016. A framework for the automatic inference of stochastic turn-taking styles. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 202–211.

Rivka Levitan. 2020. Developing an integrated model of speech entrainment. In *IJCAI*, pages 5159 – 5163.

Matthew Marge, Carol Espy-Wilson, Nigel G. Ward, et al. 2022. Spoken language interaction with robots: Research issues and recommendations. *Computer Speech and Language*, 71.

Katherine Metcalf, Barry-John Theobald, Garrett Weinberg, Robert Lee, Ing-Marie Jonsson, Russ Webb, and Nicholas Apostoloff. 2019. Mirroring to build trust in digital assistants. In *Interspeech*, pages 4000–4004.

Ellie Pavlick and Ani Nenkova. 2015. Inducing lexical style properties for paraphrase and genre differentiation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 218–224.

Florian Pecune, Shruti Murali, Vivian Tsai, Yoichi Matsuyama, and Justine Cassell. 2019. A model of social explanations for a conversational movie recommendation system. In *Proceedings of the 7th International Conference on Human-Agent Interaction*, pages 135–143.

Raymond Shen and Hideaki Kikuchi. 2014. Estimation of speaking style in speech corpora focusing on speech transcriptions. In *LREC*, pages 2747–2752.

Ingo Siegert, Ronald Böck, and Andreas Wendemuth. 2018. Using a PCA-based dataset similarity measure to improve cross-corpus emotion recognition. *Computer Speech & Language*, 51:1–23.

Deborah Tannen. 1980. The parameters of conversational style. In *18th Annual Meeting of the Association for Computational Linguistics*, pages 39–40.

Deborah Tannen. 1989. *That's Not What I Meant! How Conversational Style Makes or Breaks Relationships*. Ballantine.

Enrica Troiano, Aswathy Velutharambath, et al. 2021, submitted. From theories on styles to their transfer in text: Bridging the gap with a hierarchical survey. *Natural Language Engineering*.

Nigel G. Ward. 2019. *Prosodic Pattterns in English Conversation*. Cambridge University Press.

Nigel G. Ward. 2021a. Individual interaction styles: Evidence from a spoken chat corpus. In *SigDial*, pages 13–20.

Nigel G. Ward. 2021b. Interaction style variation: Companion website. Http://www.cs.utep.edu/nigel/istyles/.

Nigel G. Ward. 2021c. Interaction styles tools (2019-2022). https://github.com/nigelg ward/istyles.

Nigel G. Ward and Jonathan E. Avlia. 2022, submitted. A dimensional model of interaction style variation in spoken dialog. *Speech Communication*.

Richard Wright, Courtney Mansfield, and Laura Panfili. 2019. Voice quality types and uses in North American English. *Anglophonia*.

Kenta Yamamoto, Koji Inoue, Shizuka Nakamura, Katsuya Takanashi, and Tatsuya Kawahara. 2020. A character expression model affecting spoken dialogue behaviors. In *Proceedings of the International Workshop on Spoken Dialog System Technology*.

Multi-Domain Dialogue State Tracking with Top-K Slot Self Attention

Longfei Yang[1], **Jiyi Li**[2], **Sheng Li**[3], **Takahiro Shinozaki**[1]
[1]Tokyo Institute of Technology
[2] University of Yamanashi
[3] National Institute of Information and Communications Technology
longfei.yang.cs@gmail.com, jyli@yamanashi.ac.jp, sheng.li@nict.go.jp,
shinot@ict.e.titech.ac.jp

Abstract

As an important component of task-oriented dialogue systems, dialogue state tracking is designed to track the dialogue state through the conversations between users and systems. Multi-domain dialogue state tracking is a challenging task, in which the correlation among different domains and slots needs to consider. Recently, slot self-attention is proposed to provide a data-driven manner to handle it. However, a full-support slot self-attention may involve redundant information interchange. In this paper, we propose a top-k attention-based slot self-attention for multi-domain dialogue state tracking. In the slot self-attention layers, we force each slot to involve information from the other k prominent slots and mask the rest out. The experimental results on two mainstream multi-domain task-oriented dialogue datasets, MultiWOZ 2.0 and MultiWOZ 2.4, present that our proposed approach is effective to improve the performance of multi-domain dialogue state tracking. We also find that the best result is obtained when each slot interchanges information with only a few slots.

1 Introduction

As a crucial component of task-oriented dialogue systems, dialogue state tracking (DST) is designed to track the dialogue states through the conversations between users and systems (Young et al., 2010, 2013), which is generally expressed as a list of {(domain, slot, value)}. In recent years, dialogue state tracking has drawn more and more attention, and numerous methods are proposed (Mrkšić et al., 2017; Zhong et al., 2018; Nouri and Hosseini-Asl, 2018; Ramadan et al., 2018).

Despite many progresses have been achieved, these approaches track dialogue states for each slot separately without considering the correlation among slots (Ouyang et al., 2020; Wu et al., 2019; Lee et al., 2019; Hu et al., 2020; Ye et al., 2021b). Spoken language is not formal, in which ellipsis and cross-reference phenomena make multi-domain dialogue state tracking problematic as shown in Table 1. To provide the user with several options, the values of slot "attraction-area" in the domain "attraction" at the second turn, the system should look for the information in another domain "restaurant" because the user implicitly indicates that the attraction he is looking for should be near the restaurant without explicitly speaking it out. And the value of slot "taxi-destination" should be that the system mentioned at the first turn.

User: Hi, I'm looking for a cheap restaurant in the centre of the city. **Sys:** Nutnut is a steal and popular there. **State:** restaurant-area=centre; restaurant-pricerange=cheap; restaurant-name=nutnut
User: Is there any place of pleasure near it? **Sys:** What type of attraction do you like? State: **restaurant-area=centre**; restaurant-pricerange=cheap; restaurant-name=nutnut; **attraction-area=centre**;
.....
User: Can you book a taxi for me to get to the restaurant? **Sys:** Of course, could you please provide your departure place? **State:** restaurant-area=centre; restaurant-pricerange=cheap; **restaurant-name=nutnut**; attraction-area=centre; **taxi-destination=nutnut**

Table 1: An example of a dialogue with three domains.

Several researchers have paid attention to modeling the correlations to some certain degrees (Ouyang et al., 2020; Hu et al., 2020; Heck et al., 2020). In these works, the correlation between the slot names is taken into consideration (Ouyang et al., 2020) or a strong prior knowledge is involved, i.e., the similarity coefficient is set to one manually if two slots are regarded to be relevant by human (Hu et al., 2020). But it may overlook the

Proceedings of the SIGdial 2022 Conference, pages 231–236
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

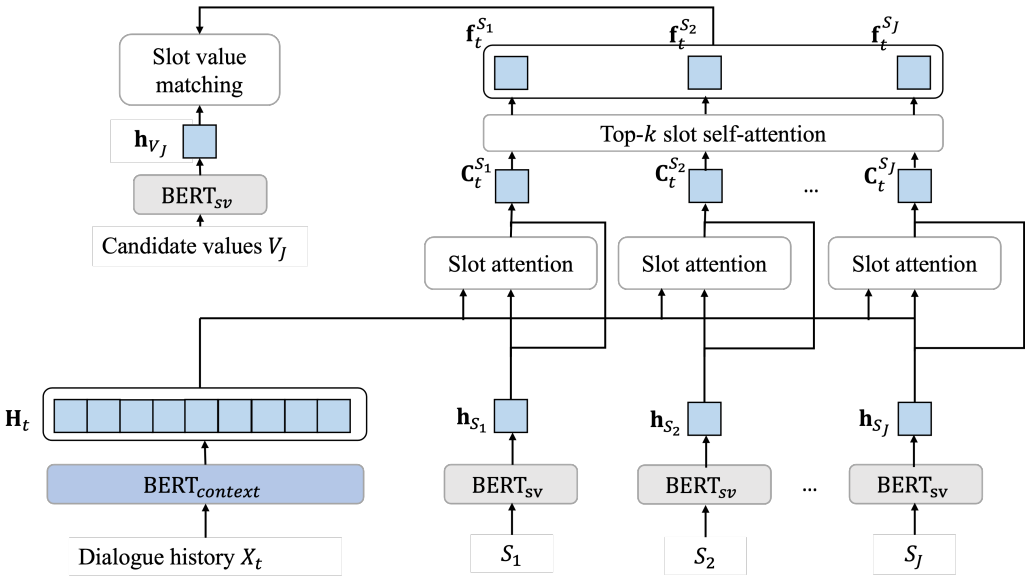


Figure 1: An overview of the proposed approach. For the BERT$_{context}$ model of the context encoder (solid blue rounded rectangle), its parameters are fine-tuned during training to encoder dialogue history; for the BERT$_{sv}$ model of the slot-value encoder (gray rounded rectangle), its parameters remain frozen to encoder slots and candidate values.

dependencies of some slots with the approach only considering the slot names. To address it, Ye et al. (2021b) proposed a slot self-attentive attention extracting slot-specific information for each slot from the dialogue context by utilizing a stacked slot self-attention module to learn the correlations among slots in a fully data-driven way without any human efforts or prior knowledge. However, it may involve some redundant information for some specific slots from other slots and result in incorrect prediction.

In this paper, we propose a dialogue state tracking with top-k slot self-attention. Here we have a premise of this work: *For each slot, not all of the others play a positive role in the value prediction for it. The more redundant information is involved, the worse would be the performance.* More specifically, in our work, in the layer where the slots interchange their information, we force each slot to pay attention to the other k slots with the highest scores and mask the rest out rather than considering all of them. We conduct experiments on MultiWOZ 2.0 and MultiWOZ 2.4 datasets and present that our proposed model works better than the methods handling the correlations with fully slot self attention.

The contributions of this paper are as follows: (1) We propose a top-k attention-based slot self-attention method for multi-domain dialogue state tracking; (2) The experimental results verify the effectiveness of our approach, and we find that the best result is obtained when each slot interchanges information with only a few slots.

2 Approach

Figure 1 shows the overview of the proposed model. It consists of a dialogue encoder, slot attention, top-k slot self-attention, and slot value matching.

2.1 Encoding

Let's define the dialogue history $D_T = \{R_1, U_1, ..., R_T, U_T\}$ as a set of system responses R and user utterances U in T turns of dialogue, where $R = \{R_t\}_{t=1}^{T}$ and $U = \{U_t\}_{t=1}^{T}, 1 \leq t \leq T$. We define $E_T = \{B_1, ..., B_T\}$ as the dialogue states of T turns, and each E_t is a set of slot value pairs $\{(S_1, V_1), ..., (S_J, V_J)\}$ of J slots. The context encoder accepts the dialogue history till turn t, which can be denoted as $X_t = \{D_1, ..., D_t, E'_{t-1}\}$, as the input and generates context vector representations $\mathbf{H}_t$.

$$\mathbf{H}_t = \text{BERT}_{context}(X_t) \quad (1)$$

Another BERT$_{sv}$ is employed to encode the slots and candidate values. The difference is that the parameters of BERT$_{sv}$ remain frozen during training. For those slots and values containing multiple tokens, the vector corresponding to the special token [CLS] is employed to represent them. For each slot

S_j and value V_j,

$$\mathbf{h}_{S_j} = \text{BERT}_{sv}(S_j) \quad (2)$$

$$\mathbf{h}_{V_j} = \text{BERT}_{sv}(V_j) \quad (3)$$

2.2 Slot attention

For predicting the states of a specific slot, it is necessary to extract slot-specific information from the dialogue history (**?**). A multi-head attention-based slot attention is employed to capture this information.

$$\mathbf{Q}_t^{S_j} = \mathbf{h}_{S_j}\mathbf{W}_Q + \mathbf{b}_Q \quad (4)$$

$$\mathbf{K}_t^{S_j} = \mathbf{H}_t\mathbf{W}_K + \mathbf{b}_K \quad (5)$$

$$\mathbf{V}_t^{S_j} = \mathbf{H}_t\mathbf{W}_V + \mathbf{b}_V \quad (6)$$

$$\boldsymbol{\alpha}_t^{S_j} = Softmax(\frac{\mathbf{Q}_t^{S_j}{\mathbf{K}_t^{S_j}}^T}{\sqrt{d_k}})\mathbf{V}_t^{S_j} \quad (7)$$

$$\mathbf{C}_t^{S_j} = \mathbf{W}_2 ReLU(\mathbf{W}_1[\mathbf{h}_{S_j}, \boldsymbol{\alpha}_t^{S_j}] + \mathbf{b}_1) + \mathbf{b}_2 \quad (8)$$

Where $\mathbf{W}_Q, \mathbf{b}_Q, \mathbf{W}_K, \mathbf{b}_K$, $\mathbf{W}_V$, and $\mathbf{b}_V$ are the parameters of the linear layers for projecting query, key and value respectively. $d_k = d_h/N$ in which d_h is the hidden size of the model, and N is the number of heads in multi-head attention.

2.3 Top-k slot self-attention

Inspired by Ye et al. (2021b), the information across different slots can be communicated by applying self-attention mechanism. In this work, we introduce a top-k slot self-attention to capture the correlation among different slots. We assume that, for each slot, not all of the other slots play a positive role in the value prediction. Forcing it with a few k slots with the highest attention scores performs better than considering all of them. To implement it, we mask out all but its k largest dot products with the keys in the slot-attention layers. For the l-th self-attention sub-layer, $\mathbf{F}_t^l = [\mathbf{C}_t^{S_1}, ..., \mathbf{C}_t^{S_J}]$, the formulations are as follows.

$$\widetilde{\mathbf{F}}_t^l = LayerNorm(\mathbf{F}_t^l) \quad (9)$$

$$\mathbf{G}_t^l = TopkAtt(\widetilde{\mathbf{F}}_t^l, \widetilde{\mathbf{F}}_t^l, \widetilde{\mathbf{F}}_t^l) + \widetilde{\mathbf{F}}_t^l \quad (10)$$

$$TopkAtt(Q, K, V) = Softmax(Topk(QK^\top))V \quad (11)$$

For the l-th feed-forward sub-layer, the formulations are as follows.

$$\widetilde{\mathbf{G}}_t^l = LayerNorm(\mathbf{G}_t^l) \quad (12)$$

$$\mathbf{F}_t^{l+1} = FFN(\widetilde{\mathbf{G}}_t^l) + \widetilde{\mathbf{G}}_t^l \quad (13)$$

The output of the final layer is regarded as the final slot specific vector $\mathbf{F}_t^{L+1} = [\mathbf{f}_t^{S_1}, ..., \mathbf{f}_t^{S_J}]$, where $\mathbf{f}_t^{S_j}$ represents the output corresponding to a slot.

2.4 Slot value matching

A Euclidean distance-based value prediction is performed for each slot. Firstly, the slot-specific vector is fed into a normalization layer. Then the distances between slot-specific vector and value are measured. Finally, the nearest value is chosen to predict the state value.

$$\boldsymbol{r}_t^{S_j} = LayerNorm(Linear(\mathbf{f}_t^{S_j})), \quad (14)$$

$$p(V_t^j|X_t, S_j) = \frac{\exp(-d(\mathbf{h}^{V_j}, \boldsymbol{r}_t^{S_j}))}{\sum_{V_j' \in \nu_j} \exp(-d(\mathbf{h}^{V_j'}, \boldsymbol{r}_t^{S_j}))} \quad (15)$$

where $d(\cdot)$ is Euclidean distance function, and ν_j denotes the value space of the slot S_j. The model is trained to maximize the joint probability of all slots. The loss function at each turn t is denoted as the sum of the negative log-likelihood.

$$\mathcal{L}_t = \sum_{j=1}^{J} -\log(p(V_t^j|X_t, S_j)) \quad (16)$$

3 Experiments

3.1 Datasets

MultiWOZ 2.0 (Budzianowski et al., 2018) and MultiWOZ 2.4 (Ye et al., 2021a) are employed as the datasets in our experiments. MultiWOZ 2.0 is one of the largest open-source human-human conversational datasets of multiple domains. It contains over 10,000 dialogues in which each dialogue averages 13.68 turns. MultiWOZ 2.4 is the latest refined version. It mainly fixes the annotation errors in the validation and test set. To make a fair comparison with the models evaluated on these two datasets, we follow the procedure in several previous works (Wu et al., 2019; Lee et al., 2019; Wang et al., 2020; Ye et al., 2021b) to keep consistent.

3.2 Training details

The used dialogue context encoder $\text{BERT}_{context}$ is a pre-trained BERT-base-uncased model of 12 layers with 768 hidden units and 12 self-attention heads. We employ another BERT-base-uncased model as the slot and value encoder BERT_{sv}. The

Table 2: The joint goal accuracy (JGA) of different models on the test set of MultiWOZ 2.0 and 2.4 dataset.

	Model	MW2.0	MW2.4
Open Vocabulary	TRADE (Wu et al., 2019)	48.62	54.89
	SOM (Kim et al., 2020)	51.72	66.78
	TripPy (Heck et al., 2020)	-	59.62
	SimpleTOD (Hosseini-Asl et al., 2020)	-	66.78
Ontology	SUMBT (Lee et al., 2019)	46.65	61.86
	DS-DST (Zhang et al., 2020)	52.24	-
	DS-Picklist (Zhang et al., 2020)	54.39	-
	SAVN (Wang et al., 2020)	54.52	60.55
	SST (Chen et al., 2020)	51.17	-
	STAR (Ye et al., 2021b)	54.53	73.94
	Top-k SSA(k=1)	**54.82**	77.10
	Top-k SSA(k=3)	54.47	**77.25**

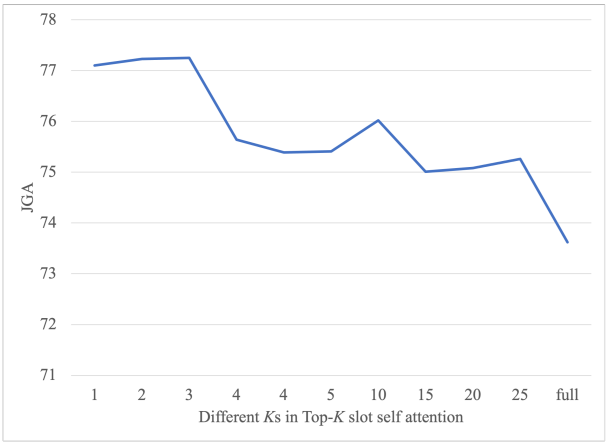


Figure 2: The results (JGA) of the proposed model based on top-k slot self-attention with different ks on MultiWOZ 2.4 dataset.

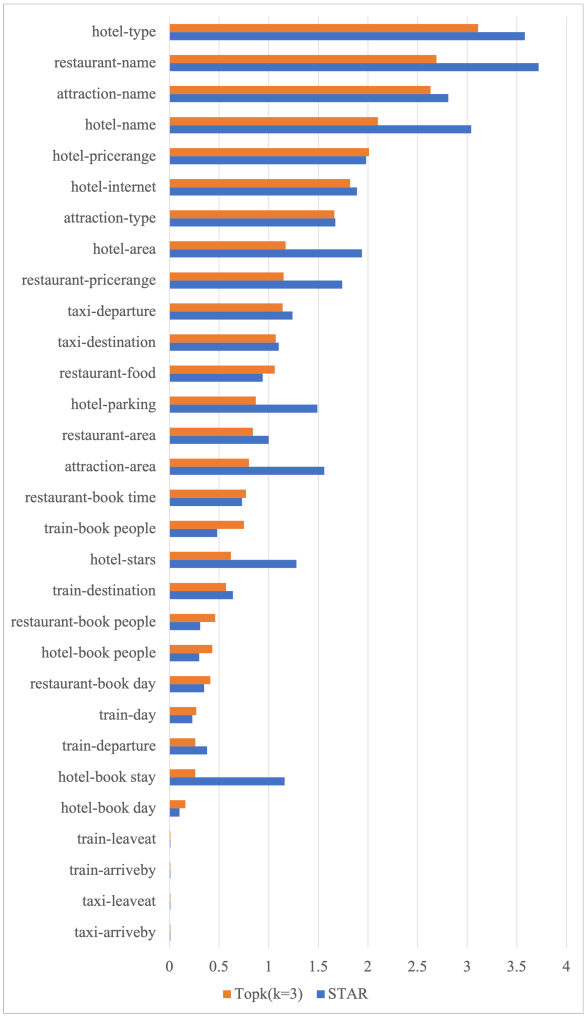


Figure 3: The error rate of each slot in the STAR and our proposed model on MultiWOZ 2.4 dataset.

number of attention heads for slot attention and slot self-attention is 4. The number of slot self-attention layers is 6. Adam optimizer is adopted with a training batch size of 8. The hidden size is set to 768 for the model. The slot attention part has 6 layers in which the number of attention heads is 6 as well. Adam is used as the optimizer with a learning rate of 4e-5 for encoder and 1e-4 for other parts. The hyper-parameters are chosen from the best-performing model over the validation set. We use the training batch size 16 and dropout rate 0.1 on utterances in a dialogue history.

4 Results and analysis

4.1 Main results

Table 2 shows the main results. We compare our approach with several typical and SOTA methods on this task. Top-k SSA denotes our proposed model with top-k slot self-attention. Joint goal accuracy (JGA) is employed to evaluate the overall performance. The joint goal accuracy is a strict measurement comparing the predicted values of each slot with ground truth for each dialogue turn, and the prediction is considered correct if and only if all the predicted values match the ground truth values without any error at each turn. As shown in Table 2, our proposed model achieves the best performance on these two datasets. We utilize the Wilcoxon signed-rank test, and the proposed approach is statistically significantly better ($p < 0.05$) than baselines. Specifically, our model with top-k (k=1) SSA for the MultiWOZ 2.0 dataset obtains a JGA of 54.82%. For the refined MultiWOZ 2.4 dataset, our model with top-k (k=3) SSA achieves a JGA of 77.25%, which outperforms other models by a large margin.

4.2 The effect of different ks

We investigate the performance using different ks to have a further understanding. As shown in Figure 2, the best performance is obtained when k is small, which means each slot performs infor-

User: Hi, I am looking for a place to eat some indian food. **Sys:** Do you have a price range in mind?
.....
User: I would like a place in the south, please. **Sys:** Taj Tandoori is the place you want to go. It meets all of your needs. STAR: restaurant-name=none Ours: restaurant-name=taj tandoori
.....
User: I want a taxi from the restaurant that I am at. **Sys:** Ok, so you would like a taxi from the restaurant to the park? Could you please let me know your desired departure and arrival times? STAR: taxi-departure=tandoori palace Ours: taxi-departure=taj tandoori
User: I am sorry, I would like a taxi from Wandlebury country to Taj Tandoori. I would like the taxi to pick me up at 10:15. **Sys:** Okay, I have booked a taxi for you it will be white tesla ... STAR: taxi-departure=tandoori palace; taxi-destination=Wandlebury country Ours: taxi-departure=Wandlebury country; taxi-destination=taj tandoori

Table 3: An example of a dialogue MUL2491 in MultiWOZ 2.4 dataset.

mation interchange with only a few slots. Then it drops a lot with the increase of k. It verifies our assumption that it is positive to force each slot to interchange information with limited slots than all of them to prevent abundant information interchange, in which the more redundant information is involved, the worse would be the performance.

4.3 Error analysis

Figure 3 presents the error rate of each slot. First it can be noticed that the overall error rate is reduced with our model. We also find that, comparing with the previous SOTA model STAR, the performance of the slots that may interchange information with others, e.g., *hotel-area*, *restaurant-area*, is improved by a large margin with our model. The performance of *"taxi"*-related and *"train"*-related is also improved slightly. Even though our model reduce the error rates of several *"name"*-related slots, like *"restaurant-name"* and *"hotel-name"*, they still have very high error rate.

4.4 Case study

Table 3 demonstrates an example in the test set of MultiWOZ 2.4 dataset. We can note that firstly STAR makes a mistake in the prediction for the slot *"restaurant-name"* while our model correctly find it. At the last turn, the user indicates his/her *"taxi-departure"* and *"taxi-destination"*. Although STAR capture the "Wandlebury country" but it fails to find the correlation of these slots, and the value of *"restaurant-name"* is copied from the error predicted value for this slot at the previous turns.

5 Conclusion

In this work, to address the correlation among different slots, we propose multi-domain dialogue state tracking with top-k slot self-attention, in which, each slots is forced to interchange information with the k slots with highest scores than all of them. We conduct experiments on MultiWOZ 2.0 and MultiWOZ 2.4 datasets and present that our model works better than existing methods that consider the correlations. The best results can be obtained when each slot interchanges information with only a few other slots.

Acknowledgements

This study was partly supported by JSPS KAKENHI Grand Number JP22K12069.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A Simple Language Model for Task-Oriented Dialogue. In

Advances in Neural Information Processing Systems, volume 33, pages 20179–20191.

Jiaying Hu, Yan Yang, Chencai Chen, Liang He, and Zhou Yu. 2020. SAS: Dialogue state tracking via slot attention and slot information sharing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6366–6375.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1777–1788.

Elnaz Nouri and Ehsan Hosseini-Asl. 2018. Toward scalable neural dialogue state tracking. In *NeurIPS 2018, 2nd Conversational AI workshop*.

Yawen Ouyang, Moxin Chen, Xinyu Dai, Yinggong Zhao, Shujian Huang, and Jiajun Chen. 2020. Dialogue state tracking with explicit slot connection modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 34–40.

Osman Ramadan, Paweł Budzianowski, and Milica Gašić. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 432–437.

Yexiang Wang, Yi Guo, and Siqi Zhu. 2020. Slot attention with value normalization for multi-domain dialogue state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3019–3028.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2021a. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. *arXiv preprint arXiv:2104.00773*.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021b. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, pages 1598–1608.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 1458–1467.

Building a Knowledge-Based Dialogue System with Text Infilling

Qiang Xue, Tetsuya Takiguchi , Yasuo Ariki
Graduate School of System Informatics, Kobe University
xueqiang,takigu,ariki@stu.kobe-u.ac.jp

Abstract

In recent years, generation-based dialogue systems using state-of-the-art (SoTA) transformer-based models have demonstrated impressive performance in simulating human-like conversations. To improve the coherence and knowledge utilization capabilities of dialogue systems, knowledge-based dialogue systems integrate retrieved graph knowledge into transformer-based models. However, knowledge-based dialog systems sometimes generate responses without using the retrieved knowledge. In this work, we propose a method in which the knowledge-based dialogue system can constantly utilize the retrieved knowledge using text infilling. Text infilling is the task of predicting missing spans of a sentence or paragraph. We utilize this text infilling to enable dialog systems to fill incomplete responses with the retrieved knowledge. Our proposed dialogue system has been proven to generate significantly more correct responses than baseline dialogue systems.

1 Introduction

Building open-domain dialog systems that generate human-like response is a challenging area for natural language processing. In recent years, generation-based dialogue systems, such as Microsoft's DialoGPT (Zhang et al., 2019) and Google's Meena (Adiwardana et al., 2020), have demonstrated impressive performance in simulating human-like conversations. However, when the human asks "What time is it?", the generation-based system will develop a conversation based on the old information contained in the training data. It has been reported that the "illusion problem" generates responses that are not based on the latest facts (Komeili et al., 2021). To address this, research on knowledge-based dialogue systems utilizing external knowledge has attracted attention as a dialogue system that can retrieve appropriate external knowledge.

Alternatively, many knowledge-based dialogue systems (Galetzka et al., 2021; Dinan et al., 2018) learn to generate target response sentences by inputting retrieved knowledge and dialogue history in a concatenated form to a language model during the learning phase. However, it has been reported that in the inference phase, the response sentences are generated based only on the input dialogue history, despite the input of retrieved knowledge (Weston et al., 2018).

In this work, we propose a knowledge-based dialogue system with text infilling, which enables the dialogue system to constantly generate responses that include retrieved knowledge. Specifically, the system first inserts blank tokens before and after the retrieved knowledge. The inserted text is the incomplete response. Next, the proposed dialogue system takes the incomplete response as input and generates text. Finally, it replaces the blank tokens in the incomplete response with this text and outputs the completed response.

2 Related work

2.1 Generation-based Dialogue System

Recent advances in pre-trained language models have had great success in dialogue response generation. DialoGPT (Zhang et al., 2019), Plato-2 (Bao et al., 2020), Meena (Adiwardana et al., 2020), and Blenderbot (Roller et al., 2020) have achieved strong generation performances by training transformer-based language models on an open-domain conversation corpus. In contrast, our proposed method focuses on controlling the content of responses in the fine-tuning process.

2.2 Knowledge-Based Dialogue System

To improve the coherence and knowledge retrieval capabilities of dialogue systems, recent knowledge-based dialogue systems (Galetzka et al., 2021) using knowledge graphs integrate fixed background

Proceedings of the SIGdial 2022 Conference, pages 237–243
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

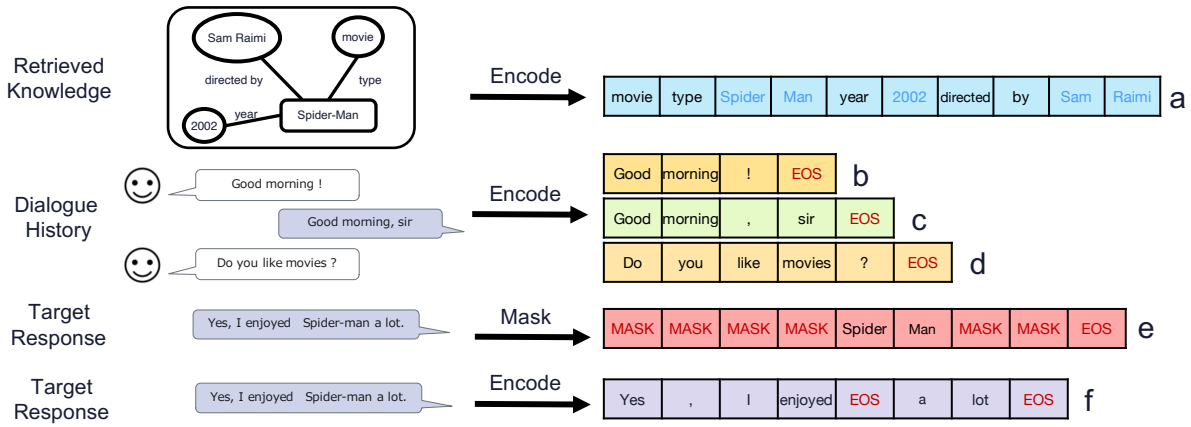


Figure 1: Encoding of knowledge and dialogue data in the training phase. Each type of encoded word sequences is indicated by a different colour.

context by creating pseudo utterances through paraphrasing knowledge triples, added into the dialogue history. Galetzka et al. (2021) proposed concise encoding for background context structured in the form of knowledge graphs, by expressing the graph connections through restrictions on the attention weights. In this work, we utilize the knowledge-based dialogue system using this encoding as our baseline.

2.3 Text Infilling

Text infilling is the task of predicting missing spans of text that are consistent with the preceding and subsequent text. Donahue et al. (2020) proposed a simple strategy for the task of text infilling which can enable language models to infill entire sentences effectively on three different domains: short stories, scientific abstracts, and lyrics. In this work, we utilize the text infilling task with this strategy to enable a knowledge-based dialogue system to generate responses that include retrieved knowledge.

3 Building the dialogue system

In this section, we introduce our proposed knowledge-based dialogue system that includes text infilling. We will introduce the training phase and the inference phase of the proposed dialogue system.

3.1 Training

In the training phase, the knowledge and dialogue history are encoded as follows:

- **Encoding Knowledge** (Figure 1-a) : The retrieved knowledge is concatenated with the entities and relations of each knowledge to form a knowledge series. Next, the different knowledge series are randomly concatenated and converted into a word sequence a.
- **Encoding Dialogue History** (Figure 1-bcd): Each utterance in the dialogue history is converted into a word sequence bcd, which consists of a sequence of tokens. A stop token <EOS> is added to the end of each converted word sequence.
- **Masking Target Response** (Figure 1-e) : First, the target response sentence is transformed into a word sequence e consisting of a sequence of tokens. Then, let L be the length of the converted word sequence e, and randomly select integers X and Y ($1 < X < Y < L$). The words from X to Y are retained (in Figure 1, X = 5 and Y = 6) and the other words in the sequence are replaced with <MASK> tokens. Finally, a stop token <EOS> is added to the end of the converted sequence e.
- **Encoding Target Response** (Figure 1-f) : First, a stop token <EOS> is added to the end of the two sequences that were replaced by the mask tokens in sequence e. Next, the two sequences are concatenated into sequence f.

The sequences encoded as described above are concatenated in the order of $abcdef$ and used as input to the language model. The training task is to maximize the probability of generating the target word sequence f.

3.2 Inference

The flow of the dialog system during the inference phase is as follows:

- **Encoding Knowledge and Dialogue History** (Figure 1-$abcd$) : The input data in the inference

phase are converted into word sequence $abcd$, as in the training phase in section 3.1.

- **Masking Knowledge** (Figure 2-e) : First, for the retrieved knowledge, we randomly select one entity of retrieved knowledge and transform it into a word sequence e. Next, integers X and Y ($0 < X, Y \leq MaskLen$, where $Masklen$ is a hyperparameter of mask tokens' number.) are randomly selected. X and Y <MASK> tokens are added in the left and right side of word sequence e. Finally, a stop token <EOS> is added to the end of the word sequence e.
- **Text Infilling** (Figure 2-f) : The word sequences encoded as described above are concatenated in the order abcde and input to the language model. The language model generates word sequence f sequentially by using a decoding strategy. Text Infilling is stopped when the second stop token <EOS> is generated.
- **Output** (Figure 2-g) : The stop token <EOS> splits the word sequence f into two word sequences, which are converted into word sequence g by replacing the left and right parts of the mask tokens <MASK> in e. The word sequence g is the output of the inference phase.

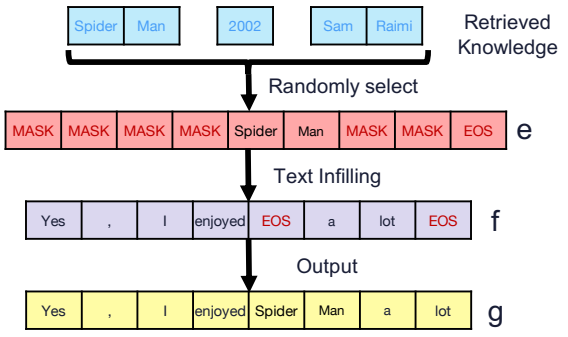


Figure 2: Encoding of knowledge and the output in the inference phase.

4 Experiments

We conducted experiments on the OpenDialKG dataset (Moon et al., 2019) which contains 15,000 dialogues. The dataset was collected in a Wizard-of-Oz setup, by connecting two human participants who were tasked to have an engaging dialogue about a given topic.

4.1 Experimental Details

Following the Zhang et al. (2019); Galetzka et al. (2021) work and section 3, we built 3 different types of the dialogue systems: a dialogue system without knowledge (generation-based dialogue system), a dialogue system with knowledge (knowledge-based dialogue system), and dialogue system with knowledge and text infilling (the proposed dialogue system).

We utilized DialoGPT-small (Zhang et al., 2019) as language model of 3 different dialogue systems. Table 1 shows the hyperparameters of the language models.

Table 1: Hyperparameters of language models

Total parameters	117M
Optimizer	AdamW
Max dialogue history	3
Decoding strategie	Greedy
Epochs	10
Batch size	4
MaskLen	10
Learning rate	6.0e-5

4.2 Evaluation metric

Automatic In the experimental evaluation, the quality of the response sentences is evaluated from two angles: diversity and correctness. DIST-n (Li et al., 2015), which represents the number of types of n-grams in the response sentences, is used as the evaluation index for diversity. BLEU-n (Papineni et al., 2002) and NIST-n (Doddington, 2002), which represent the degree of similarity between the response and the correct response, are used as evaluation indices for correctness. NIST-n is a variant of BLEU-n that weights n-gram matches by their information gain, i.e., it indirectly penalizes uninformative n-grams.

Ent-Res, which we employ to calculate the proportion of responses containing at least one entity of retrieved knowledge to all responses, and AvgLen, which represents the average number of words in the response sentences, are also used as evaluation indices. Furthermore, in order to compare the proposed method with previous models, we have listed the results achieved by previous models. The proposed method and conventional methods were compared using the same metrics, including the faithfulness metric FeQA (Durmus et al., 2020) and correctness metrics Rouge-L and BLEU-4.

Human In human evaluation, we use an evaluation technique called Best-Worst Scaling (BWS) (Flynn and Marley, 2014), which can handle a long list of options and always generates discriminating results. We employ three metrics at the utterance-level and dialogue-level: naturalness, informative-

Table 2: Results of the response sentences generated by each dialogue system. Higher is better.

Dialogue System	DIST-1	DIST-2	BLEU-1	BLEU-2	NIST-2	NIST-4	Ent-Res	Avg Len
Generation-Based	**11.93**	**36.79**	15.74	8.71	1.39	1.43	20%	10.86
Knowledge-Based	10.77	31.84	17.77	10.47	1.62	**1.69**	42%	10.45
Ours	9.09	32.18	**18.79**	**10.64**	**1.64**	**1.69**	**100%**	13.11

Table 3: Results of human evaluation using Best Worst Scaling (BWS).

Systems		Generation-Based	Knowledge-Based	Ours
naturalness	Best	30%	**40%**	30%
	Worst	35%	**21%**	44%
informativeness	Best	19%	33%	**48%**
	Worst	55%	27%	**18%**
coherence	Best	36%	**38%**	26%
	Worst	33%	**22%**	44%

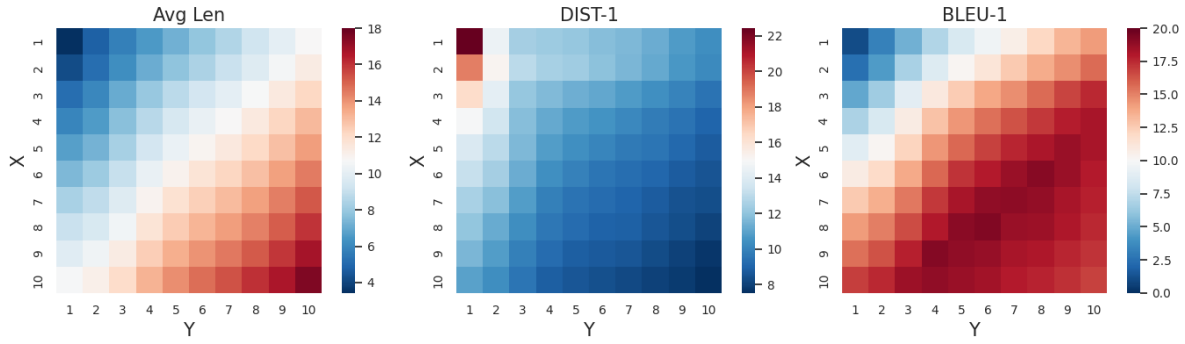


Figure 3: Heat map comparing the different X and Y impacts of the proposed dialogue system on three metrics. The blue shades denote lower values, white middle and black higher, with dark blue representing the lowest and dark black the highest values. X and Y denote the number of <MASK> tokens added to the left and right side of the word sequence e in Figure 2.

Table 4: Results of other dialogue system on OpenDialKG test data. Higher is better.

Dialogue System	FeQA	Rouge-L	BLEU-4
AdptBot	23.1	**31.0**	10.1
GPT-2+KE	19.5	19.0	5.5
GPT-2+KB	26.54	30.0	11.1
GPT-2+NPH	**28.9**	**31.0**	**11.3**
FSB	25.3	29.17	6.08
Ours	22.7	23.97	4.0

ness, and coherence. We randomly select 33 generated response examples. Three workers are asked to choose the best one and the worst one for three metrics in terms of response quality of each dialogue system with respect to the dialogue history.

- Naturalness is an utterance-level metric, judging whether the response is natural or not.
- Informativeness is also an utterance-level metric, evaluating whether the response is informative or not.
- Coherence is a dialogue-level metric, measuring whether the response is relevant and consistent with the context.

4.3 Results and Discussion

Table 2 shows the results of the response sentences generated by each dialogue system. The table shows that the proposed dialogue system reached the highest scores in the correctness evaluation index. This confirms the effectiveness of the text infilling task. On the other hand, the proposed method reached the highest value in the Ent-Res score, but the lowest value in the DIST-1 score. The improvement of the diversity in response sentences by the proposed method is a topic for future work.

Table 3 shows the results of human evaluation using Best Worst Scaling (BWS). We observed that the proposed method achieves lower naturalness and coherence scores compared to other models. Since the generated responses of the proposed method always contain entities as a result of the text infilling, the responses of the proposed method are rendered unnatural and incoherent. In future work, we will develop a module capable of determining whether knowledge should be embedded in the responses.

Table 5: Samples of responses generated by each dialog system. The retrieved knowledge entities are shown in blue.

Input 1	User A	what about tonto fistfight?
	User B	it was written by sherman alexie
	User A	was he a poet?
	Knowledge 1	poet: sherman alexie film producer: sherman alexie written by: sherman alexie
Generation-Based		yes he was a poet
Knowledge-Based		yes, he was a poet
Ours		yes, he was a poet. the sherman alexie wrote the poems and drawings of a rose.
Input 2	User A	they just got lebron james so that is a big benefit.
	User B	yes i was so happy that he signed. he is one of the greatest, right?
	User A	i'd say so. plus the lakes already have a good record, several nba finals championships under their belt.
	Knowledge 2	team: utah jazz
Generation-Based		i am sure the fans would love it. since lebron james is your favorite player, you must like him?
Knowledge-Based		that's awesome! i think he's a great player too.
Ours		i agree. i like the utah jazz. do you know who won that year?

Table 4 shows the comparison with previous models GPT-2+KB, AdapterBot (Madotto et al., 2020b), GPT-2+KE (Madotto et al., 2020a) and GPT-2+KB with Neural Path Hunter (NPH) (Dziri et al., 2021) and Few-Shot Bot (FSB) (Madotto et al., 2021). Due to the differences in the test dataset and the model sizes, the scores of the proposed method are just reference values. Nevertheless, the proposed method achieves lower FeQA, Rouge-L and BLEU-4 scores compared to previous models. Overall, NPH achieves the best performance, but it can also be applied to the proposed method; we leave this exploration to future work.

Table 5 shows samples of responses generated by each dialogue system. From the table, it can be confirmed that the proposed method can accurately use the retrieved knowledge and generate natural response sentences. On the other hand, the knowledge-based dialogue system is not able to use the knowledge. Despite this, it can be considered that knowledge 1 is not necessary to generate natural response sentences to the dialogue history of input 1. The development of a module that can determine the necessity of knowledge is a subject for future work.

4.4 Impact of <MASK> tokens

The results of the proposed dialogue system with different X and Y are compared using heat maps for various metrics, where X and Y denote the number of <MASK> tokens added to the left side of the word sequence e in Figure 2. Here, the heat map indicates a two-dimensional matrix with scores of the metrics such as BLEU-1 and DIST-1, computed by changing the length X and Y. We show the heat maps for three metrics in Figure 3. The heat maps for other metrics are shown in the Appendix A.

As can be observed in the heat map of Avg Len scores in Figure 3, the larger the sum of X and Y, the higher the value. It can be confirmed that the proposed dialogue system can correctly generate responses of the corresponding length based on X and Y. On the other hand, we can control the length of the generated responses by modifying X and Y. Due to the possibility of duplicate words in longer responses, the values in heat maps of Avg Len and DIST-1 show the opposite trend.

As can be observed in the heat map of BLEU-1 scores in Figure 3, the scores are similar when the sums of X and Y are equal, and the score is highest when the sum of X and Y is around 13. It can be confirmed that the scores are relevant to the sum of X and Y, not X nor Y. If the appropriate X and Y can be determined, the proposed dialogue system will have better performance. However, the appropriate sums of X and Y may be different in various datasets. Developing a method for finding the best combination on X and Y will be the subject of future work.

5 Conclusion

We proposed a knowledge-based dialogue system based on the text infilling method, aiming to improve the problem that the knowledge-based dialogue system generates responses without using retrieved knowledge. The proposed dialogue system can constantly incorporate external knowledge. In our experiments, the proposed dialogue system generated significantly more correct responses than baseline approaches.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2020. Plato-2: Towards building an open-domain chatbot via curriculum learning. *arXiv preprint arXiv:2006.16779*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. *arXiv preprint arXiv:2005.05339*.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. *arXiv preprint arXiv:2005.03754*.

Nouha Dziri, Andrea Madotto, Osmar Zaiane, and Avishek Joey Bose. 2021. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. *arXiv preprint arXiv:2104.08455*.

Terry N Flynn and Anthony AJ Marley. 2014. Best-worst scaling: theory and methods. In *Handbook of choice modelling*, pages 178–201. Edward Elgar Publishing.

Fabian Galetzka, Jewgeni Rose, David Schlangen, and Jens Lehmann. 2021. Space efficient context encoding for non-task-oriented dialogue generation with graph attention transformer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7028–7041.

Mojtaba Komeili, Kurt Shuster, and Jason Weston. 2021. Internet-augmented dialogue generation. *arXiv preprint arXiv:2107.07566*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020a. Learning knowledge bases with parameters for task-oriented dialogue systems. *arXiv preprint arXiv:2009.13656*.

Andrea Madotto, Zhaojiang Lin, Yejin Bang, and Pascale Fung. 2020b. The adapter-bot: All-in-one controllable conversational model. *arXiv preprint arXiv:2008.12579*.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

A Heat map of metrics

There are the heat maps of the DIST-2, BLEU-2, NIST-2 and NIST-4 scores of the proposed dialogue system with different X and Y on the test set in Figure 4.

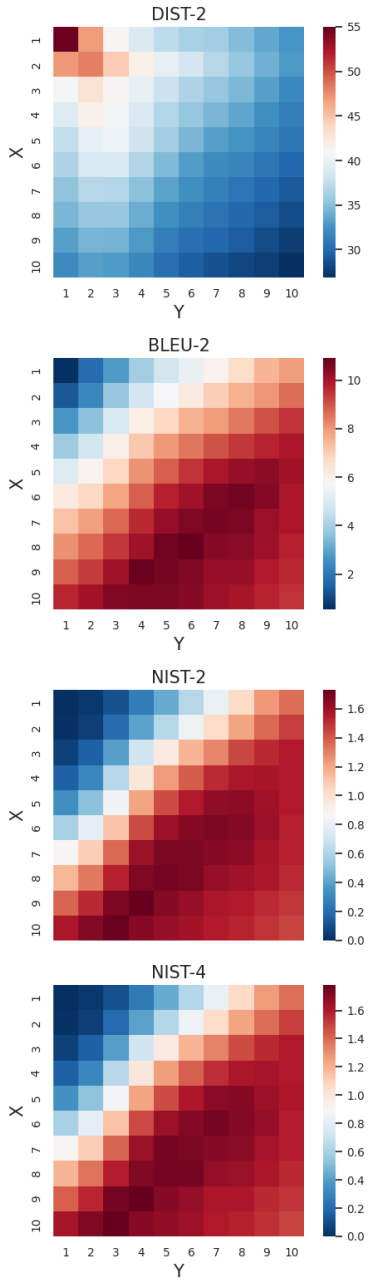


Figure 4: Heat map comparing the different X and Y impacts of the proposed dialogue system on three metrics. The blue shades denote lower values, white middle and black higher, with dark blue representing the lowest and dark black the highest values. X and Y denote the number of <MASK> tokens added to the left and right side of the word sequence e in Figure 2.

Generating Meaningful Topic Descriptions with Sentence Embeddings and LDA

Javier Miguel Sastre Martínez, Seán Gorman, Aisling Nugent, Anandita Pal
Accenture The Dock, R&D Global Innovation Center,
7 Hanover Quay, Grand Canal Dock, Dublin, Ireland
{j.sastre.martinez, sean.gorman, a.nugent, anandita.pal}
@accenture.com

Abstract

A major part of business operations is interacting with customers. Traditionally this was done by human agents, face to face or over telephone calls within customer support centers. There is now a move towards automation in this field using chatbots and virtual assistants, as well as an increased focus on analyzing recorded conversations to gather insights. Determining the different services that a human agent provides and estimating the incurred call handling costs per service are key to prioritizing service automation. We propose a new technique, ELDA (**E**mbedding based **LDA**), based on a combination of LDA topic modeling and sentence embeddings, that can take a dataset of customer-agent dialogs and extract key utterances instead of key words. The aim is to provide more meaningful and contextual topic descriptions required for interpreting and labeling the topics, reducing the need for manually reviewing dialog transcripts.

1 Introduction

Topic models are statistical tools for discovering the hidden semantic structure in a collection of documents/dialogs. One such widely used topic model is Latent Dirichlet Allocation (LDA, Blei et al., 2003). LDA is a hierarchical probabilistic model that represents each topic as a distribution over terms/words and represents each document/dialog as a mixture of the topics. One of the main issues with the standard LDA bag-of-words approach is that the discovered topics can be difficult to interpret, as the user is presented with only the key words per topic. Due to this, the user often needs to go through the documents/dialogs for each topic to gather more context. The ELDA (**E**mbedding based **LDA**) approach attempts to produce more interpretable topics by running the topic modeling at an utterance level. The resulting topics can be represented by the most relevant utterances per topic, giving more context to the analyst so they can better understand the topic, with little to no manual inspection of the dialogs.

Another issue with bag-of-word approaches is that they fail to capture co-reference resolution, homonymy, and polysemy. For example, the words "leave" and "depart" mean the same thing in similar contexts but will be treated as having different meanings. Conversely, one word, for example "right", can mean different things given the context but will be treated as having the same meaning. Representing text as embeddings can overcome these issues to some extent. For example, word and sentence encoders such as (Google's) Multilingual Universal Sentence Encoder (MUSE, Yang et al., 2020), Sentence-BERT (SBERT, Reimers and Gurevych, 2019), etc. can capture the meaning of sentences and words in context with no need for any text pre-processing (e.g. stop word removal, part-of-speech tagging, lemmatization etc.).

A further challenge in running LDA is that it requires to specify in advance the number of topics to generate, which can be hard to determine in cases where the domain or data is not known in detail. The ELDA approach includes a novel technique to automatically estimate the number of topics to generate for a given dataset.

We compared the topic descriptions of the ELDA approach with that of standard LDA on the MultiWOZ dataset (Han et al., 2021).

2 Related Work

Cygan (2021) employed a method of topic modeling that leverages SBERT (Reimers and Gurevych, 2019) to create rich semantic document embeddings by averaging sentence embeddings, after which documents are assigned to a cluster using HDBSCAN. Once the clusters are created, Cygan uses LDA to construct a single topic descriptor (a list of key words) over the documents of each cluster. They claim in their analysis that a small set of documents clustered together by SBERT em-

Proceedings of the SIGdial 2022 Conference, pages 244–254
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

beddings can generate a coherent and interpretable topic, outperforming topics made from Doc2Vec (Le and Mikolov, 2014) based document embeddings. Our approach uses sentence-level topic descriptors rather than key words, and we apply a recent sentence encoder that supports multiple languages (Yang et al., 2020).

Kozbagarov et al. (2021) present another approach to generating interpretable topics by combining sentence embeddings with a topic modeling technique, though they use EM (expectation-maximization) instead of LDA and use averaged BERT word embeddings (Devlin et al., 2019) instead of a pretrained sentence encoder. Like us, they cluster the resulting sentence embeddings and estimate the probability of sentence occurrence within texts, assuming sentences within each cluster as identical. However, they apply EM on the text distribution over sentence clusters, thereby representing each topic as a probability distribution over sentence clusters. Finally, they also labeled the clusters with the closest sentence to the cluster centroid, as we do. Their experimental results show a high level of interpretability in the formed topics compared to traditional topic modeling approaches.

Moody (2016) described the lda2vec model, which builds representations over both words and documents by mixing word vectors (word2vec) with Dirichlet-distributed latent document-level mixtures of topic vectors, yielding sparse and interpretable document-to-topic proportions in the style of LDA. The topics obtained on the 20newsgroup corpus are shown to yield high mean topic coherences, correlating with human evaluations of the topics.

Dieng et al. (2020) developed an embedded topic model (ETM) which integrates topic embeddings with traditional topic models. Like in LDA, the ETM is a generative probabilistic model, where each document is a mixture of topics, and each term is assigned to one of the topics. In contrast to LDA, each term is represented by an embedding, and each topic is a point in that embedding space. The topic's distribution over terms is proportional to the exponentiated inner product of the topic's embedding and each term's embedding. The ETM claims to discover more interpretable topics even with large vocabularies that include rare words and stop words. It claims to outperform LDA in both predictive performance and topic quality and diversity as measure by topic coherence.

Our work specifically targets topic discovery in customer call conversations rather than general documents, such as news articles or publications, as in most of the related work. We have also created novel techniques in: (i) automatically deciding on the number of topics to produce and (ii) to measure the interpretability and accuracy of the produced topics.

3 Method

Given a collection of dialogs segmented into utterances, either by a speech-to-text system that includes diarization or based on metadata provided by a text-messaging system (see Table 1), ELDA applies topic modeling at the utterance level, producing topics represented by a selection of key utterances relevant to each topic. The method is split in 5 steps, namely: computing the utterance vectors (Section 3.1), clusterizing the utterance vectors (Section 3.2), auto-labeling the clusters (Section 3.3), encoding the dialogs as bags of utterance clusters (Section 3.3), and applying LDA on these bags of utterance clusters (Section 3.5), using then their corresponding cluster auto-labels as the resulting topic key items.

3.1 Utterance encoding

We first apply a sentence encoder to each utterance to obtain a vector representation. In particular, we have tested Universal Sentence Encoder (USE, Cer et al., 2018), Multilingual Universal Sentence Encoder (MUSE, Yang et al., 2020), and Sentence-BERT (SBERT, Reimers and Gurevych, 2019). Each of these embed text segments into vectors of a fixed size. In our approach, we settled on using MUSE as it supports 16 different languages and produced results comparable to the other two. Comparison was done as explained in Section 4.3, though we only present here the results obtained with MUSE to avoid repetition.

3.2 Utterance clustering

We compute groups of semantically similar utterances by clustering the set of utterance embeddings. This allows us to represent each dialog as a collection of utterance clusters/types. For the clustering, we employ a combination of k-means (MacQueen, 1967) and DBSCAN (Ester et al., 1996) algorithms in two steps. We first apply k-means to create an initial set of k clusters, with a relatively low k proportional to the total number of utterances n (e.g.,

#	Speaker	Utterance
1	CUSTOMER	Hi , I 'm looking for a train that is going to cambridge and arriving there by 20:45 , is there anything like that?
2	AGENT	There are over 1,000 trains like that . Where will you be departing from ?
3	CUSTOMER	I am departing from birmingham new street .
4	AGENT	Can you confirm your desired travel day ?
5	CUSTOMER	I would like to leave on wednesday.
6	AGENT	I show a train leaving birmingham new street at 17:40 and arriving at 20:23 on Wednesday . Will this work for you ?
7	CUSTOMER	That will , yes . Please make a booking for 5 people please
8	AGENT	I 've booked your train tickets , and your reference number is A9NHSO9Y.
9	CUSTOMER	Thanks so much .

Table 1: Sample dialog between customer and agent regarding a train booking in the MultiWOZ dataset. Note some utterances may convey more than one sentence (e.g., utterances 2, 6 and 7).

$n/5000$). Then we apply DBSCAN to the set of utterances of each initial cluster in order to avoid having to choose a final number of clusters to generate: DBSCAN creates a cluster for each set of a minimum size min_pts of transitively connected points, where 2 points are connected (or neighbors) iff they are within a maximum distance eps. Sets smaller than min_pts do not form clusters, naturally discarding rare utterances. As a drawback, DBSCAN requires to compute the distance between every pair of points, which can be time intensive for the case of large sets of utterances. By pre-clustering the set of utterances with k-means we reduce the number of distances to compute by several orders of magnitude. The two main hyperparameters of DBSCAN, eps and min_pts, have considerable impact on the quality of ELDA results. The tuning of these hyperparameters is described in Section 4.3.

3.3 Utterance cluster auto-labeling

For each utterance cluster we select the best utterance representative to serve as the cluster's label. We first compute the cluster centroid (the average of its vectors), then select the utterance whose vector is closest to the centroid. An example of the clusters and their labels can be found in appendix A.2.

3.4 Dialog encoding

To perform topic modeling on the labeled utterances clusters, we represent each document/dialog as a bag of utterance clusters (instead of a bag of words), followed by the standard LDA approach. We use a TF-IDF-like vectorizer to compute the document/dialog vectors by considering the utterance clusters as terms (i.e., we compute utterance cluster frequency-inverse document frequency). The set of document vectors form the document-cluster matrix D.

3.5 LDA topic modeling

Like k-means, the LDA algorithm requires to specify the number of topics K to compute in advance. However, it is often difficult to choose a proper value, especially for unknown domains. We propose a new approach to automatically select the number of topics by modeling the topic coverage decay using an exponential function (see Algorithm 1). The goal of this approach is to automatically discover as many real services/use cases in call center conversations as possible, at the expense of generating an excess of topics that are either redundant, subcategories of other topics, or noise.

Instead of specifying K, the algorithm requires a rough estimate of the interval $[K_{min}, K_{max}]$ comprising K. Starting from K_{min} and at step increments, an LDA topic model is computed and tested for the given document-cluster matrix D and number of topics K until a complying model is found. To test compliance of a model, each dialog is assigned to its highest probability topic, according to the model, and the coverage of each topic (proportion of total dialogs assigned to each topic) is computed. The topic coverages are sorted in descending order and an exponential function is fitted to smooth the decay curve $y = me^{tx}$ with m as the y-intercept and t as the exponent factor (refer to Figure 1). Using the inverse of the exponential function derivative, we find the frontier between

Algorithm 1 exponential_decay_LDA(D)

Input: D, document cluster matrix
Parameters: $K_{min}, K_{max}, step,$
$slope\_threshold, min\_tail\_ratio$
Output: lda_model

for each $K = K_{min}$ to K_{max} by $step$ **do**
  $lda\_model \leftarrow$ train_lda_model(D, K)
  for each $i = 0$ to $K - 1$ **do**
    $topic\_dialogs_i \leftarrow \emptyset$
  end for
  for each dialog d **do**
    $i \leftarrow$ topic index for which d has its highest probability, according to lda_model
    $topic\_dialogs_i \leftarrow topic\_dialogs_i \cup \{d\}$
  end for
  for each $i = 0$ to $K - 1$ **do**
    $topic\_coverage_i \leftarrow \frac{|topic\_dialogs_i|}{K}$
  end for
  $X \leftarrow (0, 1, \ldots, K - 1)$
  sort_descending$(topic_coverage)$
  $m, t \leftarrow$ exponential_regression(X, Y)
  $x_t \leftarrow \frac{\ln(\frac{slope\_threshold}{mt})}{t}$
  $tail\_ratio \leftarrow \frac{|\{topic_i : i \geq x_t\}|}{K}$
  if $tail\_ratio \geq min\_tail\_ratio$ **then**
    break
  end if
end for

the head and the tail of the exponential function (dashed line in Figure 1), where the tail is the part of the curve with a slope below $slope\_threshold$. We then compute $tail\_ratio$, the proportion of topics in the tail, and check if it is greater than or equal to the threshold min_tail_ratio. If true, the algorithm stops and returns the corresponding model; otherwise, further LDA models for higher K values are computed until either min_tail_ratio or K_{max} is reached. By enforcing a minimum tail ratio, we expect to discover most of the relevant conversation topics while limiting the number of topics to compute. After a certain point, increasing K results in a greater number of topics in the tail region, each one covering a very small portion of the totality of dialogs.

We used the following parameter values in all our experiments: $K_{min} = 5$, $K_{max} = 60$, $step = 1$, $slope\_threshold = -0.001$ and $min\_tail\_ratio = 0.4$. For the MultiWOZ dataset, the algorithm stopped at $K = 24$ (refer to Figure 1).

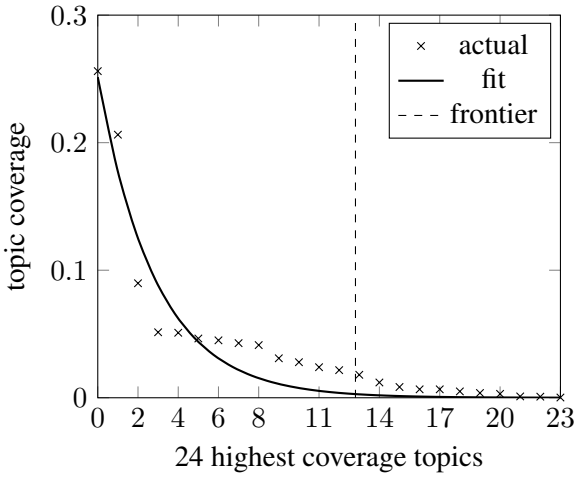


Figure 1: Plot showing the exponential decay approach for $K = 24$ (the first compliant K found) on MultiWOZ data. The $\times$'s represent the actual topic coverages, the curve denotes the best fitted exponential function and the vertical dashed line denotes the frontier between the head and the tail regions.

Each topic in the resulting model is a probability distribution of utterance clusters where each cluster is labeled with the most representative utterance. Thus, each topic can be represented by a set of key utterances, thereby providing descriptive context to the user in the process of interpreting and labeling the topics.

The ELDA result comprises a document/dialog-topic matrix (just like the standard LDA) and a topic-cluster or topic-utterance matrix (contrary to topic-word matrix of standard LDA).

4 Experiments

4.1 Data

To evaluate the quality of the ELDA approach we use the MultiWOZ dataset (Han et al., 2021), which comprises more than 10,000 annotated agent-customer dialogs across 7 domains/intents, namely: train, taxi, hotel, restaurant, attraction, police and hospital (Table 2, Figure 2). The dialogs are segmented into turns, which we use as utterances, and each dialog is annotated with the customer's intents, each dialog having at least one intent. In our case, we refer to each dialog's set of intents as its "label".

# dialogs	# utterances	# intents
10,438	224,179	7

Table 2: MultiWOZ data metrics

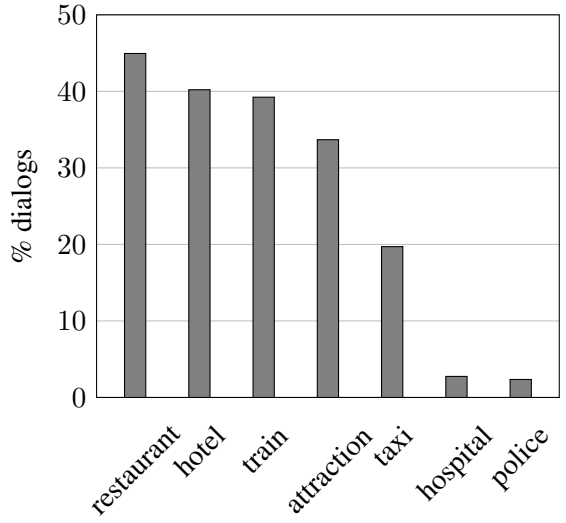


Figure 2: True intent distribution of the MultiWOZ dataset – Vertical axis denotes percentage of dialogs per intent

4.2 Evaluation methods

In this section we discuss two different aspects of evaluating ELDA. Mainly we compare ELDA's results with that of standard word-level LDA based on two evaluation criteria:

1. **Accuracy of dialog label identification**
2. **Interpretability of topic key utterances vs topic key words**

Accuracy: To measure the accuracy of a topic model, we must first manually inspect its output topics and label each topic with one of the seven MultiWOZ intents. For simplicity, we assume each topic has just one intent. For each topic, we first observe the topic key items (words for standard LDA and utterances for ELDA) and their respective scores. Giving priority to the key items with higher scores, we identify the related dominant intent and select it as the topic label (see Table 3). Topics with an equal mixture of different intents (more than one dominant intent), or those with unclear intents, are not given any label (see Table 4). We first label the bigger topics (based on topic coverage) and proceed towards the smaller ones. This strategy allows for identifying the most frequent intents first, while also considering the greatest number of dialogs in the least amount of time. Smaller topics that are subcategories of the bigger topics (e.g., Chinese restaurant booking vs restaurant booking) are given the same labels as the corresponding bigger topics.

Topic 8		
Cluster	**Cluster label**	**Score**
41	I am looking for a hotel instead of a guesthouse .	0.119
11	Is there a price range you 'd like ?	0.080
39	I need to book it for 4 people starting from saturday for 5 nights .	0.062
3	Can I get some help finding a hotel or guesthouse please ?	0.043
30	I need free parking and free wifi though .	0.034
10	I would like to book a reservation for it .	0.033
46	There are a couple of options .	0.031
23	I would like a guesthouse that is 4 stars .	0.031
34	Is there a particular area of town you 'd like to be in ?	0.029
32	I am also looking for a place to go in town .	0.027

Table 3: An example of topic with a clear dominant intent "hotel" (label given by either the oracle, annotator 1 or annotator 2 was "hotel")

Next, we assign these labeled topics to dialogs. For each dialog, we find all topics that have a probability score greater than or equal to the mean dialog-topic probability score (average of the probabilities in the dialog-topic matrix). The reason for selecting the mean as the threshold is that the topic probabilities, after being sorted for each dialog, are likely to follow a skewed distribution and the mean helps to filter out the lower probability or less frequent topics, steering the focus towards the higher probability or dominant topics for each dialog. We take the union of those dominant topics' labels as the predicted label for each dialog. Thus, each dialog will have zero, one or more of the seven MultiWOZ intents as its label. We then compare these predicted labels to the true dialog labels. Any overlap between the true and predicted dialog labels is considered a hit, and the hit rate across all dialogs

Topic 5		
Cluster	**Cluster label**	**Score**
7	Is there anything else you need ?	0.023
9	The phone number is 01223351241 .	0.023
33	Can I get the phone number , postcode , and address please ?	0.023
35	I need to book a taxi please .	0.023
44	Glad that I could help .	0.023
40	From Cambridge , which is why I asked the Cambridge TownInfo centre .	0.022
21	No , indeed .	0.022
3	Can I get some help finding a hotel or guesthouse please ?	0.022
0	I 'll take a cheap one please .	0.022
31	From where will you be departing ?	0.022

Table 4: An example of a "noisy" topic with more than one dominant intent, hence gets no label (label given by either the oracle, annotator 1 or annotator 2 was "blank")

is computed. This hit rate, or overlap score as we call it, is the accuracy of our topic model.

Interpretability: To compare interpretability of topic key words with topic key utterances, we show a few example topics obtained by running standard LDA and ELDA respectively on the MultiWOZ dataset and describe the efforts required to interpret and label them.

4.3 Experiment details

In this section, we discuss the experiments we performed to evaluate ELDA.

Baseline: For the baseline standard LDA model we start by applying a standard NLP pre-processing pipeline to the dialog words comprising lowercasing, POS tagging, lemmatization and stop word removal. We then encode the dialogs as TF-IDF vectors using the Gensim library (Řehůřek and Sojka, 2010). While encoding, we also use the inbuilt Gensim filtering utility to first remove the words that appear in more than 90% of the dialogs and in less than two dialogs, and then keep the remaining most frequent 100,000 words only. We use the described exponential decay approach to compute LDA models for different numbers of topics and for the resulting model, the topics are then manually labeled. Finally, the topics are assigned to each dialog, and an overlap score between the dialog topic labels and the MultiWOZ true labels is computed for the sake of evaluation and comparison with ELDA.

ELDA: We first run a grid search to find optimal values of the DBSCAN hyperparameters min_pts (minimum points per cluster) and eps (maximum allowed distance between neighboring points in the same cluster), computing multiple ELDA models for each combination and then calculating the overlap score between the true and predicted dialog labels. To avoid having to manually label the topics for each hyperparameter combination, we use an oracle approach: for a given topic, find the set of dialogs that are dominant using mean as the threshold, and select as topic label the most frequent MultiWOZ intent in that set of dialogs. In the case where a topic does not have dialogs above the threshold, it gets no label and will not contribute to the overlap score. We tested Gensim filtering analogous to the process used in baseline LDA on ELDA but filtering out low and high document frequency clusters barely filtered any clusters out, which in turn had little to no impact on the overlap scores. The DBSCAN density parameter values used for the grid search are as below:

- min_pts: 3 and 5
- eps: 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.6, 0.7, 0.8 and 0.9.

Comparison: In the search for optimal ELDA model, we compare the different ELDA models' overlap scores with that of baseline LDA. For fair comparison, we apply the same oracle labeling approach to both ELDA and LDA models. Then, on obtaining the optimal ELDA model, it is evaluated against the baseline LDA model using the overlap scores obtained from manual labels of two annotators. To ensure oracle labeling is consistent with manual labeling, we also compare the oracle labels and the manual labels of both the baseline LDA and optimal ELDA models.

5 Results

In this section we first report the overlap scores of the baseline LDA model and the different ELDA models, using the oracle topic labeling for all the models. Then we report the results of the comparison between the oracle labels and manual labels (from the annotators) for both the baseline LDA and optimal ELDA (best grid-search model). Next, we show a comparison of the overlap scores resulting from manual labeling obtained from the baseline LDA with the same obtained from the optimal ELDA. We also compare the true intent distribution of MultiWOZ with that produced by the manual labels of the baseline LDA and optimal ELDA. Lastly, we exhibit the topic descriptions of the three biggest topics from the baseline LDA and optimal ELDA and compare their individual level of interpretability.

5.1 ELDA optimization

The overlap score using the oracle labels of the baseline LDA model is 0.9281, showing that there is a high similarity between the predicted and the true dialog labels. This score is used as the baseline that the ELDA optimization aims to match or exceed.

The best ELDA model produced an overlap score of 0.9555 using the oracle labels for min_pts = 5 and eps = 0.5, surpassing our baseline score for LDA.

Based on these optimization results (Table 5) we expect the best ELDA model to match the baseline LDA in overlap score using the manual labels obtained from the annotators. Before that, we need to ensure that the optimization of ELDA based on oracle labels is consistent with manual labeling.

5.2 Validation of oracle labels

To validate the use of oracle labeling in optimizing the ELDA results, two annotators manually labeled the topics of the baseline LDA and the best ELDA model, and then we compared those manual labels to the oracle's labels. The results seen in Table 6 show reasonable overlaps between the oracle and manual topic labels. This validates the use of the oracle labeling as an efficient alternative to manual labeling, and so, was considered a suitable approach to enable running the ELDA optimization. Note the optimal values found for hyperparameters min_pts and eps may be extrapolable to other datasets, given that the semantic similarity distance magnitudes are given by the sentence embedding and not by the dataset. Hence, we would not need to manually annotate other datasets to repeat the tuning of the hyperparameters, which would defeat the purpose of running ELDA.

min_pts	***eps***	**# topics**	**Overlap**
3	0.2	35	0.6868
	0.25	39	0.7603
	0.3	29	0.8463
	0.35	37	0.8912
	0.4	39	0.9449
	0.45	40	0.9461
	0.475	40	0.9503
	0.5	39	0.9517
	0.525	38	0.9516
	0.6	41	0.9493
	0.7	40	0.9428
	0.8	40	0.9289
	0.9	40	0.9415
5	0.3	35	0.8818
	0.4	40	0.9488
	0.475	37	0.9538
	0.5	**38**	**0.9555**
	0.525	40	0.9303
	0.6	42	0.9428

Table 5: Overlap scores of ELDA for different values of DBSCAN hyperparameters min_pts and eps, and different number of topics, based on oracle labels (best result in bold)

Model	**# topics**	**Annotator 1**	**Annotator 2**
Baseline LDA	24	0.71	0.75
Best ELDA	38	0.71	0.76

Table 6: Average annotator overlap scores between oracle and manual topic labels

5.3 Evaluation of ELDA

As discussed earlier, we evaluate ELDA against LDA based on two aspects: accuracy and interpretability. We measure both on the best ELDA model and the baseline LDA model.

5.3.1 Accuracy

The overlap scores of the best ELDA model are evaluated against that of the baseline LDA according to the manual annotations of their respective topics obtained from the two annotators (see Table 7). From the results we observe an average of

Model	# topics	Annotator 1	Annotator 2	Annotator avg.
Baseline LDA	24	0.8921	0.9157	0.904
Best ELDA	38	0.9040	0.8827	0.8934

Table 7: Average annotator overlap scores of the baseline LDA and best ELDA models

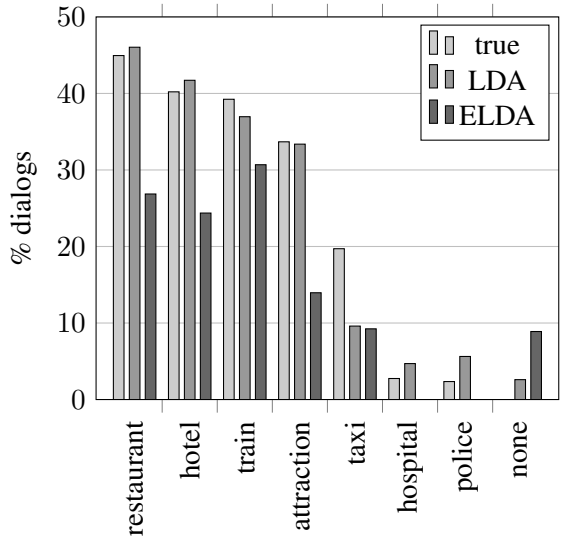


Figure 3: True vs baseline LDA vs best ELDA – Comparison of the intent ratio over the dialogs (% of dialogs per intent)

89% overlap with the best ELDA as opposed to an average of 90% overlap with the baseline LDA.

We also compare the true ratio of intents across the dialogs (% dialogs per intent) with that obtained from the best ELDA and baseline LDA models (Figure 3). For both LDA and ELDA, we show the average ratio of intents for each of the two annotators. Observing the plot, we see that ELDA has successfully identified the most frequent five of the seven intents, however LDA performs better in matching the true intent ratio. Potentially, further fine-tuning of the ELDA approach may improve these results.

5.3.2 Interpretability

In this section we analyze the top key items for the topics of the best ELDA model (along with the clusters) and the baseline LDA model (see Tables 8, 9 and 10 in the appendix). At first glance, the topic key words in Table 8 would be meaningful only to someone familiar with the MultiWOZ intents. To anyone with no knowledge of MultiWOZ, these key words lack the context required to interpret the topics, the context which can only be discovered when the same key words are used in sentences or utterances like in Table 9. For example, the highest scoring key word "train" in the largest baseline LDA topic versus the highest scoring key utterance "I need a train on thursday" in the largest ELDA topic, the key word "depart" versus the key utterance "What day and time would you like to depart", the key word "leave" versus the key utterance "I want to leave on Tuesday after 12:45", the key words "parking", "wifi", "free". versus the key utterance "I need free parking and free wifi though.", etc. show the power of utterances over words. As discussed before these key utterances are cluster labels and Table 10 provides a good idea about the quality of the clusters and validates the selection of their respective labels. Often in topic modeling evaluation, the reviewer must read the actual documents within the topics to better grasp what the topic is about, as the key words alone may not provide enough context. ELDA reduces this manual effort as the top utterances provide this context.

We ran both LDA and ELDA on an unseen, unlabeled technical helpdesk Accenture dataset (containing customer-agent dialogs resolving technical issues) with the optimal ELDA hyperparameters found for MultiWOZ and labeled the topics for both approaches. As expected, the topic key words were not descriptive enough to label the LDA topics and we had to manually review a few dialogs of each topic to understand what they were about. In contrast, the topic utterances provided the required context and meaning to understand and label the ELDA topics, with little to no need of reviewing the dialogs. For legal/privacy reasons we are not able to share these results.

6 Conclusions and future work

In this work we developed ELDA, an embedding-based LDA method, that represents each document or dialog in a dataset as a bag of utterance clusters instead of a bag of words. As a result, this approach represents each LDA topic as a probability distribution over utterance clusters which are labeled by the utterances closest to the cluster centroids. Unlike key words, the key utterances (cluster labels) provide more context to each topic, which helps to better interpret and label the topics. The ELDA and LDA approaches were evaluated and

compared using the MultiWOZ dataset. The results indicate ELDA is on par with the standard LDA in accurately identifying the existing topics or dialog intents, while producing easier-to-interpret topic descriptions that facilitate and accelerate the task of manually labeling the resulting topics.

The optimal ELDA hyperparameter values presented here may be extrapolable to other datasets, given that the semantic similarity distance magnitudes are given by the sentence embedding and not by the dataset. We continue testing ELDA with other (proprietary) datasets to verify this hypothesis.

One proposal for improving this work is to use a more stringent overlap metric in order to force the hyperparameter fine-tuning process to converge to better values. Note that the current approach considers a match between the predicted and true intents if any one of the intents match. Hence better hyperparameter values than the ones selected in this paper may be yet found.

7 Acknowledgements

We thank Paul A. Walsh, Ondřej Dušek and the SIGDIAL 2022 reviewers for their feedback.

References

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Natalie Cygan. 2021. Sentence-BERT for interpretable topic modeling in Web browsing data. Technical Report CS224N, Department of Computer Science, Stanford University.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

Ting Han, Ximing Liu, Ryuichi Takanabu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2021. MultiWOZ 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. In *Natural Language Processing and Chinese Computing*, pages 206–218, Cham. Springer International Publishing.

Olzhas Kozbagarov, Rustam Mussabayev, and Nenad Mladenovic. 2021. A new sentence-based interpretative topic modeling and automatic topic labeling. *Symmetry*, 13(5).

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org.

James B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Christopher E. Moody. 2016. Mixing dirichlet topic models and word embeddings to make lda2vec. *Computing Research Repository*, abs/1605.02019.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

A Appendix

A.1 Topic modeling results for baseline LDA and best ELDA models

Tables 8 and 9 list the three largest topics obtained from the baseline LDA and best ELDA models, respectively, along with the top 9 key items and their probability scores.

Topic 17		**Topic 12**		**Topic 2**	
Word	**Score**	**Word**	**Score**	**Word**	**Score**
train	0.045	hotel	0.032	hotel	0.024
leave	0.027	stay	0.020	guesthouse	0.022
arrive	0.022	guesthouse	0.020	parking	0.017
travel	0.021	parking	0.019	stay	0.016
ticket	0.021	night	0.019	free	0.016
depart	0.020	free	0.018	east	0.015
time	0.018	wifi	0.016	allenbell	0.015
cambridge	0.013	guest	0.016	north	0.013
departure	0.011	house	0.014	night	0.013

Table 8: Top 9 key words (with probability scores) for the three largest LDA topics

Topic	**Cluster**	**Cluster label**	**Score**
	25	**I need a train on thursday .**	**0.131**
	5	Train TR1526 leaves 17:40 and will get you there by 18:08 .	0.085
	14	I need to find a train leaving on Thursday going to Cambridge .	0.060
	27	I want to leave on tuesday after 12:45 .	0.048
6	38	What day and time would you like to depart ?	0.047
	31	From where will you be departing ?	0.034
	40	From Cambridge , which is why I asked the Cambridge TownInfo centre .	0.033
	43	Would you like me to book a reservation for it ?	0.027
	37	Its entrance fee is free .	0.027
	23	**I would like a guesthouse that is 4 stars .**	**0.150**
	41	I am looking for a hotel instead of a guesthouse .	0.063
	11	Is there a price range you 'd like ?	0.047
	39	I need to book it for 4 people starting from saturday for 5 nights .	0.046
37	30	I need free parking and free wifi though .	0.042
	3	Can I get some help finding a hotel or guesthouse please ?	0.041
	46	There are a couple of options .	0.034
	34	Is there a particular area of town you 'd like to be in ?	0.034
	36	Would you like any other info ?	0.030
	19	**I have your table booked for Tuesday at 15:15 .**	**0.146**
	8	I 'm looking for a moderately priced restaurant that serves chinese food .	0.095
	28	Is there a particular kind of restaurant you would like ?	0.065
	12	Your reference number is AJSQZY8R .	0.034
31	11	Is there a price range you 'd like ?	0.031
	6	It is in the centre part of town .	0.030
	42	The Booking was successful .	0.026
	22	I need the reference number please .	0.025
	10	I would like to book a reservation for it .	0.024

Table 9: Top 9 key utterances (with probability scores) for the three largest ELDA topics

A.2 Clustering results

Table 10 contains a sample of three clusters and some of their utterances. Each of these clusters is a top-scoring key item for each of the largest three topics from the best ELDA model (see the rows in bold in Table 9). To exhibit the quality of these clusters and represent them fairly, we take all the utterance embeddings within a given cluster, compute the distances to the cluster centroid, and rank them in ascending order. We display nine utterances in total, the first three are the three closest to the centroid, the next three are in the middle of the ranked list, and the last three are the three furthest from the centroid.

Cluster 25: I need a train on thursday .	**Cluster 23: I would like a guesthouse that is 4 stars .**	**Cluster 19: I have your table booked for Tuesday at 15:15 .**
Closest		
I need a train on thursday .	I would like a guesthouse that is 4 stars .	I have your table booked for Tuesday at 15:15 .
I need a train that gets me where I 'm going by 4:15 PM .	I am looking for a moderately priced hotel , that has a 4 star rating .	I would like to book a table for 6 at 15:15 on Tuesday .
I need a train that is leaving on wednesday .	I would prefer a 4 star hotel , are any of those three rated 4 stars ?	Please book a table for 7 at 15:15 on Wednesday .
Middle		
I have a number of trains leaving from london liverpool street .	yes it is 4 star	Can you book a table for seven people on Thursday at 15:00 ?
Actually yes , can you help me find a train to london liverpool street ?	Might you be willing to accept a place with 4 stars and free parking ?	Can you book me a table for 7 people on Sunday at 13:00 ?
Could I have the price for that train please ?	Yes , I would like to stay in the West area of town and I would also like it to have a 3 star rating .	Please book a table for 1 at 20:00 on friday .
Furthest		
The last train of the day will work for you .	Lucky star .	I am very sorry , our system was giving me an error , but I have managed to book your party of 5 at 16:45 on Tuesday .
There are 10 results of trains departing from Ely on Thursday .	It is four starts and it does have wifi .	You 'll find a table for 8 at Loch Fyne for 18:15 , reference number NGNNFSHD .
With your new criteria , that train wo n't work anymore , but there are other options .	The lucky star is chinese .	OK , a yellow Skoda will pick you up at the Cherry Hinton at 12:30 to get you to the restaurant in time for that 13:00 reservation .

Table 10: Sample of three utterance clusters (the highest scoring for each of the three largest topics from the best ELDA model) each with a sample of nine utterances.

How Well Do You Know Your Audience? Toward Socially-aware Question Generation

Ian Stewart and **Rada Mihalcea**
Computer Science and Engineering
University of Michigan
{ianbstew,mihalcea}@umich.edu

Abstract

When writing, a person may need to anticipate questions from their audience, but different social groups may ask very different types of questions. If someone is writing about a problem they want to resolve, what kind of follow-up question will a domain expert ask, and could the writer better address the expert's information needs by rewriting their original post? In this paper, we explore the task of *socially-aware* question generation. We collect a data set of questions and posts from social media, including background information about the question-askers' social groups. We find that different social groups, such as experts and novices, consistently ask different types of questions. We train several text-generation models that incorporate social information, and we find that a discrete social-representation model outperforms the text-only model when different social groups ask highly different questions from one another. Our work provides a framework for developing text generation models that can help writers anticipate the information expectations of highly different social groups.

1 Introduction

Writers are often expected to be aware of their audience (Park, 1986) and to minimize the effort required for others to understand them, especially if they cannot receive immediate feedback. However, NLP tools for writing assistance are not often made aware of the *social* composition of the audience (Ito et al., 2019; Zhang et al., 2020) and the information needs that different people may have. Preemptive writing feedback may therefore fail to help writers address the expectations of different people in their audience. This is especially important when the writer requests feedback from a specific group of people: in one post on a forum related to personal finance, a writer asks for help from financial "gurus" for advice about accepting a job offer.

A system that can preempt the hypothetical audience's information needs would enable the writer to revise their original post and avoid possible information gaps (Liu et al., 2012). Some online forums have already implemented crude solutions for this problem with automated reminders for writers to include basic information (e.g., location) in their post. Providing writers with preemptive questions can help especially in domains where different social groups have diverse information expectations. In the earlier example about personal finance, the advice-seeker could adapt their original post with answers to hypothetical "expert-level" questions (e.g. "Have you saved enough money for retirement?"), adding extra information that would enable experts to provide advice more quickly.

We cannot predict everyone's information needs, but some social groups with similar backgrounds (e.g., domain experts) will likely have consistent patterns in information expectations (Garimella et al., 2019; Welch et al., 2020). In this work, we evaluate several *socially-aware* question generation models with the goal of providing customized clarification questions to writers.

Our work contributes answers to the following questions:

- **How different are social groups based on the questions that they ask?** We collect a dataset of 200,000 Reddit posts seeking advice about a variety of everyday topics such as technology, legal issues, and finance, containing 700,000 questions.[1] We define several social groups that are relevant to possible information expectations such as expertise (§ 4.1). We demonstrate that different social groups, e.g. experts vs. novices, ask consistently different questions (§ 4.2, § 5.2.2).
- **How well can generation models predict socially-specific questions?** We extend an

[1] We will release the IDs for the post and author data, as well as the data processing code, to aid replication.

Proceedings of the SIGdial 2022 Conference, pages 255–269
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

existing generation model to incorporate social information about the question-askers (§ 5.1). In automated evaluation, a token-based socially-aware model outperforms the baseline for questions that are "divisive" and questions that are specific to a social group, particularly with respect to location as a social group (§ 5.2.3, § 5.2.4).

• **Are socially-aware questions useful for writers?** In human evaluations, we found that the socially-aware model is preferred over the text-only model for questions related to the question-asker's location and within the general advice-seeking domain (§ 5.3). This reinforces the utility of socially-aware models in scenarios where the social information is well-defined and where the topics are related to everyday concerns.

Importantly, the research presented in this paper shows that there are significant differences across groups with respect to questions they ask, and that we can develop models that are more attuned to these differences. Note that the goal of our work is not to improve the overall accuracy of a question generation system, but rather to develop methods that are sensitive to the needs of specific groups, thus paving the way toward technology that is available and useful for all.

2 Related Work

Question generation Question generation (QG) is unique among text generation tasks because it tries to address what a person *does not know*, rather than what they already know and want to write. QG systems are expected to create fluent and relevant questions based on prior text, in order to provide QA systems with augmented data (Dong et al., 2019) and students with question prompts to help their learning (Becker et al., 2012; Liu et al., 2012). In addition to typical supervised learning approaches (Du et al., 2017), reinforcement learning has proven useful, where questions are assigned a higher "reward" if they are more likely to have interesting answers (Qi et al., 2020a) and more relevant to the context (Rao and Daumé, 2019). Furthermore, work such as Gao et al. (2019) has proposed *controllable* generation techniques to encourage less generic questions, e.g. with higher difficulty. Such controllable-generation systems often leverage human-generated questions from a variety of domains, including Wikipedia (Du and Cardie, 2018), Stack Overflow (Kumar and Black, 2020), and Twitter (Xiong et al., 2019).

To our knowledge, prior work in question generation did not leverage the prior expectations of the question-askers. While sometimes providing controls for difficulty, no datasets currently include information about the inferred *background* of the question-askers. It seems natural that a person's prior knowledge would shape the information that they seek in response to a particular situation, yet analysis of the impact of social information on question generation remains absent. This study tests the role of social information in question generation using a dataset of posts from online forums, which feature complicated scenarios that can result in different information expectations between social groups.

Language model personalization Personalized language modeling often seeks to improve the performance of common language tasks, such as generation, using prior knowledge about the text's author (Paik et al., 2001). Personalization can improve task performance and make language processing more human-aware (Hovy, 2018), which ensures that a more diverse population is included in language models (Hovy and Spruit, 2016). To represent the text writer, personalized systems often integrate a writer's identity (Welch et al., 2020) or a writer's social network information (Del Tredici et al., 2019) into existing language models. A more generalizable approach converts the text-writer to a latent social representation such as an embedding (Pan and Ding, 2019), to be combined with the language representation in a neural network model where the social and text representations are learned jointly (Miura et al., 2017). We draw inspiration from the *contextualized* view of personalization from Flek (2020), and we represent the question-askers based on their prior behavior with respect to the specific *context* of a given post.

3 Data

In this study, we consider the task of generating clarification questions on information-sharing posts in online forums. We choose to study subreddits that have a high proportion of text-only posts, diverse topics, and where community members often ask information-seeking questions: `Advice` (lifestyle improvement), `AmItheAsshole` (social norms in complicated

Total posts	270694
Total questions	730620
Post length	304 ± 221
Question length	13.9 ± 8.08
Questions with question-asker data	77.7%
Questions with discrete question-asker data	75.2%
Questions with question-asker embeddings	43.5%

Table 1: Summary statistics about posts, questions, and question-asker data.

Subreddit	Posts	Questions
`Advice`	48858	87592
`AmItheAsshole`	61857	331345
`LegalAdvice`	53577	92737
`PCMasterRace`	31657	47613
`PersonalFinance`	74745	171333

Table 2: Summary statistics about subreddits.

situations), `LegalAdvice` (law disputes), `PCMasterRace` (computer technology), and `PersonalFinance` (money and investment). We collect all submissions ($\sim$ 8 million) to the above subreddits from January 2018 through December 2019, using a public archive (Baumgartner et al., 2020). We filter the post data to only include submissions written in English with at least 25 words, which we chose as a cutoff for posts that lack the context necessary for people to ask informed questions. To identify potential clarification questions, we collect all the comments of the submissions ($\sim$ 6 million) that are not written by bots, based on a list of known bot accounts like `AutoModerator`.

We conduct extensive filtering to include questions that are relevant and that seek extra information from the original post. The details are available in Appendix A. We summarize the overall data in Table 1, and we show the distribution of the posts and questions among subreddits in Table 2. Example posts and associated clarification questions are shown in Table 3.

4 Defining social groups

In this work, we assess the relevance of the question-asker's background in the task of question generation, by defining social groups and assessing their differences in question-asking.

4.1 Defining social groups

We collect a limited history for the question-askers ($N = 1000$ comments) to quantify relevant aspects of their background that may explain their information-seeking behavior. We consider the following social groups who are likely to have different information expectations:

1. `EXPERTISE`: A question-asker with less experience may ask about surface-level aspects of the post, while someone with more expertise might ask about a more fundamental aspect of the post. We quantify "expertise" using the proportion of prior comments that the question-asker made in the subreddit s (or a topically related subreddit; see § B.1) in which the original post was made. For example, if a question-asker has frequently written comments in `WallStreetBets` before asking a question in `PersonalFinance`, they are likely more familiar with financial terms than the average person. We define an `Expert` question-asker as anyone at or above the 75^{th} percentile of rate of commenting in a relevant subreddit, and a `Novice` question-asker as anyone below the percentile, where we chose the threshold to fit the skewed data distribution. Other threshold values produced similar results in social group classification.
2. `TIME`: A question-asker who replies soon after the original post was written may ask about missing information that is easily corrected (e.g. clarifying terminology), while a question-asker who replies more slowly may ask about more complicated aspects of the writer's request (e.g. the writer's intent). We quantify this with the mean speed of responses of the question-asker's prior comments *relative* to the parent post. We define a `Slow` question-asker as anyone at or above the 50^{th} percentile of mean response time, and a `Fast` question-asker as anyone below the threshold.
3. `LOCATION`: A question-asker who is based in the US may ask questions that reflect US-centric assumptions, while a non-US question-asker may ask about aspects of the post that are unfamiliar to them. We quantify location with the question-asker's self-identification from prior comments, using Stanza's English NER tool (Qi et al., 2020b) to identify LOCATION entities and OpenStreetMap to geo-locate the most likely locations. For those without self-identification, we identify all location-specific subreddits $\mathcal{S}_L$ in a question-asker's previous posts based on whether the subreddit name can be geolocated with high confidence (e.g., `r/NYC` maps to New York City). A question-asker a's location is identified with the location-specific subreddit where a writes at least 5 comments and where they write the most comments out of all location-specific subreddits $\mathcal{S}_L$.

Group category	Description	Social group	Example question	Example post title
EXPERTISE	Prior rate of commenting in the target subreddit, or a topically-related subreddit.	Expert ($\geq$ 75th percentile)	How much would you need to make on day 1 to meet your current financial obligations?	(PersonalFinance) Changing careers at 39
		Novice ($<$ 75th percentile)	Where do you live?	
TIME	Mean amount of time elapsed between original post and question-asker's comment, among all prior comments.	Fast ($<$ 50th percentile)	Does your wife have a relationship with him?	(LegalAdvice) Having a child and partner's father is sex offender
		Slow ($\geq$ 50th percentile)	If he is a sex offender, shouldn't he be kept away from children?	
LOCATION	Inferred location of question-asker.	US	Have you looked at the RX 580?	(PCMasterRace) Should I buy GTX 1050Ti?
		non-US	The 1050 is 160$ in India?	

Table 3: Group categories for question-askers, with example questions and posts.

Group category	Top-3 LIWC categories (absolute frequency difference)
LOCATION	
US >non-US	MONEY (0.512%), WORK (0.361%), RELATIV (0.337%)
non-US >US	FOCUSPRESENT (0.356%), FUNCTION (0.327%), AUXVERB (0.305%)
EXPERTISE	
Expert >Novice	MONEY (0.207%), YOU (0.135%), FOCUSPRESENT (0.106%)
Novice >Expert	DRIVES (0.097%), AFFILIATION (0.056%), REWARD (0.055%)
TIME	
Fast >Slow	YOU (0.312%), PPRON (0.225%), PRONOUN (0.160%)
Slow >Fast	DRIVES (0.105%), AFFECT (0.082%), IPRON (0.066%)

Table 4: LIWC category word usage differences across social groups (% indicates absolute difference in normalized frequency). All differences are significant with $p < 0.05$ via Mann-Whitney U test.

We summarize these definitions of different social groups in Table 3. The example questions in demonstrate that question-askers who occupy different groups tend to ask questions about different aspects of the original post: e.g. the Fast question-asker addresses a basic fact about the situation, while the Slow question-asker addresses a more complicated/hypothetical point.

4.2 Validating group differences

As a first step, we test for consistent differences in the types of questions asked by different social groups. We test for topical differences between the groups by comparing the relative rate of LIWC word usage in their questions, a common strategy to identify salient differences between social groups (Pennebaker et al., 2001). The results in Table 4 show consistent differences in word usage in the questions. Expert question-askers ask about money more often than Novices, which could indicate an assumption from prior experience that post authors' core problems stem from their financial decisions (even outside of the finance-related subreddits). Similarly, US question-askers have more questions about money and work than non-US readers, who often frame questions to address present-tense issues and write with more auxiliary verbs. Fast-response question-askers ask more often about the post author (YOU), which may indicate a stronger interest toward the post author's background, as opposed to slow-response question-askers who address the poster's high-level intentions (DRIVE) and emotional behavior (AFFECT). While it is possible that some of these differences are spurious, it is unlikely that they all relate to stylistic patterns such as regional differences (LOCATION), considering the prevalence of relevant LIWC categories (e.g. MONEY relates to financial questions, which are relevant to the data).

We verify these differences with a classification task, which we detail in § B.2.

5 Question generation

5.1 Model design

We build the generation models on top of the BART model (Lewis et al., 2020), a transformer model known to be resistant to data noise. We use the same pre-trained model (bart-base; $|V| \approx 50,000$) and the same training settings for all models.[2] The main point of the model modifications is not to achieve universally high accuracy, but to assess the value of different social data representations in question generation.

[2]Learning rate 0.0001, weight decay 0.01, Adam optimizer, 10 training epochs, batch size 2, max source length 1024 tokens, max target length 64 tokens, cross-entropy loss.

5.1.1 Social tokens

For the "social-token" model (a discrete representation), we add a special token $\{GROUP_g\}$ to the text input of the baseline model to indicate whether the asker belongs to social group g (cf. prior work in controllable generation; Keskar et al. 2019). The embeddings for these social tokens are learned during training in the same way as the other text tokens. All question-askers who could not be assigned to a group are represented with `UNK` tokens.

5.1.2 Social attention

For discrete modeling, we also consider customizing a separate part of the model for different social groups. Specifically, we change one of the attention layers of the typical transformer model (Vaswani et al., 2017) to represent differences in how different question-askers may perceive a post.

We replace attention module ℓ in the encoder with a different module for each social group g. For regularization, we train a separate *generic* attention module at the same time as the social-group attention, concatenate the social attention with the generic attention, and pass the concatenated attention through a linear layer to produce the final attention distribution. We choose the layer index $\ell = 1$ from $\{1, 3, 5\}$ through performance on validation data. For a question written by an asker who belongs to group g (gen indicates generic attention, f indicates a feed-forward linear layer), the attention is computed as follows:

$$\text{Multihead}_\ell(x) = f([\text{Multihead}_g(x); \text{Multihead}_{gen}(x)])$$

5.1.3 Social embeddings

For a *continuous* approach to personalization (Wu et al., 2021), we represent question-askers using latent embeddings $e^{(a)}$ based on their prior *subreddit* and *text* behavior.

For *subreddit* behavior, we compute the cross-posting matrix $\mathcal{P}$ for all subreddits and all question-askers in our data, where $P_{i,j}$ is equal to the NPMI of question-asker j writing a comment in subreddit i. We compress the matrix using SVD ($d = 100$), and the subreddit embedding $e_s^{(a)}$ for question-asker a is set to the average of the embeddings across all subreddits in which a previously posted. For *text*, we compute an embedding based on the question-asker's previous comments. We train a `Doc2Vec` model $\mathcal{D}$ (Le and Mikolov, 2014) on all prior comments and represent each comment as a single document embedding ($d = 100$, default skip-gram parameters). The text embedding $e_t^{(a)}$ for question-asker a is computed as the average over all prior comments.

To add the social embedding to the input text, we pass $e^{(a)}$ through a linear layer to match the text dimensionality ($d = 768$). We append a special "social embedding" token and the embedding $\hat{e}^{(a)}$ to the end of the text input.

5.2 Results

We use the models proposed above and a text-only baseline, and train them on the same task of question generation. We use a sample of our data for training/testing, for a total of 155396 questions for training, 51774 for validation, and 53080 for test.

We use the following metrics to automatically evaluate text quality for target question q and generated question $\hat{q}$: BLEU-1 (single word overlap between q and $\hat{q}$); perplexity; BERT Distance (cosine distance between sentence embeddings for q and $\hat{q}$, via the same DistilBERT system used throughout; Sanh et al. 2019); Type/token ratio (among bigrams in $\hat{q}$); Diversity (% unique questions among all generated questions $\hat{\mathcal{Q}}$); Redundancy (% generated questions $\hat{\mathcal{Q}}$ that also appear in training data $\mathcal{Q}_{\text{train}}$). The text overlap metrics like BLEU are important in judging performance even in our open-domain setting, because the models should produce questions that are faithful to the original intent of the question-askers (Wu et al., 2021). Without measuring overlap, it would be possible for a socially-aware model to generate highly diverse questions that are completely unrelated to the question-asker's intent.

5.2.1 Aggregate results

The aggregate results are shown in Table 5. Overall, we see that the simpler socially-aware models (tokens and attention) perform roughly the same as the text-only model via traditional BLEU and BERT Distance metrics. The socially-aware model generates questions that have higher overall diversity, but also higher perplexity. These results echo prior work in text generation which finds that models which incorporate pragmatic information often produce more diverse text than expected (Schüz et al., 2021). The higher perplexity can be explained

stat	BLEU-1 ↑	BERT Dist. ↓	Diversity ↑	Type/token ↑	Redundancy ↓	PPL ↓
Text-only	**0.159**	**0.728**	0.613	0.122	**0.187**	**264**
Social token	0.159	0.731	0.675	**0.127**	0.191	271
Social attention	0.157	0.752	0.511	0.068	0.468	488
Subreddit embedding	0.153	0.746	**0.744**	0.091	0.277	657
Text embedding	0.154	0.745	0.732	0.090	0.292	609

Table 5: Question generation results by model on full test data. ↑ means higher score is better, ↓ means lower is better.

Subreddit	LegalAdvice	AmITheAsshole
Text context	My five year old son is in kindergarten. The teacher let the kids out of their recess area and did not watch them properly, and my son got lost.	My roommate has been dating someone with a young child. Both the woman and her child are generally annoying.
Social group	LOCATION (US)	EXPERTISE (Novice)
Actual question	What is your location?	Have you talked to your roommate?
Text-only	What are your damages?	Have you spoken to your roommate about this?
Social token	Was this a private school or a government agency?	Do you and your roommate pay rent to the landlord?
Model performance	social token > text-only (BERT Dist.)	text-only > social-token (BLEU)

Table 6: Example posts, target questions, and generated questions.

partly by the unconstrained nature of the generation task (e.g., not providing an answer to generate the question; see § 6.2) as well as the relatively complex nature of most of the questions.

5.2.2 Qualitative analysis of model output

We first show several examples of generated text (Table 6). In a legal context (first column), the social-token model correctly predicts that the question-asker will focus on the location of the incident rather than the outcome (text-only model), possibly because a US question-asker may have location-specific advice to provide.

We also use the social-token model to generate attention distributions over the input sequence for different groups. We input the same text for both reader groups in the same category, changing only the social token appended to the text. We compute the attention distribution from the first layer of the encoder, compute per-word attention scores via the mean over all heads and all token-pairs, and compute the ratio of attention for each group category. The distributions for an example post are shown in Table 7, and they seem to match our earlier findings with word category differences (§ 4.2). For LOCATION, we see that the model prompted with a US token pays more attention to MONEY words ("booked tickets"), while the model prompted with NONUS focuses on time-related words that could be translated to FOCUSPRESENT in the question ("happened," "'few days ago"). For Expertise, the NOVICE social token produces higher attention on social relationships ("friend," "daughter"), and the model with EXPERT input attends to pronouns that could be converted to "you" pronouns in a following question ("my"). For TIME, the model with SLOW input pays attention to DRIVE words ("planning," "looking"), while the model with FAST input pays more attention to personal pronouns ("I", "my"). While we do not perform large-scale annotation of attention distributions, the examples shown here complement the generated text and reveal potential concepts that the model has learned to associate with different social groups.

5.2.3 Divisive posts

Socially-aware question generation should perform well in cases where different social groups have divergent opinions, e.g. where experts disagree with novices. We now test the models' ability to predict divisive questions. For post p, question q_1 written by an author of group 1, and question q_2 written by an author of group 2, we define $\text{sim}(q_1, q_2)$ based on the cosine similarity of the latent representations of the questions, generated by DistilBERT as before (Sanh et al., 2019). We label as "divisive" all pairs of questions that have a similarity score in the lowest n^{th} percentiles. We show examples of divisive posts in § C.3.

The results of the question prediction task on divisive posts are shown in Table 8. The social-token model slightly outperforms the text-only baseline for questions that are highly dissimilar (i.e. less similar than 90% and 95% of the question pairs), and all socially-aware models tend to do better in diversity. This suggests that the social-token model may pick up information specific to the different social groups that is required to anticipate how the question-askers approach potentially subjective posts. We also note the unusually high perplexity across all models,

EXPERTISE EXPERT, NOVICE	So my friend is having difficulty getting her 15 year old daughter to school . My friend will let her off at school , watch her enter the building , and then later will find her back home during school time .
LOCATION NONUS, US	This happened a few days ago and my friend thought I was a bit rude , but I felt I was totally justified . So we booked tickets for a nearly full flight and the only row with 2 seats beside each other had somebody that already booked the seat...
TIME SLOW, FAST	Folks , I am planning to return to PCs after an absence . my budget is about 3 k and I already found a machine that will be around 2 , 5 k . So right now I am searching for monitors and I am looking for...

Table 7: Ratio of encoder attention generated by social-token model for input conditioned on different social groups. Attention computed via mean over all pairwise scores between tokens.

Model	BLEU-1	Div.	Red.	PPL
sim(q_1, q_2) $\leq$ 5% (N=1074)				
Text-only	0.137	0.688	0.222	383
Social token	**0.142**	0.771	**0.208**	**359**
Social attention	0.130	**0.875**	0.479	601
Subreddit emb.	0.137	0.854	0.292	945
Text emb.	0.137	0.840	0.375	623
sim(q_1, q_2) $\leq$ 10% (N=2146)				
Text-only	0.160	0.699	**0.232**	**325**
Social token	**0.164**	0.781	0.235	327
Social attention	0.155	0.798	0.500	547
Subreddit emb.	0.148	0.864	0.308	1048
Text emb.	0.150	**0.868**	0.348	617

Table 8: Question generation results for divisive posts.

which may indicate that socially-specific questions are complicated and far from "normal" questions.

5.2.4 Group-specific questions

We investigate another desired property of socially-aware models, the ability to predict questions that are strongly associated with a particular group. Post writers would benefit from such questions, e.g. technical questions from "expert" askers, because these questions would help the post writer preempt specific and unexpected information needs from that group. We subset the data to all questions q with question-asker a that the trained social group classifiers assign to the group g_a with high confidence ($P \geq 95\%$) (see § B.2 for classifier details).

We report the results for this data subset in Figure 1. The relative performance of the socially-aware models increases when only considering data with highly group-specific questions. This is particularly apparent for the LOCATION group category, illustrated by the following example. In our data, a socially-specific question was written by a non-US question-asker in LegalAdvice in response to a post about a mailing problem: "Have you sent a change of address notice to the post office?" In this situation, the social-token model generated the question "Did you give them your current address?" The social-token model seems to have identified a concern that a non-US question-asker might be more likely to focus on (e.g. due to moving frequently) than a US question-asker.

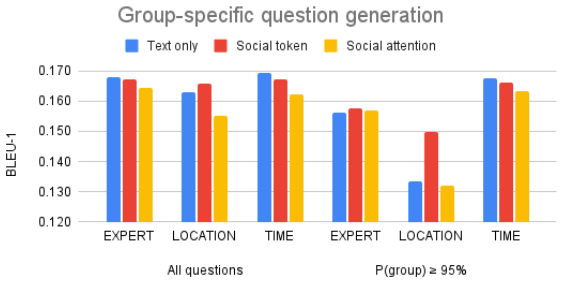


Figure 1: BLEU-1 scores for question generation, on (1) full data and (2) subset of data with high group-specific probability (determined by classifier).

5.3 Human evaluation

To corroborate the generation results about divisive questions, we collect human annotations about: (1) question quality; and (2) guessing the social group based on the generated question (see § C.6 for (2)).

We use the text-only model and the social-token model to generate questions from a sample of the test data, as follows. For each subreddit s and social group category $\mathcal{G}$, we sample up to $N = 10$ posts that have divisive questions from groups g_1 and g_2 where the similarity is below the 10^{th} percentile (§ 5.2.3).[3] We then generate a single question for the post from the text-only model and two questions from the social-token model, one for each social group in the category (e.g., for EXPERTISE, q_1 for Expert and q_2 for Novice). We provide details

[3]Some combinations of subreddits and reader groups have fewer than 10 posts, due to the data sampling strategy.

Text type	A	R	U
Overall			
Ground-truth	3.83	3.59	3.92
Text-only	3.84	3.68*	3.96*
Social-token	3.80	3.35	3.73
Social group			
EXPERTISE			
Ground-truth	3.89	3.68	3.85
Text-only	3.81	3.62*	3.91*
Social-token	3.61	2.99	3.49
LOCATION			
Ground-truth	4.06	3.58	4.20
Text-only	4.01	3.69	4.05
Social-token	4.20	3.63	4.19
TIME			
Ground-truth	3.64	3.52	3.83
Text-only	3.77	3.74	3.95*
Social-token	3.74	3.53	3.70
Subreddit			
Advice			
Ground-truth	3.75	3.63	3.91
Text-only	3.32	3.29	3.57
Social-token	3.49	3.15	3.67
AmItheAsshole			
Ground-truth	3.79	3.58	4.01
Text-only	3.74	3.61	3.89
Social-token	3.82	3.39	3.69
LegalAdvice			
Ground-truth	4.18	3.88	4.47
Text-only	3.95	3.60	4.19*
Social-token	3.86	3.23	3.81
PCMasterRace			
Ground-truth	3.72	3.44	3.62
Text-only	4.20	4.07*	4.16
Social-token	3.98	3.39	3.84
PersonalFinance			
Ground-truth	3.72	3.43	3.58
Text-only	4.04	3.89	4.01*
Social-token	3.87	3.56	3.70

Table 9: Human annotation scores for question quality, including **A**nswerable, **R**elevant, **U**nderstandable (scale 1-5). * indicates that the score is greater than the scores from the other model type with $p > 0.05$ (Wilcoxon test). Underline indicates best generation model.

of annotation in § C.5.

We show the results in Table 9. The annotators in aggregate preferred the questions from the text-only model over the social-token model. However, the social-token model questions were perceived as more answerable and understandable for questions generated using LOCATION information, which aligns with prior results (§ 5.2.4). The social-token model is also perceived as more answerable and understandable in the context of Advice, which makes sense considering that the social-token model has more diverse output that may suit the broad domain of general-advice posts.

We show example generated and actual questions with their human evaluation ratings in Table 10. In the first example, the text-only model addresses an important missing gap in original post (how the scammer got information), while the social-token model seems to focus too much on details ("on the hook") which leads to a less understandable question. In the second example, the social-token model addresses missing information that may be more salient to a non-US question-asker who wants to know more about homelessness (possibly less salient to a US question-asker), while the text-only model produces a question that is not answerable due to a misunderstanding of the original post (focus on the text rather than the writer). Note that this type of question is not marked by a surface-level feature such as regional style, but rather a deeper focus on cause and effect, which suggests that the model has learned more fundamental differences about the nature of LOCATION as a social group.

Subreddit	LegalAdvice	Advice
Text context	My mother lost $50000 on an online dating site to a scam. If something happened to her, would I be on the hook for this?	I want to break up with my girlfriend but: number 1 I don't want to hurt her, number 2 I don't know if I can manage on my own, number 3 I don't always believe in myself, and if I lose my job I'll be homeless.
Social group	EXPERTISE (Expert)	LOCATION (non-US)
Actual question	Has your mother contacted the police? (Understandable=4.67)	Have you tried talking to her? (Answerable=5)
Text-only model	How did the scammer get the info from your Mom? (Understandable=4.33)	Number 2 doesn't even sound like a good idea, have you tried number 3? (Answerable=2.33)
Social token model	Are you on the hook for what? (Understandable=2.67)	Why do you think you'll be homeless? (Answerable=5)

Table 10: Example questions with human evaluation scores.

6 Conclusion

6.1 Discussion

This study evaluated the incorporation of social information into question generation, to help writers understand the information needs of different people. We found that social groups related to expertise, time, and location can all be differentiated based on the questions that they ask. In generation, the discrete social representations outperformed continuous representations, and the social-token model outperformed the baseline when the questions are divisive. In human evaluation, the social-token models produced better output for the LOCATION group, implying a more clear definition versus other social groups.

Future research in question generation should focus on divisive questions as the main area of improvement. Researchers may also consider ensemble models (Liu et al., 2021) that use a text-only generator with less subjective input text (e.g., in technical settings), and a social-token generator in more divisive settings. For future evaluation, socially-aware question generation may benefit other contexts such as journalism, medicine, and public policy, where people are likely to have differing information needs based on their background experience (Assmann and Diakopoulos, 2017). No matter the case, writers will always benefit from knowing in advance what information their audience will need.

6.2 Limitations

The primary limitations of this work relate to the definition of "social group," which may have contributed to the minimal gains by the social token model. This work focused on generic social groups that can be extended to other domains, which may leave out domain-specific social groups (e.g., socioeconomic status). The social groups may not mean the same thing in different domains: an `EXPERT` question-asker in the legal domain may be a professional lawyer, while in personal advice the average `EXPERT` may lack professional experience. Most notably, the social groups used in this work were not validated by any annotators or by the question-askers. This especially matters for the `EXPERTISE` category, considering the subjective status of expertise within online communities (Johnson, 2001). To accurately identify non-obvious social groups, researchers should ask domain experts to label a small set of user data as gold labels, and then compare the automated labels against this gold standard set.

In terms of the task, this work focuses on unconstrained question generation, i.e. we do not use answers (Dong et al., 2019) or intentions (Cao and Wang, 2021) to guide generation. The results presented in this work represent a lower bound on performance, which includes unusually high perplexity (Table 5) and sometimes unexpected topic choices (Table 6). This problem is compounded by the fact that social group information may not always be useful e.g. for non-divisive questions, and therefore such social guidance may simply confuse the model. Future work would collect both questions and answers, or at least question type labels, to provide consistent guidance for socially-aware question generation.

7 Ethics statement

We acknowledge that text generation is an ethically fraught application of NLP that can be used to manipulate public opinion (Zellers et al., 2020) and reinforce negative stereotypes (Bender et al., 2021). Our models could be modified to generate abusive or factually misleading questions, which we do not endorse. Furthermore, our models may accidentally memorize private information from the training data. We intend for our work to benefit people who share information about themselves for the purpose of gaining feedback from peers.

All data used in this project was publicly available via the Pushshift API (Baumgartner et al., 2020). In our final release we will not release any data with personally identifiable information (e.g., `LOCATION` data), in order to protect the original authors. This is not ideal considering that `LOCATION` seemed to be the most useful input to the model, but the remaining social attributes may prove useful for future researchers who want to test other definitions of "divisive" questions, e.g. positive versus negative valence. Furthermore, we do not claim that we have the perfect definitions of the social groups that we attempted to identify in our study, and it is possible that a Reddit user who finds themselves labeled as e.g. an "expert" would disagree. We encourage future researchers to compare their own definition of the various social groups against our own labels, e.g. a different definition of "expertise."

Acknowledgments

This material is based in part on work supported by the John Templeton Foundation (grant #61156) and by the Michigan Institute for Data Science (MIDAS). Any opinions, findings, conclusions, or recommendations in this material are those of the authors and do not necessarily reflect the views of the John Templeton Foundation or MIDAS. We thank members of the LIT Lab, including Artem Abzaliev and Aylin Gunal, for their help piloting the human annotation task, and for providing feedback on early results. We also thank the annotators who identified valid questions as part of the data filtering process.

References

Karin Assmann and Nicholas Diakopoulos. 2017. Negotiating change: Audience engagement editors as newsroom intermediaries. In *International symposium on online journalism (ISOJ)*, pages 25–44.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *ICWSM*, volume 14, pages 830–839.

Lee Becker, Sumit Basu, and Lucy Vanderwende. 2012. Mind the gap: Learning to choose gaps for question generation. In *NAACL HLT 2012 - 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 742–751.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Shuyang Cao and Lu Wang. 2021. Controllable open-ended question generation with a new question type ontology. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6424–6439.

Marco Del Tredici, Diego Marcheggiani, Sabine Schulte im Walde, and Raquel Fernández. 2019. You shall know a user by the company it keeps: Dynamic representations for social media users in nlp. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4701–4711.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *NeurIPS*.

Xinya Du and Claire Cardie. 2018. Harvesting paragraph-level question-answer pairs from wikipedia. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, pages 1907–1917.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:1342–1352.

Lucie Flek. 2020. Returning the n to nlp: Towards contextually personalized classification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7828–7838.

Yifan Gao, Lidong Bing, Wang Chen, Michael R. Lyu, and Irwin King. 2019. Difficulty controllable generation of reading comprehension questions. In *IJCAI*, pages 4968–4974.

Aparna Garimella, Carmen Banea, Dirk Hovy, and Rada Mihalcea. 2019. Women's syntactic resilience and men's grammatical luck: Gender-bias in part-of-speech tagging and dependency parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3493–3498.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1373–1378.

Dirk Hovy. 2018. The social and the neural network: How to make natural language processing about people again. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 42–49.

Dirk Hovy and Shannon L Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598.

Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, and Kentaro Inui. 2019. Diamonds in the Rough: Generating Fluent Sentences from Early-Stage Drafts for Academic Writing Assistance. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53.

Christopher M Johnson. 2001. A survey of current research on online communities of practice. *The internet and higher education*, 4(1):45–60.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *EACL*, pages 427–431.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Vaibhav Kumar and Alan W. Black. 2020. ClarQ: A large-scale and diverse dataset for Clarification Question Generation. In *ACL*, pages 7296–7301.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-Time Controlled Text Generation with Experts and Anti-Experts. In *ACL*, pages 6691–6706.

Ming Liu, Rafael A. Calvo, and Vasile Rus. 2012. G-Asks: An Intelligent Automatic Question Generation System for Academic Writing Support. *Dialogue & Discourse*, 3(2):101–124.

Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. 2017. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1260–1272.

Woojin Paik, Sibel Yilmazel, Eric Brown, Maryjane Poulin, Stephane Dubon, and Christophe Amice. 2001. Applying natural language processing (nlp) based metadata extraction to automatically acquire user preferences. In *Proceedings of the 1st international conference on Knowledge capture*, pages 116–122.

Shimei Pan and Tao Ding. 2019. Social media-based user embedding: A literature review. In *IJCAI*.

Douglas B Park. 1986. Analyzing audiences. *College Composition and Communication*, 37(4):478–488.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Peng Qi, Yuhao Zhang, and Christopher D Manning. 2020a. Stay hungry, stay focused: Generating informative and specific questions in information-seeking conversations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 25–40.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020b. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108.

Sudha Rao and Hal Daumé. 2019. Answer-based adversarial training for generating clarification questions. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:143–155.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *EC2*.

Simeon Schüz, Ting Han, and Sina Zarrieß. 2021. Diversity as a by-product: Goal-oriented language generation leads to linguistic variation. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 411–422.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Charles Welch, Jonathan K Kummerfeld, Verónica Pérez-Rosas, and Rada Mihalcea. 2020. Compositional demographic word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4076–4089.

Yuwei Wu, Xuezhe Ma, and Diyi Yang. 2021. Personalized response generation via generative split memory network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1956–1970.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. TWEETQA: A Social Media Focused Question Answering Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5020–5031.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2020. Defending against neural fake news. In *NeurIPS*.

Justine Zhang, James Pennebaker, Susan Dumais, and Eric Horvitz. 2020. Configuring audiences: A case study of email communication. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW1):1–26.

A Data: question filtering

Initial analysis revealed that some questions were either irrelevant to the post (e.g., "what about X" where X is unrelated to the post topic) or did not actually seek more information from the original post (e.g., rhetorical questions). To address this, we sampled 100 questions from each subreddit in the data along with the parent post, and we collected binary annotations for relevance ("question is relevant") and information-seeking ("question asks for more information") from three annotators, who are undergraduate students and native English speakers. We provided instructions and a sample of 20 questions labeled by one of the authors as training data for the annotators. On the full data, the annotators achieved fair agreement on question relevance ($\kappa = 0.56$) and on whether questions are information-seeking ($\kappa = 0.62$).

After annotation, we removed all instances of disagreement among annotators to yield questions with perfect agreement for relevance (76% perfect-agreement) and information-seeking (80%). In the perfect-agreement data, the majority of questions (94%) were marked as relevant by both annotators, which makes sense considering that the advice forums generally attract good-faith responses from commenters. We therefore chose to not filter questions based on potential relevance. To filter information-seeking questions, we trained a simple bag-of-words classifier on the annotated data (binary 1/0; based on questions with perfect annotator agreement).[4] The annotated data were split into 10 folds for training and testing, and the model achieved 87.5% mean F1 score, which is reasonable for "noisy" user-generated text. We applied the classifier to the full dataset and removed questions for which the classifier's probability was below 50%.

B Defining social groups

B.1 Social embeddings: topically-related subreddits

In our discrete-representation models, the criterion for defining a question-asker for post p in subreddit s as an `Expert` or `Novice` is whether they have previously written comments in s or in a topically similar subreddit.

We find similar subreddits for each target subreddit s by (1) computing the top-20 nearest neighbors for subreddit s in subreddit embedding space (see § 5.1.3) and (2) manually filtering unrelated subreddits. We report the related subreddits in Table 11.

Subreddit	Neighbors
`Advice`	`answers, ask, askdocs, dating_advice, getdisciplined, mentalhealth, needadvice, socialskills, tipofmytongue`
`AmItheAsshole`	`askdocs, isitbullshit, tooafraidtoask`
`LegalAdvice`	`askhr, bestoflegaladvice, insurance, landlord, lawschool, legaladviceuk, scams`
`PCMasterRace`	`bapcsalescanada, buildmeapc, linuxmasterrace, monitors, overclocking, pcgaming, suggestalaptop, watercooling`
`PersonalFinance`	`accounting, askcarsales, churning, creditcards, financialindependence, financialplanning, investing, realestate, smallbusiness, studentloans, tax, whatcarshouldibuy, yna`

Table 11: Filtered neighbor subreddits for advice-related subreddits.

B.2 Validating group differences: classification

To verify the differences in question content observed in § 4.2, we train a single-layer neural network to classify social groups, using a latent semantic representation of the question-asker's question q and the related post p generated by the DistilBERT transformer model (Sanh et al., 2019). The embedding for the question and the post are each converted to $d = 100$ dimensions via PCA for regularization, and then concatenated. We train a separate model for each subreddit, and we

[4] We restricted the vocabulary to the 50 most frequent words, minus stop-words, to avoid overfitting. Initial tests with SVM, logistic regression, and random forest models revealed that the random forest model performed the best, which we used for the final classification model.

Features	Social group	Accuracy
Question text	EXPERTISE	70.1 (± 2.5)
	TIME	81.6 (± 7.5)
	LOCATION	75.4 (± 2.8)
Post + question text	EXPERTISE	73.5 (± 6.4)
	TIME	83.1 (± 8.3)
	LOCATION	66.3 (± 10.2)

Table 12: Social group prediction accuracy (mean, standard deviation measured across subreddits).

up-sample data from the minority class.

We report mean accuracy over all subreddits in Table 12. The models consistently outperform the random baseline across all group categories tested, which suggests a clear difference between social group members. The models trained on the combined post and question text generally help prediction improve over the question text alone, which supports the hypothesis that a question-asker's background is reflected in both the question they ask and the context in which the question is asked. Therefore, generating group-specific questions requires understanding how the question relates to the original post content, in addition to the question writing style. We find an unusually high performance for TIME, which may be due to a more consistent writing style among Fast question-askers.

C Results: question generation

We report here the results of additional tests to evaluate the relative utility of the socially-aware models with respect to different types of question-post scenarios.

C.1 Performance by question type

First, we assess the relative performance of different question generation models according to the type of question asked. Questions are categorized based on the root question word, e.g. "who," "what," "when."[5] We compare the BLEU-1 scores of all question generation models on the specified questions, restricting to questions asked by question-askers who could be assigned to at least one social group or an embedding.

The results are shown in Figure 2. In contrast to the aggregate results, the social-attention model outperforms the text-only baseline for "do," "where," and "who" questions. All socially-aware models outperform the text-only model for "when" questions. These questions may reflect more of a focus on concrete details such as locations, times, and people mentioned by the original poster, and therefore the socially-aware models may generally identify differences among question-askers in terms of the details requested. The text-only model outperforms the socially-aware models for questions that are potentially more subjective, including "can," "could," "would," and "should" questions. These more subjective questions may require the models to focus more precisely on the original post (e.g. a "would" question to pose a hypothetical concern about the post author's situation), and therefore such questions may be less dependent on question-asker identity.

C.2 Post similarity

A helpful question should be related to the original post, but should not be so similar that it requests information that the post has already provided. We therefore assess the tendency of the models to generate semantically related questions for the given posts. We compute the similarity between each generated question q and the associated post p using the maximum cosine similarity between the sentence embedding for q and each sentence s in p. The sentence embeddings are generated using the DistilBERT model (Sanh et al., 2019).

The results in Figure 3 show that the best overall models, text-only and social-token, generate questions that are more similar to the original post than expected (cf. "target text" i.e. ground-truth). The other socially-aware models show a significantly lower similarity, implying that their generated questions address *new information* about the post that is not mentioned in the post itself. For example, in response to a r/Advice post about self-improvement ("I just need some tips on maybe motivating myself"), the model with social text embeddings asks "What do you want to do with your life?" The generated question is less semantically similar to the original post than the target question ("Have you talked to a doctor about this?") but addresses an underlying personal issue for the post author that only a particularly thoughtful question-asker would uncover.

C.3 Divisive questions: examples

We provide examples of divisive posts in Table 13 (§ 5.2.3). For TIME, the Slow question-asker seems to target a more complicated and underlying

[5] We use the dependency parser from spacy (Honnibal and Johnson, 2015) to identify root question words based on their dependency to the root verb of the question (e.g. advmod for "where" in "where do you live?").

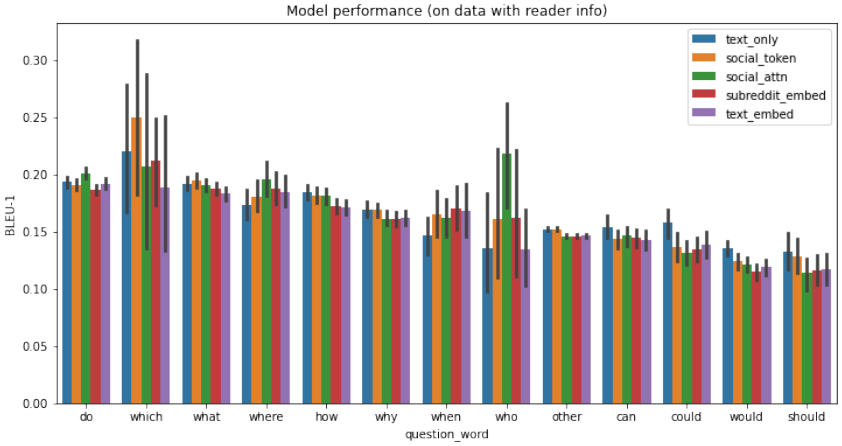


Figure 2: Model performance by question type.

Subreddit	PersonalFinance	LegalAdvice	AmITheAsshole
Text context	I need help figuring out what's the best next step. I have $1200 saved for car payments but I have no idea after that.	Last month I got a letter from a law firm representing someone that I owe a debt to. Two years ago I couldn't continue to make payments to the creditor and almost went bankrupt.	My younger brother is autistic. He can function and he has a job (janitor), hangs out with his friends but he can't live on his own.
Social group	EXPERTISE	TIME	LOCATION
Group 1	(Expert) Have you been applying for jobs all day?	(Slow) Have you asked what they are willing to settle for?	(US) What if down the road you had to re-locate for work or your wife's work?
Group 2	(Novice) Are you above water on the car?	(Fast) Do you actually intend on filing bankruptcy?	(non-US) How disabled is your brother?
Question similarity	0.209	0.256	0.190

Table 13: Example divisive questions for different social groups.

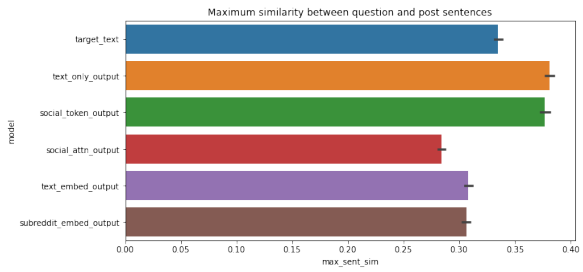


Figure 3: Maximum semantic similarity between questions and sentence from original post.

issue around the debt problem, while the Fast question-seeker clarifies a basic detail about the case. For LOCATION, the question from the US asker focuses on adapting to work needs, while the non-US question addresses the writer's brother and his medical situation. In all cases, we can see that these kinds of questions are more likely to be anticipated by a generation model that produces more diverse output.

C.4 Divisive posts: word embeddings

In § 5.2.3, we identified questions as "divisive" based on low similarity between the latent representations of the questions, as generated by a sentence encoder. We also experiment with determining divisiveness based on static word embeddings. We leverage a set of word embeddings trained with the FastText algorithm (Joulin et al., 2017), and we convert each questions to a latent representation using the average over all embeddings for the tokens in the question. We then compute paired question similarity as before, with cosine similarity. The questions from the sentence embeddings and those from the static word embeddings have a high degree of overlap: setting the similarity threshold below 5% yields an overlap of 23.7%, and a similarity threshold below 10% yields an overlap of 45.3%. Next, we test the correlation between the sentence embedding similarity and the word embedding similarity and find a high amount of correlation ($R = 0.98$, $p < 0.001$). We

Data	Accuracy
Overall	47.5
Social group	
EXPERTISE	49.3
LOCATION	60.8
TIME	36.9
Subreddit	
Advice	45.6
AmItheAsshole	48.9
LegalAdvice	53.3
PCMasterRace	42.9
PersonalFinance	47.8

Table 14: Human annotation accuracy for group guessing task.

conclude that labeling divisive questions using word embedding similarity rather than sentence embedding similarity would yield similar results to those observed earlier.

C.5 Human evaluation: annotation details

We provide the details of the annotation required for the human evaluation task (§ 5.3). We annotate the questions for each combination of subreddit and group category, and we recruit 1 annotator per task via Prolific, with 3 social groups $\times$ 5 subreddits $\times$ 3 annotators = 45 annotators total, and a maximum of 50 questions total for each annotator. For domain-specific subreddits, we recruit annotators based on profession, e.g. annotators who work in the finance industry for r/PersonalFinance. We pay our annotators $5 for the task, assuming about 30 minutes per task. Annotators judged question quality on a 5-point scale based on whether they were answerable, relevant, and understandable. The annotators achieved reasonable agreement considering the subjective nature of the task, with Krippendorff's alpha at 0.153 for "Answerable," 0.309 for "Relevant," and 0.23 for "Understandable" (compared to 0 for random chance).

C.6 Human evaluation: social group prediction

We report here the results of the additional annotation task mentioned in § 5.3. Following the question quality task, for each post we provide the two social-token model questions in random order for a group prediction task, where annotators must choose the question that corresponds to a given social group in the category: e.g. "Which question was more likely to be written by an **expert** reader?" We show the results for the group-guessing task in Table 14. Annotators generally had trouble guessing the identity of the social groups except for the LOCATION category, which corresponds with the higher quality ratings reported in Table 9. We also find slightly higher guessing accuracy for LegalAdvice, which may be due to intuitive understanding among annotators on what constitutes a difference in social groups for the legal domain (e.g. experts using particular terminology). The low performance in this task may indicate that human-understandable differences between the questions may be less obvious in individual pairs of questions as compared to the aggregate groups of questions (see differences in § 4.2).

GenTUS: Simulating User Behaviour and Language in Task-oriented Dialogues with Generative Transformers

Hsien-Chin Lin, Christian Geishauser, Shutong Feng, Nurul Lubis, Carel van Niekerk, Michael Heck and **Milica Gašić**
Heinrich Heine University Düsseldorf, Germany
{linh,geishaus,shutong.feng,lubis,niekerk,heckmi,gasic}@hhu.de

Abstract

User simulators (USs) are commonly used to train task-oriented dialogue systems (DSs) via reinforcement learning. The interactions often take place on semantic level for efficiency, but there is still a gap from semantic actions to natural language, which causes a mismatch between training and deployment environment. Incorporating a natural language generation (NLG) module with USs during training can partly deal with this problem. However, since the policy and NLG of USs are optimised separately, these simulated user utterances may not be natural enough in a given context. In this work, we propose a generative transformer-based user simulator (GenTUS). GenTUS consists of an encoder-decoder structure, which means it can optimise both the user policy and natural language generation jointly. GenTUS generates both semantic actions and natural language utterances, preserving interpretability and enhancing language variation. In addition, by representing the inputs and outputs as word sequences and by using a large pre-trained language model we can achieve generalisability in feature representation. We evaluate GenTUS with automatic metrics and human evaluation. Our results show that GenTUS generates more natural language and is able to transfer to an unseen ontology in a zero-shot fashion. In addition, its behaviour can be further shaped with reinforcement learning opening the door to training specialised user simulators.

1 Introduction

Task-oriented dialogue systems (DSs) assist their users in accomplishing a goal, such as booking a flight ticket or making a payment. This should be done through natural language interactions between the system and the user, whilst the system interacts with various external databases and API calls in the background. The core component of such a DS is the dialogue policy module, which decides what should be said to the user next. This module can be trained via interaction with users, through reinforcement learning (RL). However, this creates a conflict between the high cost of interacting with real users and the large amount of interactions required for RL. As a result, user simulators (USs) are often utilised instead to train dialogue policies, as they make it possible for the system to learn from a large number of interactions in a controlled environment at a fraction of the cost.

Rule-based USs are widely used both in research and industry because they are interpretable and can be built without a labelled dataset. However, designing the rules demands expert knowledge and creating these rules becomes intractable on complex domains, making them only suitable for small and simple domains. In addition, human behaviour is too complex and diverse to be manually described by rules, leading to sub-optimal performance of DSs in deployment scenarios (Schatzmann et al., 2006).

On the other hand, data-driven USs can be built with less expert involvement. However, these models are either ontology-dependent (El Asri et al., 2016; Gür et al., 2018; Kreyssig et al., 2018), which means adapting to a new domain requires re-engineering the feature representation or re-training the model, or they do not model the language of the user (Lin et al., 2021). Both shortcomings are serious. The user simulator needs to support zero-shot transfer across ontologies, as it is difficult to collect enough labelled data for each new domain. The ability to produce natural language output is also critical as it makes the training and testing environment more challenging and similar to the real user scenario. Therefore, models that can attain both properties are much needed.

In this work, we propose a model that has both desired properties. More specifically, our contributions are as follows:

- We, propose a **gen**erative **t**ransformer-based

Proceedings of the SIGdial 2022 Conference, pages 270–282
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

user **s**imulator that we call *GenTUS*[1]. The response of *GenTUS* includes both semantic actions and natural language utterances, which retains interpretability and induces linguistic variation.

- By optimising the user policy and natural language jointly, GenTUS generates more natural language in the given context.
- GenTUS can adapt to an unseen ontology in a zero-shot fashion and have its behaviour further shaped by reinforcement learning (RL).

The rest of the paper is organised as follows. In Section 2, we review the related work. Section 3 describes in detail the proposed simulation framework. In Section 4, we present the experimental set-up, followed by the experimental results in Section 5. We conclude with Section 6.

2 Related Work

The performance of a task-oriented dialogue policy trained by RL is significantly affected by the quality of the US used to generate the interactions (Schatzmann et al., 2005). An N-gram user simulator proposed by Eckert et al. (1997) is one of the earliest data-driven models. This model predicts the user action a_u according to the system action a_m based on a bi-gram model $P(a_u|a_m)$. Its behaviour is often unreasonable since it only takes the latest system action as input without any information about the user goal. Therefore, models which can act on a given user goal were introduced (Georgila et al., 2006; Eshky et al., 2012). A Bayesian user simulation model which predicts the user action based on the user goal is proposed by Daubigney et al. (2012). In Cuayáhuitl et al. (2005), the user and the system behaviour are modelled by hidden Markov models. A graph-based US, which constructs a graph from all possible dialogue paths, is proposed by Scheffler and Young (2002). This simulator can act reasonably and consistently, but it is not practical to implement in a complex scenario, as it requires extensive domain knowledge.

The agenda-based user simulator (ABUS) (Schatzmann et al., 2007) is widely used to train tourist-information DSs. Its behaviour is based on hand-crafted stacking and popping of rules with a stack-like agenda user goal, ordered by the priority of the user actions. It is difficult to transfer this model to a new ontology because the rules need to be redesigned. Moreover, it only provides semantic-level dialogue acts.

To reduce the involvement of experts, further data-driven user simulator approaches have been proposed. The sequence-to-sequence (Seq2Seq) model structure is the most common framework. A semantic level Seq2Seq user simulator with an encoder-decoder structure is proposed by El Asri et al. (2016). This model embeds the dialogue history into a context vector via a recurrent neural network (RNN) encoder. Its decoder then generates user actions based on the context embedding vector.

Instead of generating dialogue acts, the neural user simulator (NUS) of Kreyssig et al. (2018) can generate responses in natural language. However, this model has limited interpretability because it does not provide semantic-level outputs and its input representation is domain-dependent.

The variational hierarchical Seq2Seq user simulator (VHUS) proposed by Gür et al. (2018) encodes the system actions and the user goal by RNNs instead of complex dialogue history features and generates semantic user actions. Its features are still domain-dependent as system actions and user goals are represented by domain-dependent one-hot encodings. As VHUS has no constraints in the decoding process, it often generates impossible actions under the given ontology.

A domain-independent transformer-based user simulator (TUS) is proposed by Lin et al. (2021). With domain-independent input and output feature representations, TUS can adapt to an unseen domain in a zero-shot fashion. However, it does not model natural language output. Moreover, all intents are part of the model, which makes transfer to an unrelated ontology, i.e. the one with a different sets of intents, difficult.

To convert the dialogue acts from the semantic level to natural language, a user simulator commonly includes an NLG module connected to the semantic level user policy. Although template-based NLGs are widely used in research, creating templates for every dialogue act is labour-intensive and lacks language variation. Data-driven NLG models, such as SC-LSTM (Wen et al., 2015) and SC-GPT (Peng et al., 2020) can generate natural language utterances conditioned on given semantic actions. However, taking only semantic actions as input, their results may not be sufficiently natural in a given context. In addition, the user policy and

[1] https://gitlab.cs.uni-duesseldorf.de/general/dsml/gentus-public.git

NLG model cannot be optimised jointly within the modular architecture.

An end-to-end US which generates both dialogue acts and utterances is proposed by Tseng et al. (2021), although in their evaluation they train a DS using only the semantic actions from the US. The NLG of this US is based on a simple delexicalised LSTM model. The user goal is represented as a binary vector, with each dimension representing a domain-slot pair in the ontology. This creates several obstacles for transfer to an unseen ontology: such a transfer would require further hand-coded lexicalisation rules for the NLG component, modifications of the feature representations and further fine-tuning of the US policy.

3 Generative Transformer-based User Simulation

Task-oriented DSs are expected to handle the requests of real users in natural language. Therefore, when designing USs, it is important to endow them with the ability to converse with the system via natural language as well. In this way, we can study, for example, the robustness of the systems towards misunderstandings that may occur when conversing with real users. On the other hand, users rarely misunderstand the DS response. It is hence reasonable to assume that the input to the US may be on the semantic level. This is also practical in such cases as when DSs need to execute API calls, such as playing a song or turning off the light.

Task-oriented DSs are built upon an *ontology* which includes all possible *intents* that the user or the system can exhibit in their actions and *domains*, which describes the entities the user or the system can talk about. Domains are further characterised by a number of *slots* and each slot can take a number of *values*. In task-oriented DS we assume that the user has a particular goal they want to achieve. We define goal as the following set $G = \{domain_1 : [(slot_1, value_1), (slot_2, value_2), \dots], domain_2 : [(slot_3, value_3), \dots], \dots\}$, where domains, slots and values are selected from the ontology.

The semantic *user action* and *system action* are composed of several tuples of the following structure: $(intent, domain, slot, value)$. Users and systems may have different intents, e.g., systems can *recommend* an option and users can *negate* the recommended offer. A semantic action can be converted into a natural language utterance, which we denote with $text_{usr}$ in the case of a user action.

User simulation in a task-oriented dialogue can be modelled as a sequence-to-sequence problem. For each turn, GenTUS takes the context information as an input sequence, including the system action, the user history, the user goal, and turn information, and generates the semantic action and the natural language response as the output sequence. In following sections, we provide more details.

3.1 Model Structure

The backbone of the proposed GenTUS user simulation model is an encoder-decoder structure as shown in Fig. 1. In turn t, the user goal is updated by the user action from the previous turn and the current system action. If the system informs that the user's request is not possible or fails, the value of constraint slots will be replaced by a random value. The encoder takes the system action $action^t_{sys}$, user actions from previous 3 turns $action^{t-1:t-3}_{usr}$, the user goal $goal$, and the turn number t as input. Then the decoder generates both the user semantic action $action^t_{usr}$ conditioned on the output of the encoder and the associated natural language response $text_{usr}$. We initialise GenTUS by BART (Lewis et al., 2020), which is a transformer-based natural language generator with a bidirectional encoder and a left-to-right decoder. BART achieves convincing results on text generation and comprehension tasks after fine-tuning.

3.2 Input and Output Representation

The *system action* and *user action* are semantic level dialogue acts and are represented by a list of tuples $(intent, domain, slot, value)$. Note that the output of this user simulator is a semantic as well as a natural language representation of the user action. The natural language action is sent to the system, while the semantic action is retained by the user simulator for the next turn. The user goal $goal$ is represented by a list of tuples,

$$[(domain_1, type_1, slot_1, value_1, status_1), (domain_2, type_2, slot_2, value_2, status_2), \dots] \quad (1)$$

Following the setting in Lin et al. (2021), the tuples are ordered by the user preference, which means one tuple is in front of the others if the user prefer to mention it earlier. The *intent*, *domain*, *slot*, and *value* are sampled from the ontology. The *type* represents whether a slot in the goal is a constraint *info*, a request *reqt*, or a booking informa-

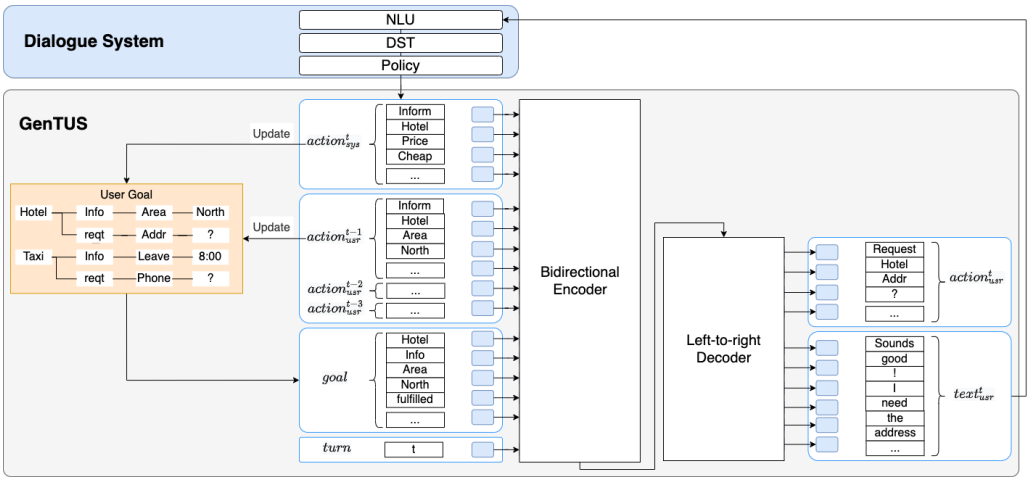


Figure 1: The model structure of GenTUS. Both input and output are JSON-formatted word sequences.

tion *book*. The $status$ represents the condition of each domain-slot pair. It can be *fulfilled*, *in conflict*, *requested*, or *not mentioned*. The turn information is the number of the current dialogue turn. We represent the input to GenTUS as a JSON-formatted string: "{"system": $action^{t}_{sys}$, "user": $action^{t-1:t-3}_{usr}$, "goal": $goal$, "turn": t}".

The output of GenTUS is a set of semantic-level user sub-actions and the corresponding utterance in natural language. The output is also easily represented as a JSON-formatted string: "{"action": $action^{t}_{usr}$, "text": $text^{t}_{usr}$}".

As ultimately both input and output contain only words, we can train GenTUS as a sequence-to-sequence model. By using a pre-trained language model for initialisation, we can harness the generalisation capabilities of these powerful models when adapting to a new ontology.

3.3 Constrained Semantic Decoding Space

The downside of using a large pre-trained language model as a generator is that it may suffer from generating hallucinations. This means that we should place constraints on the output to prevent generating illegal semantic actions, which is particularly problematic for DSs.

In order to only produce valid actions, every semantic action $(intent, domain, slot, value)$ is created by following a path in a graph that defines the valid actions, where the graph is constructed as follows. The possible *intents* in the diagram are derived from the ontology. For example, the MultiWOZ dataset (Budzianowski et al., 2018) contains general intents like *greeting* and *bye*, and domain-specific intents like *inform* and *request*. The possible *domains*, *slots* and *values* are derived from the user goal, and system actions are used to update the nodes. The possible paths following intent, domain, slot and value are constrained by the ontology, which defines what valid actions are comprised of. Fig. 2 depicts an example, where GenTUS selected the action `[(Inform, Hotel, Area, North)]` in turn 0 and `[(Request, Hotel, Addr, ?), (Inform, Taxi, Leave, 8:00)]` in turn 1 by following the two paths in the diagram. The graph derived from the user goal is depicted on the left of Fig. 2 and updated after the system asked about a cheap hotel. After every decoded action the model can decide whether to continue or stop the decoding process. It is important to highlight that while we use the ontology to constrain the generation process, no part of the ontology is ever part of the model, but the model uses the ontology as one additional input. In that way it can be transferred to a new ontology in a purely zero-shot manner.

4 Experimental Setup

The objective of our experiments is four-fold. First, we want to show that when trained and tested on the same ontology, the user simulator can adequately capture the semantics represented in the real user data. At the same time, we also want to examine its zero-shot capability by conducting the evaluation on another unseen ontology. Second, as natural language output is an important component of the proposed model, we evaluate it separately using both automatic measures and a human preference test. Third, we jointly evaluate the GenTUS dia-

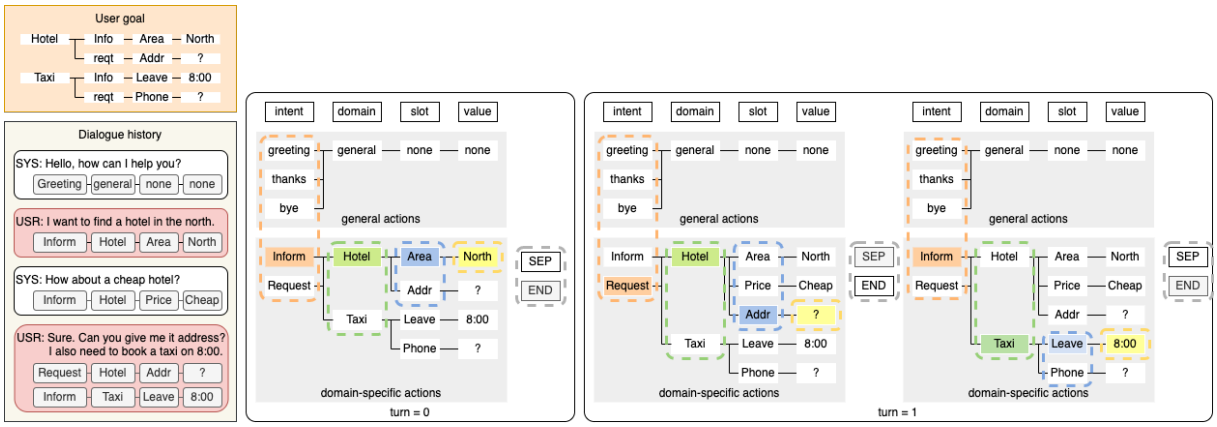


Figure 2: An example of a constrained semantic decoding space. The intents come from the ontology whereas domains, slots and values come from the user goal. In addition, system actions can insert new nodes. The user semantic actions can only contain nodes from the graph. More details are mentioned in section 3.3.

logue policy and its natural language output using a human trial and compare it to the state of the art. This aims to show the value of optimising the user simulator behaviour and language at the same time. Finally, we show how the behaviour of GenTUS can be further shaped by RL in interaction with a DS, with the aim of demonstrating that this model can yield a number of specialised user simulators.

4.1 Datasets

We conduct our experiments on two corpora, the Multi-Domain Wizard-of-Oz (MultiWOZ) (Budzianowski et al., 2018) and Schema-Guided Dialogue (SGD) (Lee et al., 2022) datasets. MultiWOZ is a human-to-human conversation dataset including around 10k dialogues, one person posing as a user and the other as an operator. In this dataset, more than one domain may be involved in one dialogue, even in the same turn. SGD consists of more than 20k dialogues between humans and a virtual assistant. The ontology of MultiWOZ includes 5 intents (3 general intents, e.g., *greeting* and *bye*, and 2 domain-specific intent, i.e., *inform* and *request*) and 7 different domains, e.g. *hotel* and *attraction*. On the other hand, the ontology of SGD includes 11 intents (2 general intents, i.e., *thank-you* and *goodbye*, and 9 domain-specific intents, e.g. *inform*, *request*, and *confirm*) and 20 different domains, e.g., *bank* and *music*. More details of these two datasets are listed in Appendix A.

4.2 Supervised Learning for GenTUS

Our model is inherited from Huggingface's transformers (Wolf et al., 2020) and trained on both MultiWOZ and SGD. To measure how well GenTUS can transfer to a new ontology, the model trained on MultiWOZ is not only tested on the MultiWOZ test set but also evaluated on the SGD test set without any further fine-tuning, and vice versa. To the best of our knowledge, no other data-driven US has been tested in such a rigorous zero-shot transfer set-up.

We evaluate NLG performance by automatic metrics, including slot error rate (SER), sacreBLEU score (Post, 2018) and self-BLEU score (Zhu et al., 2018), and a human preference test. SER evaluates the exact matching of semantic actions in the candidate utterance. $SER = (m + h)/N$, where N is the total number of slots in semantic actions, m and h stand for the number of missing and hallucinated slots, respectively. The self-BLEU is a diversity evaluation metric. For every data point we generate a sentence. Given such a sentence, we calculate a BLEU score where the reference sentences are all other generated sentences. Then we can get the self-BLEU score by averaging all these results. The lower self-BLEU score implies the higher diversity. We conduct the human preference test on the Amazon Mechanical Turk[2] platform. Following the setting of Peng et al. (2021), the workers are requested to rate each utterance from 1 (bad) to 3 (good) in terms of informativeness and naturalness. *Informativeness* measures whether the given utterance contains all the information specified in the semantic actions. *Naturalness* evaluates whether the given utterance is human-like. A screenshot of this questionnaire can be found in Appendix C.

In addition, we measure how well GenTUS can

[2] https://www.mturk.com/

fit or transfer to a dataset using precision, recall, F1 score, as well as turn accuracy on the semantic level and sacre-BLEU on the language level.

4.3 Training the Dialogue System with User Simulators

USs are designed to simulate the real-world scenario for training DSs, thus USs should respond in natural language as real users' utterances. In this section, we investigate the ability of the proposed model to train a dialogue policy by interacting on the natural language level.

The policies of different DSs are trained by proximal policy optimization (PPO) (Schulman et al., 2017), a simple and stable RL algorithm, with different USs, including the agenda-based US (ABUS) with template-based NLG (ABUS-T), ABUS with SC-GPT (ABUS-S), and GenTUS which generates language. Note that we do not include NLG modules in evaluation which are based on delexicalisation, such as Tseng et al. (2021), as their performance strongly depends on the amount of hand-coding invested in defining the delexicalisation rules. The downsides of delexicalisation already became clear in early neural network dialogue state trackers (Mrkšić et al., 2017) and are further exacerbated in natural language generation (Peng et al., 2020). We do however include a rule-based user simulator (Schatzmann et al., 2007) with a template-based NLG, noted as ABUS-T in our experiments, as the rule-based user simulator has achieved competitive results in human evaluations (Kreyssig et al., 2018; Lin et al., 2021). Also, TUS (Lin et al., 2021) did not significantly outperform ABUS in the human trial, so we exclude it from the evaluation here.

To deal with the user response in natural language, a natural language understanding module composed with BERT (Devlin et al., 2019) (BERTNLU) is included and a rule-based dialogue state tracker (RuleDST) is used to track the users' states for each DS. These modules, e.g., BERTNLU, RuleDST, ABUS, a template-based NLG, and SC-GPT, are provided in the ConvLab-2 framework (Zhu et al., 2020).

We train policies for 200 epochs, each of which consists of 1000 dialogues. The reward function gives a reward of 80 for a successful dialogue and -1 for each dialogue turn, with the maximum number of dialogue turns set to 40. For failed dialogues, an additional penalty is set to -40. Each dialogue policy is trained on 5 random seeds.

We apply the cross-model evaluation (Schatztnann et al., 2005) to evaluate these DSs. Different USs are used to evaluate a DS which is trained with a particular US to estimate the generalisation ability. We also conduct an interactive human trial. For evaluation, we select the DS policy performing best on the US it was trained on. For each DS we collected 300 dialogues. The human trial is implemented with DialCrowd (Lee et al., 2018; Huynh et al., 2022) connected to the Amazon Mechanical Turk platform. Users are provided with randomly generated user goals based on the ontology of MultiWOZ and are required to interact with DSs in natural language.

4.4 Fine-tuning GenTUS with RL

Simulators purely trained using supervised learning will learn behaviour that best fits the data and most likely will result in general behaviour. As behaviour can be very different from one user to another, it is important to be able to model different user behaviours, which will in turn result in more robust policies. To this end, we further fine-tune GenTUS using RL and shape its behaviour by deploying different reward functions. In order to achieve that, for a given user action $\{(intent_i, domain_i, slot_i, value_i)\}_{i=1}^{m}$, we define the turn level reward $r := -\rho_{\mathit{eff}} + \rho_{act} \cdot m$, where ρ_{eff} and ρ_{act} are hyperparameters. In addition, as for the system reward, we give a reward of 80 for a successful dialogue and -40 for a failed dialogue at the very end of the dialogue. We let GenTUS interact with the rule-based dialogue system both on semantic level and optimise its behaviour using PPO. We test two different reward settings that are distinguished by the turn level reward: $r_1 := -5 \cdot m$ (turn level penalty and low action reward) and $r_2 := -10 + 20 \cdot m$ (turn level penalty and high action reward). The corresponding average returns and trained user simulators associated with the rewards are abbreviated with R_1, R_2 and $User_1, User_2$ respectively. We train each model on 4 different seeds. We then take for every seed the model with highest average return on its respective reward and evaluate on the other reward functions to obtain a cross-reward evaluation.

5 Experimental Results

Our experimental results can be divided into five parts. In Section 5.1, we analysis the impact of

different features with an ablation study. In Section 5.2, we conduct the *direct* evaluation by measuring automatic metrics (SER, sacre-BLEU, and self-BLEU) and human ratings (informativeness and naturalness) from the preference test. In Section 5.3, we focus on the generalisability of GenTUS by a zero-shot ontology transfer experiment, measured by semantic level and language level metrics on two different corpora. The *indirect* evaluation is in Section 5.4. We compare DSs trained by different USs with cross-model evaluation. The result from the interactive human trial is discussed in Section 5.5. In Section 5.6, we show that it is possible to further configure the behaviour of GenTUS via RL.

5.1 Impact of different features

We conduct an ablation study to investigate the usefulness of our proposed feature representation. The result is shown in Table 1. First, we measure the performance of the model which takes the turn information, the system action $action^{t}_{sys}$ and the user action $action^{t-1}_{usr}$ from previous turn. Without context information, the model can only achieve 0.21 turn accuracy and 0.35 F1-score. After including the user goal $goal$, the F1-score is improved by 0.30 and the turn accuracy is also improved by 0.30 absolutely. After adding more user history $action^{t-1:t-3}_{usr}$, the F1 score is also improved slightly with the same turn accuracy.

This result indicates that the context information can improve the performance especially including the user goal in the input sequence.

Model	P	R	F1	ACC
System and user action only	0.42	0.30	0.35	0.21
+ user goal	0.66	0.64	0.65	0.51
+ history	0.68	0.66	0.66	0.51

Table 1: The GenTUS ablation experiments on MultiWOZ. We analyse the impact of different input features by measuring precision (P), recall (R), F1 score (F1), and turn accuracy (ACC).

5.2 Natural Language Evaluation

The NLG performance of different models on MultiWOZ is shown in Table 2. TemplateNLG, SC-GPT, and GenTUS-golden generate natural language responses from golden semantic actions and their SER is calculated based on these golden semantic actions. On the other hand, the language of GenTUS is generated based on semantic actions predicted by itself, which means we can directly measure the agreement between the semantic action the simulator indented to produce and the final natural language content produced by the simulated user. The sacre-BLEU is calculated with golden utterances.

Model	SER ↓	sacre-BLEU ↑	self-BLEU ↓
Human	3.92%	-	0.77
TemplateNLG	1.67%	10.46	0.89
SC-GPT	5.33%	10.51	**0.79**
GenTUS-golden	5.73%	**19.61**	0.93
GenTUS	**3.97%**	-	0.95

Table 2: The NLG performance on MultiWOZ. GenTUS-golden is generated based on the golden semantic actions and GenTUS is using its own semantic action prediction. The arrow direction means which trend is better.

Model	Informativeness	Naturalness
SC-GPT	2.50	2.45
GenTUS	2.55	**2.58**

Table 3: Human preference test for NLG on MultiWOZ. The naturalness score is statistically significantly different ($p_v < 0.05$).

Although data-driven NLG models have higher SER than template-based NLG, these models have better scores in BLEU. GenTUS-golden outperforms SC-GPT by 9.10 points in BLEU because our model not only takes semantic actions as input but also context information, e.g., the user goal. Moreover, there is no statistically significant difference in SER between SC-GPT and GenTUS-golden. The human preference test in Table 3 also shows that GenTUS is more natural than SC-GPT with similar informativeness. The diversity of the proposed model is the worst, which is not surprising as we didn't include beam-search or sampling to keep the computational complexity as low as possible. An investigation of a method which balances the two we leave for future work.

Without golden dialogue acts in the input, the SER of GenTUS drops by 1.77% absolute when GenTUS generates utterances from its prediction dialogue acts instead of from golden dialogue acts, which means the language-level and semantic-level outputs of GenTUS are in agreement. In other words, with the context information and its predicted semantic actions, GenTUS can generate more natural language and have fewer missing and redundant pieces of information.

5.3 Zero-shot Ontology Transfer

The results of zero-shot ontology transfer are shown in Table 4. For the semantic level evaluation, GenTUS has higher precision, recall, F1 score and turn accuracy on MultiWOZ than SGD when training and testing on the same corpus. The reason is the ontology of SGD is more complicated than MultiWOZ, i.e., contains more intents, domains, slots and values as shown in Section 4.1.

The performance of GenTUS trained on MultiWOZ dropped by 0.39 on F1 score and 0.35 on turn accuracy when testing on SGD. On the other hand, GenTUS trained on SGD can still achieve 0.49 on F1 score and 0.34 turn accuracy when testing on MultiWOZ without fine-tuning on the unseen MultiWOZ ontology. In other words, GenTUS trained on SGD can get a comparable F1 score and turn accuracy on both known and unknown ontology.

When testing and training on the same corpus, the BLEU score of GenTUS is 17.84 on MultiWOZ and 18.30 on SGD. However, when transferring to another corpus, the BLEU score drops because users in MultiWOZ and SGD have different vocabulary and language styles.

Train data	Test data	Semantic P	R	F1	ACC	Language sacreBLEU
M	M	0.68	0.66	0.66	0.51	17.84
S	S	0.60	0.58	0.58	0.47	18.30
S	M	0.51	0.51	0.49	0.34	2.70
M	S	0.30	0.26	0.27	0.16	1.86

Table 4: The cross-dataset evaluation of GenTUS based on two different corpora, MultiWOZ 2.1 (M) and Schema-Guided Dialogue dataset (S). The semantic actions and language responses generated by GenTUS are evaluated by semantic level metrics, i.e., precision (P), recall (R), F1 score (F1) and turn accuracy (ACC), and language level metric, i.e., sacre-BLEU.

5.4 Cross-model Evaluation

The results of cross-model evaluation are presented in Table 5. The DS trained with GenTUS has the best performance when interacting with ABUS-T in a 15% absolute improvement in success rate over its performance on GenTUS. On the other hand, although the DS trained with ABUS-T achieves 78% success rate, its performance drops by 28% absolute when evaluated by GenTUS. The DS trained with ABUS-S also performs best when interacting with ABUS-T, with 17% absolute improvement in success rate interacting with ABUS-S. All three DSs achieve their best performance when evaluated by ABUS-T, which means ABUS is the easiest setting. This indicates that it may not be sufficient to simulate real world scenario with only a handcrafted policy and a template-based NLG.

On the other hand, the USs with data-driven NLG are more difficult for the DS to handle. The DS trained by ABUS-T performs better than the DS trained by ABUS-S because they learn from the same policy and SC-GPT has higher SER, making the DS hard to be fully optimised.

US for training	US for testing ABUS-T	ABUS-S	GenTUS
ABUS-T	**0.78**	**0.63**	0.50
ABUS-S	0.74	0.57	0.45
GenTUS	0.68	0.43	**0.53**

Table 5: The success rates of policies trained on GenTUS, ABUS with template NLG (ABUS-T), and ABUS with SC-GPT (ABUS-S) when tested on various USs. Each pair is evaluated by 400 dialogues on 5 seeds, which is 2K dialogues in total.

5.5 Interactive Human Trial

US for training	Success	Overall
ABUS-T	0.75	3.71
ABUS-S	0.79	3.83
GenTUS	**0.86**	**4.08**

Table 6: The interactive human trial results include success rate and overall rating as judged by users. Each system is evaluated by 300 dialogues. The success rate and overall score of GenTUS are statistically significantly different from ABUS-S and ABUS-T ($p_v < 0.05$)

The result of the interactive human trial is shown in Table 6. 155 users were involved in this trial. The number of interactions per user varies from 1 to 48. A dialogue is rated as successful if the system fulfils the user's given goal. The overall rating ranges from 1 (very poor) to 5 (excellent).

The DS trained by GenTUS outperforms the DS trained by ABUS-T and the DS trained by ABUS-S both on success rate and overall rating, which shows that is beneficial to train a DS with a jointly optimised user policy and NLG. However, we cannot observe statistically significant differences between ABUS-T and ABUS-S on success and overall rating, which means including a data-driven NLG module with the rule-based US is not sufficient to train an optimal DS.

Models	Success	Avg Acts	Turns	R_1	R_2
User 1	0.84 ± 0.03	1.33 ± 0.03	7.01 ± 0.27	33.5 ± 3.5	34.2 ± 3.7
User 2	0.78 ± 0.04	1.81 ± 0.04	7.24 ± 0.33	4.3 ± 6.2	119.1 ± 15.5
Supervised	0.76 ± 0.08	1.39 ± 0.04	7.38 ± 0.32	30.9 ± 8.2	38.6 ± 10.0

Table 7: Results after fine-tuning GenTUS using RL on three different reward functions. Results show mean and 95% confidence intervals.

5.6 Fine-tuning GenTUS with RL

The results of RL training are depicted in Table 7. We can observe that both users obtain the highest return on the respective reward function. The success rate of both user 1 and user 2 are higher than supervised model because of the success reward signal in RL. User 1, which tries to lower its number of actions, has a similar average number of actions compared to supervised model, suggesting that paid users from the corpus do not want to say more than is necessary to achieve a successful dialogue. User 2, which is rewarded for taking many actions in a turn, shows a much higher average number of actions compared to the other users, reflecting a different user behaviour – a chatty user.

6 Conclusion

We propose a generative transformer-based user simulator (GenTUS), which achieves high interpretability and linguistic variation by generating both semantic actions and natural language utterances. Moreover, it produces generalisable feature representation by treating the inputs and outputs as word sequences and leveraging a large pre-trained language model. Our results show that GenTUS generates more natural language than SC-GPT in a given context and it can transfer to an unseen ontology in a zero-shot fashion. We consolidate our findings by a number of automatic as well as human evaluations. In addition, the GenTUS behaviour can be further configured by RL with different reward functions, providing an opportunity to build specialised USs. In future work, we hope to modify also the NLG of GenTUS via RL in order to model user sentiment or personality.

Acknowledgements

This work is a part of DYMO project which has received funding from the European Research Council (ERC) provided under the Horizon 2020 research and innovation programme (Grant agreement No. STG2018 804636). N. Lubis, C. van Niekerk, M. Heck and S. Feng are funded by an Alexander von Humboldt Sofja Kovalevskaja Award endowed by the German Federal Ministry of Education and Research. Computing resources were provided by Google Cloud and HHU ZIM.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Heriberto Cuayáhuitl, Steve Renals, Oliver Lemon, and Hiroshi Shimodaira. 2005. Human-computer dialogue simulation using hidden markov models. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 290–295. IEEE.

Lucie Daubigney, Matthieu Geist, Senthilkumar Chandramohan, and Olivier Pietquin. 2012. A comprehensive reinforcement learning framework for dialogue management optimization. *IEEE Journal of Selected Topics in Signal Processing*, 6(8):891–902.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87. IEEE.

Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. *Interspeech 2016*, pages 1151–1155.

Aciel Eshky, Ben Allison, and Mark Steedman. 2012. Generative goal-driven user simulation for dialog management. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language*

Learning, pages 71–81, Jeju Island, Korea. Association for Computational Linguistics.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Ninth International Conference on Spoken Language Processing*.

Izzeddin Gür, Dilek Hakkani-Tür, Gokhan Tür, and Pararth Shah. 2018. User modeling for task oriented dialogues. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 900–906. IEEE.

Jessica Huynh, Ting-Rui Chiang, Jeffrey Bigham, and Maxine Eskenazi. 2022. Dialcrowd 2.0: A quality-focused dialog system crowdsourcing toolkit. In *Proceedings of the Language Resources and Evaluation Conference*, pages 1256–1263, Marseille, France. European Language Resources Association.

Florian Kreyssig, Iñigo Casanueva, Paweł Budzianowski, and Milica Gašić. 2018. Neural user simulation for corpus-based policy optimisation of spoken dialogue systems. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 60–69, Melbourne, Australia. Association for Computational Linguistics.

Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2022. Sgd-x: A benchmark for robust generalization in schema-guided dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Kyusong Lee, Tiancheng Zhao, Alan W Black, and Maxine Eskenazi. 2018. Dialcrowd: A toolkit for easy dialog system assessment. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 245–248.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gasic. 2021. Domain-independent user simulation with transformers for task-oriented dialogue systems. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 445–456, Singapore and Online. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 172–182.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Jost Schatzmann, Kallirroi Georgila, and Steve Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 45–54.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152, Rochester, New York. Association for Computational Linguistics.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowledge Engineering Review*, 21(2):97–126.

Jost Schatztnann, Matthew N Stuttle, Karl Weilhammer, and Steve Young. 2005. Effects of the user model on simulation-based learning of dialogue strategies. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 220–225. IEEE.

Konrad Scheffler and Steve Young. 2002. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proceedings of the second international conference on Human Language Technology Research*, pages 12–19. Citeseer.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne. 2021. Transferable dialogue systems and user simulators. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 152–166, Online. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

A Intents and domains in MultiWOZ and SGD

type	system	user
general	welcome, reqmore, bye, thank, greet	bye, thank, greet
domain-specific	recommend, inform, request, select, book, nobook, offerbook, offerbooked, nooffer	inform, request

Table 8: Al intents in the MultiWOZ dataset.

All intents in the MultiWOZ dataset are listed in Table 8 and all intents in SGD dataset are listed in Table 9. The domains in SGD follow the form of `<domain_name>_<number>` and the number is used to disambiguate services from the same domain (Lee et al., 2022). We normalize them to domain name only. All domains in MultiWOZ and SGD and listed in Table 10.

type	system	user
general	goodbye, req_more	thank_you, goodbye
domain-specific	inform, notify_success, request, notify_failure, confirm, offer_intent, offer, inform_count	inform_intent, inform negate_intent, negate affirm_intent, affirm, request_alts, request, select

Table 9: All intents in the SGD dataset.

dataset	domains
MultiWOZ	attraction, hospital, hotel, police, restaurant, taxi, train
SGD	alarm, banks, bus, calendar, events, flights, homes, hotels, media, messaging, movies, music, payment, rental_cars, restaurants, ride_sharing, servies, train, travel, weather

Table 10: All domains in MultiWOZ and SGD. The names of domains in SGD are normalized.

B A generated dialogue example

Here we show a dialogue example of GenTUS interacting with a rule-based DS in Table 11. The system's semantic actions are converted to natural language by a template-based NLG.

We can notice that GenTUS provides a natural response in turn 5. This utterance is difficult for the natural language understanding model and the system is not capable to handle it. As a result, GenTUS mentions the departure and destination explicitly in turn 8.

C An example of human preference test

The dialogue history, semantic actions, and utterances generated by two different systems are presented to workers on AMT platform. They are required to rate the informativeness and naturalness of these utterances respectively.

User goal	
attraction: {info: {type: college}, reqt: {postcode: ?, entrance fee: ?}}, hotel: {info: {area: north, star:0}, reqt: {parking:?}, taxi: {arriveby:13:00}, reqt:{phone:?, car type:?}}	
Dialogue	
0-sys	Welcome to the dialogue system. How can I help you?
0-usr	I'm looking for a college to visit in town.
1-sys	There 's 18 college in the centre , anything in mind ? What area would you like this in ?
1-usr	No, but I would like the postcode and entrance fee.
2-sys	We have christ's college. Its postcode is cb23bu . Would you be interested in visiting a college? The entry fee is free. The postcode is cb23bu. Did you need any further assistance today?
2-usr	Yes, I'm also looking for a place to stay in the north.
3-sys	I have 13 moderate guesthouse that fit your criteria. Want one of those? I 'd be happy to help. May I ask what price range and area of town you are looking for?
3-usr	I would like a 0 star rating.
4-sys	I have 1 option for you. It's listed as cheap. It is a guesthouse. Would city centre north b and b work for you?
4-usr	Does it have free parking?
5-sys	City centre north b and b is available would you like to try that ? There is wifi available at the hotel . The parking is free . The hotel is 0 stars . It is a guesthouse .
5-usr	**Great. I also need a taxi to commute between the two places.**
6-sys	Okay, I can help with that. What are your departure site and destination? Also, when would you like to leave or arrive by?
6-usr	I need to arrive by 13:00.
7-sys	There are 18. Would you like me to recommend one for you?
7-usr	I 'd like the car type and contact number please.
8-sys	There are 18 entries found for "college" in the centre of town. Would you like the first listing? How about the christ's college? Glad I could be of assistance! If you need anything else, let me know.
8-usr	**Yes, I need a taxi to get me from the college to the hotel by 13:00.**
...	

Table 11: A dialogue example of GenTUS interacting with rule-based dialogue system. GenTUS responds in a natural way utilising anaphora in turn 5. After it finds out the system is not capable to understand, GenTUS provides the departure and destination explicitly in turn 8.

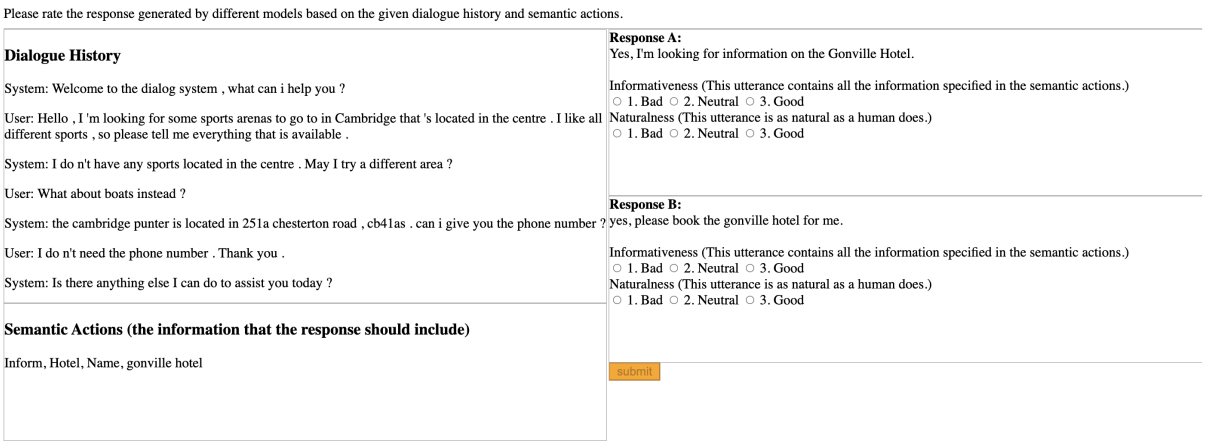


Figure 3: An example of human preference test.

AARGH! End-to-end Retrieval-Generation for Task-Oriented Dialog

Tomáš Nekvinda and **Ondřej Dušek**
Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Prague, Czech Republic
{nekvinda,odusek}@ufal.mff.cuni.cz

Abstract

We introduce AARGH, an end-to-end task-oriented dialog system combining retrieval and generative approaches in a single model, aiming at improving dialog management and lexical diversity of outputs. The model features a new response selection method based on an action-aware training objective and a simplified single-encoder retrieval architecture which allow us to build an end-to-end retrieval-enhanced generation model where retrieval and generation share most of the parameters.

On the MultiWOZ dataset, we show that our approach produces more diverse outputs while maintaining or improving state tracking and context-to-response generation performance, compared to state-of-the-art baselines.

1 Introduction

Most research task-oriented dialog models nowadays focus on end-to-end modeling, i.e., the whole dialog system is integrated into a single neural network (Wen et al., 2017; Ham et al., 2020). Although recent end-to-end *generative* approaches based on pre-trained language models produce fluent and natural responses, they suffer from two major problems: (1) hallucinations and lack of grounding (Dziri et al., 2021), which result in faulty dialog management or responses inconsistent with the dialog state or database results, and (2) blandness and low lexical diversity of outputs (Zhang et al., 2020b). On the other hand, *retrieval-based* dialog systems (Chaudhuri et al., 2018) select the most appropriate response candidate from a human-generated training set, thus producing varied outputs. However, their responses might not fit the context and can lead to disfluent conversations, especially when the set of candidates is sparse. This limits their usage to very large datasets which do not support dialog state tracking or database access (Lowe et al., 2015; Al-Rfou et al., 2016).

Several recent works focus on combining the retrieval and generative dialog systems via response selection and subsequent refinement, i.e., *retrieval-augmented generation* (Pandey et al., 2018; Weston et al., 2018; Cai et al., 2019b; Thulke et al., 2021). These models are used for open-domain conversations or to incorporate external knowledge into task-oriented systems and do not consider an explicit dialog state.

Our work follows the retrieve-and-refine approach, but we adapt it for database-aware task-oriented dialog. We aim at improving diversity of produced responses while preserving their appropriateness. In other words, we do not retrieve any new information from an external knowledge base, instead, we retrieve relevant training data responses to support the decoder in producing varied outputs. To the best of our knowledge, we are the first to use retrieval-augmented models in this context. Unlike previous works, we merge the retrieval and generative components into a single neural network and train both tasks jointly, instead of using two separately trained models. Our contributions are summarized as follows:[1]

- We propose a single-encoder retrieval model utilizing dialog action annotation during training, and we show its superior retrieval capabilities in the task-oriented setting compared to two-encoder baseline models (Humeau et al., 2020).
- We propose an end-to-end *task-oriented* generative system with an *integrated* minimalistic retrieval module. We compare it to strong baselines that model response selection and generation separately.
- On the MultiWOZ benchmark (Budzianowski et al., 2018), our approaches outperform previous methods in terms of lexical diversity and achieve competitive or better results in automatic metrics and human evaluation.

[1]Code: https://github.com/Tomiinek/Aargh

Proceedings of the SIGdial 2022 Conference, pages 283–297
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

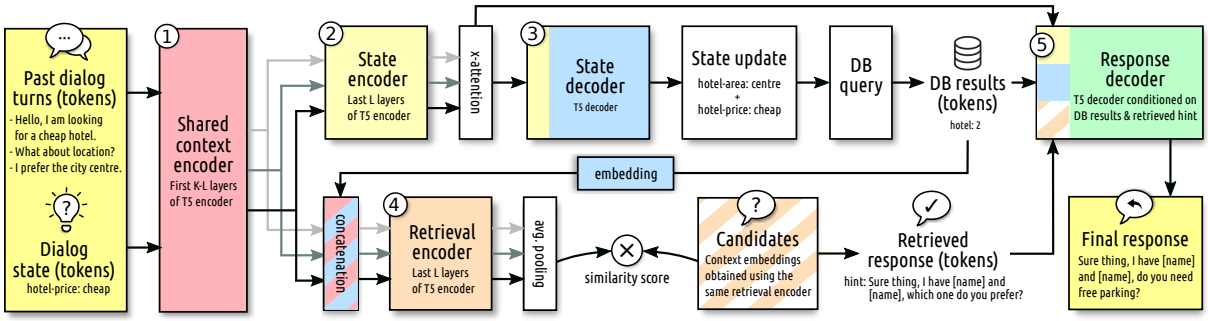


Figure 1: Our retrieval-based generative task-oriented system (AARGH, see Section 3.5). Numbers in module boxes mark the order of processing during inference: (1) inputs are pushed through the shared context encoder and (2) state encoder; (3) the state decoder produces the update to the current dialog state. The new state is used to query the database whose outputs are discretized, embedded, and (4) used in the retrieval encoder whose output is reduced to a single vector via average pooling. The context embedding is used to get the best response candidate (hint). Finally, (5) the response decoder, which can attend to the state encoder outputs via cross-attention and is conditioned on the database results and the hint, generates the final system response to be shown to the user.

2 Related Work

Task-Oriented Response Generation Most current works focus on building multi-domain database-grounded systems. The breeding ground for this research is the large-scale conversational dataset MultiWOZ (Budzianowski et al., 2018; Eric et al., 2020; Zang et al., 2020).

Recent models often benefit from *action annotation*. Zhang et al. (2020a) use action-based data augmentation and a three-stage architecture, decoding the dialog state, action, and response. Chen et al. (2019) generate responses without state tracking, exploiting a hierarchical structure of the action annotation. On the other hand, reinforcement learning models (Wang et al., 2021) learn latent actions from data without using annotation.

Recent works focus on *end-to-end* systems based on pre-trained language models. Budzianowski and Vulic (2019) fine-tune GPT-2 (Radford et al., 2019) to model task-oriented dialogs, Hosseini-Asl et al. (2020) enhance this approach with explicitly decoded system actions. Peng et al. (2021b) use auxiliary training objectives and machine teaching for GPT-2 fine-tuning. Lin et al. (2020) introduced the encoder-decoder-based framework MinTL with BART (Devlin et al., 2019a) or T5 (Kale and Rastogi, 2020) backbones (see Section 3.1).

Response Selection can be viewed as scoring response candidates given a dialog context. A popular approach is the dual encoder architecture (Lowe et al., 2015; Henderson et al., 2019b) where the response and context encoders model a joint embedding space. The encoders can take various forms: Henderson et al. (2019a) compare encoders based on BERT (Devlin et al., 2019b) and custom encoders pre-trained on Reddit; Wu et al. (2020) pre-train encoders specifically for task-oriented conversations. Humeau et al. (2020) introduce poly-encoders, which produce multiple context encodings and add an attention layer to allow rich interaction with the candidate encoding (cf. Section 3.3).

Retrieval-Augmented Generation To benefit from both retrieval and generative models, Weston et al. (2018) proposed an open-domain dialog system utilizing a retrieval network and a decoder to refine retrieved responses. Roller et al. (2021) further developed this approach, using poly-encoders with a large pre-trained decoder. They found that their decoder tends to ignore the retrieved response hints. To combat this, they propose the α-blending method (replacing retrieval output with ground truth, see Section 3.2). Similarly, Gupta et al. (2021) and Cai et al. (2019a,b) focus on retrieval-augmented open-domain dialog, but to prevent the inflow of erroneous information into the generative part of their models, they use semantic frames or reduced forms of retrieved responses instead of raw response texts.

Thulke et al. (2021) aim at knowledge retrieval from external documents for resolution of out-of-domain questions on MultiWOZ (Kim et al., 2020). Shalyminov et al. (2020) present the only work using generation and retrieval in a single model. They finetune GPT-2 (Radford et al., 2019) for response generation in a low-resource task-oriented setup, retrieve alternative responses based on the model's embedding similarity, and choose between gener-

ated and retrieved responses on-the-fly. However, their model is not trained for retrieval, cannot alter retrieved responses, and does not take a dialog state or database into account.

3 Method

We aim at end-to-end modeling of database-aware task-oriented systems, i.e., systems supporting both dialog state tracking and response generation tasks (Young et al., 2013). We combine retrieval and generative models to reduce hallucinations and boost output diversity. We first describe our purely generative baseline (Section 3.1), then explain baseline generation based on retrieved hints (Section 3.2). We then introduce baseline retrieval models (Section 3.3) and our action-aware retrieval (Section 3.4). Finally, we describe AARGH, our single-model retrieval generation hybrid, in Section 3.5. AARGH is shown in Figure 1; other setups are depicted in Appendix A.

3.1 Generative Baseline

Our purely-generative baseline model (**Gen**) follows MinTL (Lin et al., 2020). It is based on an encoder-decoder backbone with a context encoder, shared among two decoders: one for modeling the dialog state updates, the other for producing the final system response. Both decoders attend to the encoded input tokens via an attention mechanism.

The encoder input sequence consists of a concatenation of two parts: (1) past dialog utterances prepended with <|*system*|> or <|*user*|> tokens, and (2) the initial dialog state converted to a string, e.g., *hotel [area: center] restaurant [food: African, pricerange: expensive]*. The first decoder is conditioned only on the start-of-sequence token and predicts the dialog state update as a difference between the current state and the initial state. The second decoder is conditioned on the number of database results for each queried domain, e.g. *train: 6* if there are six matching results for a train search, and generates the final response.

During inference, the input is passed through the encoder, then the state update is predicted, merged with the initial dialog state, and this new state is used to query the database (see Section 4 for details). The final system response is predicted based on the context, state, and database results.

3.2 Retrieval-Augmented Response Generation

To combine the retrieval and generative approaches, we follow Weston et al. (2018) and incorporate response *hints*, i.e., the outputs of a retrieval module (Sections 3.3, 3.4), into the generative module in their original form as raw sub-word tokens. Specifically, we add the retrieved response prepended with <|*hint*|> to the input of Gen's response decoder (Section 3.1), alongside the database results.

Gupta et al. (2021) state that this straightforward token-based retrieve & refine setup might lead to generating incoherent responses due to over-copying of contextually irrelevant tokens. However, using more abstract outputs of the retrieval module, e.g. semantic frames or salient words would go against our goal of reducing blandness and increasing responses lexical diversity. To smoothly control the amount of token copying, we follow Roller et al. (2021) and use the so-called α-blending. During training, we replace the retrieved utterance with the ground-truth final response with probability α. This method also ensures that the decoder learns to attend to the retrieval part of its input successfully.

3.3 Baseline Response Selection

We consider two baseline retrieval model variants:

Dual-encoder (DE) follows the very popular retrieval architecture (Lowe et al., 2015; Humeau et al., 2020) which makes use of context and response encoders. Both produce a single vector in a joint embedding space. During training, the context embedding and the corresponding response embedding are pushed towards each other, while other responses in the training batch are used as negative examples, i.e., cross-entropy loss is used:

$$\mathcal{L}(S) = \frac{1}{N} \sum_j \left(-S_{j,j} + \log \sum_i e^{S_{j,i}} \right)$$

where $S \in \mathbb{R}^{N \times N}$ is the similarity matrix between *normalized* encoded responses $\mathbf{e}^r$ and contexts $\mathbf{e}^c$ in the batch, specifically $S_{i,j} = w \cdot (\mathbf{e}^c_i \cdot \mathbf{e}^r_j)$, where $w > 0$ is a trainable scaling factor.

Inference-time retrieval is as simple as finding the nearest candidate embedding given a context embedding. The context input is similar to Gen's (see Section 3.1): a concatenation of the current updated dialog state, the number of matching database results and past user and system utterances. En-

coders are followed by average pooling and a fully-connected layer for dimensionality reduction.

Poly-encoder (PE) an extension of DE, aiming at richer interaction between the candidate and the context. The candidate encoder is unchanged. In the context encoder, the average pooling is replaced with two levels of dot-product attention (Vaswani et al., 2017; Humeau et al., 2020). The first level summarizes the encoded context tokens into m vectors. The context tokens act as attention keys and values; queries to this attention are m learned embeddings (query codes). The second attention level provides the candidate-context interaction: it takes the m context summary vectors as keys and values, and the candidate encoder output acts as the query. The parameter m provides trade-off between inference complexity and richness of the context encoding. The loss term remains the same.

3.4 Action-aware Response Selection

We argue that the dual- or poly-encoder models are not practical for the task-oriented settings as their performance depends on the way negative examples are sampled during training (Nugmanova et al., 2019). Choosing appropriate negative examples is difficult in task-oriented datasets as system responses are often very similar to each other (with the conversations being in a narrow domain and following similar patterns). Therefore, we propose a method for candidate selection based on system action annotation, which is usually available in task-oriented datasets. We designed the method to be usable with a single encoder only, but we also include a dual-encoder version for comparison.

Action-aware-encoder (AAE) Using two separate encoders to encode the response and the context might be impractical due to large model size. Some recent works (e.g., Wu et al. (2020); Roller et al. (2021)) use a single shared encoder instead, and Henderson et al. (2020) discuss parameter sharing between the two encoders. In view of that, we propose a single-encoder action-aware retrieval model. We train it to produce embeddings of dialog contexts which are close to each other if the corresponding responses in the training data have similar action annotation. More precisely, we adapt Wan et al. (2018)'s generalized end-to-end loss, originally developed for batch-wise training of speaker classification from audio: To form training mini-batches, we first sample M random dialog actions, and for each of those actions, we sample N examples that include the particular action in their system action annotation. We then encode dialog contexts corresponding to the sampled examples into normalized embeddings $\mathbf{e}_{m,n}$, and compute the similarity matrix as follows:

$$S_{ji,k} = \begin{cases} w \cdot (\mathbf{e}_{j,i} \cdot \mathbf{c}_j^{\{i\}}) & \text{if k = j} \\ w \cdot (\mathbf{e}_{j,i} \cdot \mathbf{c}_k^{\varnothing}) & \text{otherwise} \end{cases}$$

$$\mathbf{c}_j^A = \frac{1}{N - |A|} \sum_{i \in [N]-A} \mathbf{e}_{j,i}$$

where $S \in \mathbb{R}^{N \cdot M \times M}$, $i \in [N]$, $j, k \in [M]$, and $A \subseteq [N]$ is a set of indices. Same as for DE, $w > 0$ is a trainable scaling factor of the similarity matrix. In other words, the similarity matrix describes the similarity between embeddings of each example and centroids, i.e., the means of N embeddings that correspond to the same particular action. For stability reasons and to avoid trivial solutions, we follow Wan et al. (2018) and exclude $\mathbf{e}_{j,i}$ from the centroid calculation when computing $S_{ji,j}$.

We then maximize the similarity between the examples and their corresponding centroids while using other centroids as negative examples:

$$\mathcal{L}(S) = \frac{1}{N \cdot M} \sum_{j,i} \left(-S_{ji,j} + \log \sum_k e^{S_{ji,k}} \right)$$

During inference, we rank the responses from the training set according to the cosine similarity of their corresponding contexts and the query context. Again, the contexts consist of the current updated dialog state, the number of matching database results and past utterances.

Action-aware-dual-encoder (AADE) This setup follows the DE architecture (see Section 3.3), but it is trained in a similar way as AAE, i.e., we form training mini-batches identically and for each of M distinct actions in the batch, we treat all N examples as positive examples.

3.5 Hybrid End-to-end Model

To further simplify the retrieval-augmented setup, reduce the number of trainable parameters and gain back computational efficiency, we introduce an end-to-end Action-Aware Retrieval-Generative Hybrid model (**AARGH**), which jointly models both response selection and context-to-response generation (see Figure 1). It is a natural extension of the

Gen generative model (Section 3.1), enabled by our new single-encoder action-aware response retrieval (AAE, Section 3.4).

A new retrieval encoder, which produces normalized context embeddings, shares most parameters with the original encoder, which is followed by the two decoders and is partially responsible for state tracking and response generation. To build the retrieval encoder, we fork the last L layers of the original encoder and condition them on the outputs of the shared preceding layers, concatenated with an embedding of the number of current database results. To obtain this embedding, we convert the number of database results into a small set of bins, which are then embedded via a learnt embedding layer of size E.[2] The new retrieval encoder is followed by average pooling and trained using the same objective as AAE (see Section 3.4).

During inference, we pass the input through the partially shared context encoder and decode and update the dialog state. The new state is used to query the database. Database results are embedded and added to the output of the last encoder shared layer to form the input to the retrieval encoder, which produces the context embedding and a retrieved response. Based on state, database results, and retrieved response, the response decoder produces the final (delexicalized) response.

4 Experimental Setup

Models Our models are based on pre-trained models from HuggingFace (Wolf et al., 2020): We implement Gen and the generative parts in our retrieval-based models using T5-base (Kale and Rastogi, 2020). Retrieval encoders in DE, AADE, PE and AAE are implemented as fine-tuned BERT-base (Devlin et al., 2019a). AARGH is built upon T5-base, same as Gen; we fork the last $L = 2$ out of $K = 12$ encoder layers. The choice of L is a trade-off between model performance and size.[3] The database embedding has size $E = 4$. For simplicity, we do not use specialized backbones pre-trained on dialogs such as ToD-BERT (Wu et al., 2020). PE uses $m = 16$ query codes (see Section 3.3) and single-headed attention mechanisms.

[2]This conversion is dataset-specific and not used in other compared models such as Gen. We use the label 0 if there are no results, 1 for 0 matching results, 2, 3, 4 if there are 1, 2 or 3 results, respectively, 5, 6 if there are less than 6 or 11 results, and 7 if there are 11 or more results.

[3]We noticed a performance drop when using $L = 1$, and $L = 3$ did not bring any large gain.

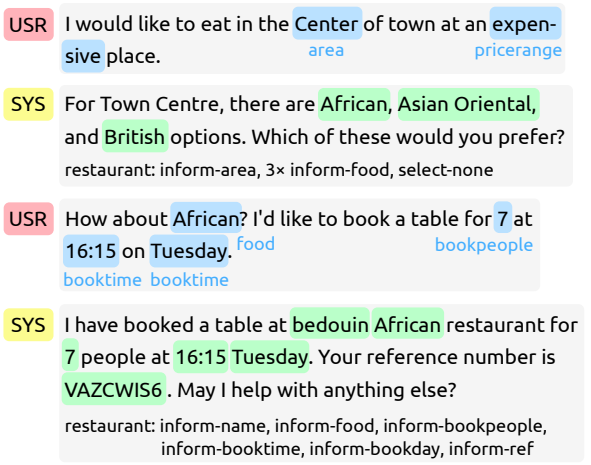


Figure 2: Part of a short conversation from MultiWOZ. It has user and system turns, and annotated slot spans. Both, user and system affect the dialog state. Actions are shown below system texts.

Data and database We experiment on the MultiWOZ 2.2 dataset (Budzianowski et al., 2018; Zang et al., 2020) which is a popular dataset with around 10k task-oriented conversations in 7 different domains such as trains, restaurants, or hotels (see Figure 2). A single conversation can touch multiple domains. The dataset has an associated database, dialog state annotation, dialog action annotation of system turns, and slot value span annotation for easy delexicalization (Wen et al., 2015), thus enabling development of realistic end-to-end dialog systems.[4] To query the database using the belief state, we use the fuzzy matching implementation by Nekvinda and Dušek (2021). To filter out inactive domains from database results during inference, we follow previous work and estimate the currently active domain from dialog state updates.

Input and output format We use the same formats for all models. Target responses are delexicalized using MultiWOZ 2.2 span annotation, and we limit the context to 5 utterances. MultiWOZ action labels include domain, action, and slot name, e.g., *train-inform-price*. We remove domains from the labels to limit data sparsity.

Training procedure DE, AADE, PE and AAE are trained in two stages. The retrieval part is trained first and provides response hints to the generative model during the second phase. Modules in AARGH are trained jointly, but we alternate param-

[4]Unlike the similar-sized Taskmaster (Byrne et al., 2019) and SGD (Rastogi et al., 2020) datasets, which lack databases and annotation detail.

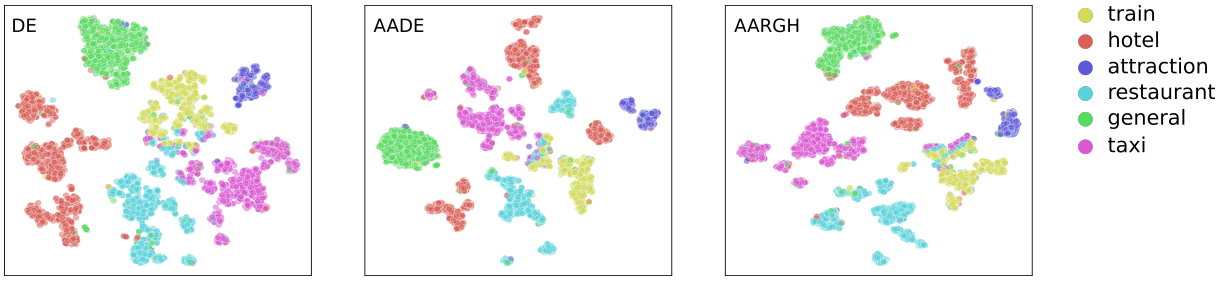


Figure 3: t-SNE projection of test set context embeddings (colored by domains) of retrieval modules of our models. The colors indicate the different MultiWOZ domains that are associated with the corresponding dialog turns.

eter updates for the retrieval encoder and the rest of the network. To do so, we use two separate optimizers. AARGH's hints used in the response decoder during training are refreshed after every epoch. All models are optimized using Adam (Kingma and Ba, 2015) and cosine learning rate decay with warmup. With respect to memory limits of our hardware, we set $N = 6$, $M = 8$ for batch sampling during training of retrieval parts of AAE and AARGH.

α-blending We experiment with two α-blending values: a conservative one ($\alpha = 0.05$, marked "$\downarrow$") and a greedy one ($\alpha = 0.4$, marked "$\uparrow$"), targeting a mostly generation-focused and a mostly retrieval-focused setting.[5]

Decoding We use greedy decoding for dialog state update generation. For response generation, we report results with greedy decoding in Section 5 and with beam search in Appendix B.

5 Evaluation and Results

We focus on end-to-end modeling, which includes dialog state tracking and response generation. All reported results are on MultiWOZ test set with 1000 dialogs, averaged over 8 different random seeds. We generated responses given ground truth contexts. We follow MinTL and predict the dialog state cumulatively for each conversation turn, which means that state tracking errors may compound. See Appendix C for an example end-to-end conversation without any ground-truth information.

5.1 Response selection

First, we assess the performance of retrieval components of DE, AADE, PE, AAE and AARGH. We cannot use the popular R@k metric (Chaudhuri et al., 2018) as AAE and AARGH use embeddings of dialog contexts (not responses) of candidates as the search criterion and would always score 100%. Instead, we use the action annotation and measure the intersection over union (IoU), full-match and no-match rates on sets of actions associated with top-1 retrieved and ground-truth responses. We add BLEU (Papineni et al., 2002; Liu et al., 2016a) between ground-truth and retrieved responses and the proportion of distinct retrieval outputs to assess their lexical similarity to references and diversity.

Setting	BLEU	Action IoU	% full match	% no match	% uniq. hints
Random	2.1	5.1 ± 0.2	1.1	85.0	93.5
DE	8.9	34.7 ± 0.5	11.0	29.7	54.1
AADE	7.9	30.9 ± 1.7	8.9	33.4	24.2
PE	8.8	35.0 ± 0.8	11.4	28.9	44.1
AAE	**12.8**	**37.1** ± 0.2	**14.5**	**28.6**	88.6
AARGH	12.6	36.6 ± 0.2	14.2	29.0	**89.6**

Table 1: Evaluation of retrieval components of our models (Section 3.3, 3.5). See Section 5.1 for details.

Table 1 shows that AAE and AARGH significantly outperform other setups on all measures except for the no-match rate,[6] where PE has comparable results. This is expected as they use the additional action annotation during training, unlike DE and PE. AADE performs surprisingly bad. According to the unique hints rate, AAE and AARGH retrieve a much wider range of outputs, which could improve lexical diversity of final responses. The higher BLEU, Action IoU and full match rates suggest that the models retrieve responses more similar to the ground truth.

To further compare the approaches to response selection, we computed the Silhouette coefficient (Rousseeuw, 1987) based on the active domain and action annotation (see Table 3).[7] We omit PE

[5] The values were chosen empirically, based on preliminary experiments on development data.

[6] According to a paired t-test with 95% confidence level.

[7] In the case of action-based clustering, we treat each action as a separate cluster; each example can belong to multiple clusters. The clustering measure is calculated for each cluster and averaged over all actions which are weighted by the size

	Setting	BLEU	Inform	Success	Unique trigrams	BCE	Hint-BLEU	Hint-copy	Joint acc.
	Corpus	-	93.7	90.9	25,212	3.37	-	-	-
	SOLOIST (Peng et al., 2021a)	13.6	82.3	72.4	7,923	2.41	-	-	-
	PPTOD (Su et al., 2022)	18.2	83.1	72.7	2,538	1.88	-	-	-
	MTTOD (Lee, 2021)	19.0	85.9	76.5	4,066	1.93	-	-	-
	MinTL (Lin et al., 2020)	19.4	73.7	65.4	2,525	1.81	-	-	-
	Gen (equiv. to MinTL)	18.6 ± 0.3	77.0 ± 1.2	66.4 ± 1.0	3,209	1.94	-	-	54.1 ± 0.2
	AAE (retrieval only)	12.8 ± 0.1	79.9 ± 0.6	58.3 ± 0.7	22,457	3.34	100.0	100.0 %	-
α=0.05	DE +Gen ↓	**17.6** ± 0.3	80.9 ± 0.5	68.8 ± 0.6	8,190	**2.36**	32.5	15.2 %	54.2 ± 0.1
α=0.05	AADE +Gen ↓	17.3 ± 0.3	81.2 ± 0.9	69.1 ± 1.0	6,613	2.29	26.7	12.8 %	54.3 ± 0.1
α=0.05	PE +Gen ↓	17.4 ± 0.3	79.9 ± 0.9	66.8 ± 1.0	7,736	2.35	31.3	14.5 %	**54.4** ± 0.2
α=0.05	AAE +Gen ↓	17.5 ± 0.6	**82.0** ± 1.0	**70.3** ± 0.8	8,152	2.32	32.0	16.2 %	54.2 ± 0.2
α=0.05	AARGH ↓	17.3 ± 0.3	81.2 ± 0.6	69.5 ± 0.5	**8,200**	2.33	28.4	14.2 %	53.8 ± 0.2
α=0.4	DE +Gen ↑	12.3 ± 0.3	87.8 ± 0.3	69.1 ± 0.5	18,800	3.20	80.4	76.5 %	54.2 ± 0.2
α=0.4	AADE +Gen ↑	**14.6** ± 0.4	81.0 ± 0.8	66.7 ± 0.4	10,723	2.72	51.7	44.8 %	54.2 ± 0.1
α=0.4	PE +Gen ↑	12.9 ± 0.4	86.0 ± 0.8	67.1 ± 0.6	16,632	3.13	74.0	69.1%	**54.4** ± 0.1
α=0.4	AAE +Gen ↑	11.9 ± 0.2	**90.5** ± 0.3	**71.3** ± 0.3	19,436	**3.23**	91.1	89.3 %	54.3 ± 0.2
α=0.4	AARGH ↑	12.1 ± 0.2	89.6 ± 0.2	70.7 ± 0.5	**19,813**	3.21	87.6	85.0 %	53.6 ± 0.2

Table 2: Response generation and state tracking evaluation on MultiWOZ using automatic metrics, including the bi-gram conditional entropy (BCE) and number of unique trigrams. We compare previous work, the baseline and retrieval-based generative models. See Section 5.2 for details about the metrics; Section 3, 4 for model descriptions.

Silhouette coefficient	DE	AADE	AAE	AARGH
per Domain	0.098	**0.179**	0.151	0.159
per Action	0.147	0.316	0.312	**0.320**

Table 3: Evaluation of domain and action separation (Section 5.1). We show averages over 8 random seeds.

because its context embeddings depend on queries, i.e., the candidate embeddings (other models output the same context regardless of candidates). DE has the worst results; other systems perform similarly, but AARGH is the best on action separation while AADE has the best scores for domains.

We see that AADE's context encoder is successful in clustering, but it lags behind in terms of correct action selection. Unlike AARGH and AAE, AADE retrieves candidates based on response embeddings. We hypothesize that lower response variability (compared to context variability) leads the model to prefer responses seen more frequently during training. AARGH and AAE are not affected by this as they use purely context-based retrieval.

Figure 3 provides a visualisation of the domain clusters projected using t-SNE (van der Maaten and Hinton, 2008). It supports the findings of our evaluation based on the Silhouette coefficient: We see that visualisations of AARGH and AADE embedding spaces look similarly whereas DE's clusters appear more noisy.

of the corresponding clusters.

5.2 Response generation

We evaluate the response generation abilities of our models using automatic metrics and human assessment of delexicalized texts (see Table 4 for examples).

Evaluation with automatic metrics We use the corpus-based evaluator by Nekvinda and Dušek (2021) to measure commonly used metrics on MultiWOZ (Inform & Success rates, BLEU) as well as lexical diversity measures, namely the number of distinct trigrams in the outputs and bigram conditional entropy (Li et al., 2016; Novikova et al., 2019). State tracking joint accuracy is calculated with scripts adapted from TRADE (Wu et al., 2019). To better understand the effect of using retrieved hints and to quantify the amount of copying, we calculate BLEU between retrieved hints and final generated responses (Hint-BLEU) and the proportion of generated responses exactly matching the corresponding retrieved hints (Hint-copy).

We include comparisons with recent strong end-to-end models on MultiWOZ: SOLOIST (Peng et al., 2021a), MTTOD (Lee, 2021), PPTOD (Su et al., 2022), and MinTL (Lin et al., 2020), which has the same architecture as Gen. To show the importance of the generative parts of our models, we also include AAE without the refining decoder.

Table 2 shows scores obtained with greedy decoding (see Appendix B for beam search results). All models have similar state tracking performance.

System	Response
Context	User: I need a taxi to take me to Yippee Noodle Bar by 9:45.
Corpus	Absolutely, where will the taxi be picking you up from today?
Gen	I can help with that. Where are you departing from?
DE +Gen ↓	Where will you be departing from?
AARGH ↓	Okay, I can help with that. Where are you coming from?
AARGH ↑	Of course, I just need to know your *destination* so I can book that for you.
Context	User: I am looking for an expensive hotel with free parking. System: We have 5 places that offer all of that. Any preference on star rating or area? User: No, that's not important to me. But, I would prefer to stay in a guesthouse with free parking.
Corpus	Sorry, there are no [type] [price] places with free parking. Perhaps you'd prefer a different type or price range?
Gen	I'm sorry, but there are no [type] that meet your criteria. Would you like to try a different price range?
DE +Gen ↓	I'm sorry, but there are no [type] that meet your criteria. Would you like to try a different price range or area?
AARGH ↓	I'm sorry, but I don't have any [type] that meet your criteria. Would you like to try a [type] or a [type]?
AARGH ↑	I'm sorry but I don't have any [type] matching that criteria. Could we look at [type] instead?
Context	User: Thank you so much. You have been very helpful.
Corpus	Great! I hope you enjoy your stay!
Gen	You're welcome. Have a great day!
DE +Gen ↓	You're welcome. Have a great day!
AARGH ↓	Is there anything else I can help you with today?
AARGH ↑	You are welcome. I can also help arrange transportation or suggest attractions, if you wish.

Table 4: Side-by-side system output comparison (see Sections 3 and 4 for model details). The responses are continuations of conversations SNG0016, SNG1048, and MUL2138 from MultiWOZ.

AARGH has slightly lower numbers, which is not surprising as it shares a substantial part of the encoder with its retrieval component. As expected, we notice a huge difference in Hint-BLEU and Hint-copy of versions with different α-blending probabilities (↓ vs. ↑).[8] The performance boost over Gen and retrieval-only AAE is, for ↓ variants, mainly in terms of Success. In ↑, more frequent hint copying reduces BLEU and improves lexical diversity; we also see higher Inform. AAE +Gen and AARGH (both ↓ and ↑) perform better than corresponding DE +Gen or PE +Gen on Inform and Success rates.[9] Differences between AAE +Gen and AARGH are not statistically significant and their Success scores are better than MinTL, competitive with PPTOD and SOLOIST but lower than MTTOD. In terms of lexical diversity, all models are better than most generative baselines.[10]

Human evaluation We arranged an in-house human evaluation on the delexicalized outputs of Gen (i.e., MinTL's architecture), DE +Gen ↓, AARGH ↓ and AARGH ↑. We used side-by-side relative ranking evaluation, which has been repeatedly found to increase consistency compared to rating isolated examples (Callison-Burch et al., 2007; Belz and Kow, 2010; Kiritchenko and Mohammad, 2017). Participants were given full dialog context and current database results, and we asked them to rank responses of the compared models from the best-fitting to the worst, where multiple responses could be ranked the same (see Appendix D for details). We collected rankings for 346 turns of 50 conversations from 5 linguists with experience in natural language generation. All of them were given a different set of dialogs and they were instructed to focus on consistency with the context and database results, naturalness, and attractiveness of the responses. See Table 5 for results.

	Gen	DE +Gen ↓	AARGH ↓	AARGH ↑
Mean Ranking	2.03	1.99	**1.91**	2.3
Ranked #1	36.1%	35.5%	**40.8%**	37.9%
Ranked #2	34.7%	36.7%	33.5%	18.8%
Ranked #3	18.8%	20.5%	19.7%	18.8%
Ranked #4	10.4%	7.2%	**6.1%**	24.6%

Table 5: Human evaluation results – mean ranks (1-4) established from 50 evaluated conversations.

Although AARGH ↑ scored the best on automatic metrics, it has worse mean ranks than other models, which all have similar mean ranks.[11] This confirms previous findings of low correla-

[8] Hint-copy of 15% roughly means one turn per dialog.

[9] According to a paired t-test with 95% confidence level.

[10] The ↓ variants are similar to SOLOIST, which, however, reaches diversity by employng sampling (Holtzman et al., 2020) instead of greedy decoding.

[11] According to Friedman test with 95% confidence level and Nemenyi post-hoc test; only the difference between AARGH ↑ and other models is statistically significant.

tion between automatic metrics and human assessments (Liu et al., 2016b; Novikova et al., 2017). Upon detailed manual error analysis, we found that AARGH ↑ often copies whole hints including words that do not fit the context, i.e., contradictions to earlier statements or noisy non-delexicalized values from the training set. AARGH ↓ performs slightly better than the baselines and is more often ranked best and least often ranked worst.

6 Conclusion

We present AARGH, an end-to-end task-oriented dialog system, combining retrieval and generative approaches. It uses an embedded single-encoder retrieval component which extends a purely generative model without the need for a large number of new parameters. AARGH features an action-aware response selection training objective. Our experiments on the MultiWOZ dataset show that AARGH outperforms baselines in terms of automatic metrics and human evaluation and it is competitive with state-of-the-art models such as SOLOIST or MTTOD. We showed that our proposed action-aware retrieval training objective supports retrieval of a larger variety of unique and relevant responses in the task-oriented setting and makes efficient use of the available system action annotation. Further, using the retrieval module improves dialog management in terms of the Success rate. A limitation of our approach is the need for careful hyperparameter setting, coupled with the risk of overuse of retrieved responses that match the dialogue state but are not appropriate for the context.

In future work, we would like to confirm our results on more datasets and explore more complex ways of usage of the retrieved responses to encourage the model to copy interesting language structures while ignoring inappropriate tokens or relics of faulty delexicalization.

Acknowledgements

This research was supported by Charles University projects GAUK 373921, SVV 260575 and PRIMUS/19/SCI/10, and by the European Research Council (Grant agreement No. 101039303 NG-NLG). It used resources provided by the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth and Sports project No. LM2018101).

References

Rami Al-Rfou, Marc Pickett, Javier Snaider, Yun hsuan Sung, Brian Strope, and Ray Kurzweil. 2016. Conversational Contextual Cues: The Case of Personalization and History for Response Ranking. *arXiv:1606.00372*.

Anja Belz and Eric Kow. 2010. Comparing Rating Scales and Preference Judgements in Language Evaluation. In *INLG 2010 - Proceedings of the Sixth International Natural Language Generation Conference*, Trim, Co. Meath, Ireland.

Pawel Budzianowski and Ivan Vulic. 2019. Hello, It's GPT-2 - How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation@EMNLP-IJCNLP 2019*, pages 15–22, Hong Kong.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a Realistic and Diverse Dialog Dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 4515–4524, Hong Kong, China.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2019a. Skeleton-to-Response: Dialogue Generation Guided by Retrieval Memory. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 1219–1228, Minneapolis, MN, USA.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019b. Retrieval-guided Dialogue Response Generation via a Matching-to-Generation Framework. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019*, pages 1866–1875, Hong Kong, China.

Chris Callison-Burch, Cameron S. Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007.

(meta-) Evaluation of Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation, WMT@ACL 2007*, pages 136–158, Prague, Czech Republic.

Debanjan Chaudhuri, Agustinus Kristiadi, Jens Lehmann, and Asja Fischer. 2018. Improving Response Selection in Multi-Turn Dialogue Systems by Incorporating Domain Knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018*, pages 497–507, Brussels, Belgium.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-, Volume 1: Long Papers*, pages 3696–3709.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, MN, USA.

Nouha Dziri, Andrea Madotto, Osmar Zaïane, and Avishek Joey Bose. 2021. Neural Path Hunter: Reducing Hallucination in Dialogue Systems via Path Grounding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 2197–2214, Virtual Event / Punta Cana, Dominican Republic.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset With State Corrections and State Tracking Baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France.

Prakhar Gupta, Jeffrey P. Bigham, Yulia Tsvetkov, and Amy Pavel. 2021. Controlling Dialogue Generation with Semantic Exemplars. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 3018–3029, Online.

DongHoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-End Neural Pipeline for Goal-Oriented Dialogue Systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 583–592, Online.

Matthew Henderson, Paweł Budzianowski, Iñigo Casanueva, Sam Coope, Daniela Gerz, Girish Kumar, Nikola Mrkšić, Georgios Spithourakis, Pei-Hao Su, Ivan Vulić, and Tsung-Hsien Wen. 2019a. A Repository of Conversational Datasets. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 1–10, Florence, Italy.

Matthew Henderson, Iñigo Casanueva, Nikola Mrkšić, Pei-Hao Su, Tsung-Hsien Wen, and Ivan Vulić. 2020. ConveRT: Efficient and Accurate Conversational Representations From Transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2161–2174, Online.

Matthew Henderson, Ivan Vulic, Daniela Gerz, Iñigo Casanueva, Pawel Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrksic, and Pei-Hao Su. 2019b. Training Neural Response Selection for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-, Volume 1: Long Papers*, pages 5392–5404.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A Simple Language Model for Task-Oriented Dialogue. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS*.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. In *8th International Conference on Learning Representations, ICLR 2020*, Addis Ababa, Ethiopia.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-Text Pre-Training for Data-to-Text Tasks. In *Proceedings of the 13th International Conference on Natural Language Generation, INLG 2020*, pages 97–102, Dublin, Ireland.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tür. 2020. Beyond Domain APIs: Task-oriented Conversational Modeling with Unstructured Knowledge Access. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue, SIGdial 2020*, pages 278–289, Online.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015Proceedings*, San Diego, CA, USA.

Svetlana Kiritchenko and Saif M. Mohammad. 2017. Best-Worst Scaling More Reliable than Rating Scales: A Case Study on Sentiment Intensity Annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017 4, Volume 2: Short Papers*, pages 465–470, Vancouver, Canada.

Yohan Lee. 2021. Improving End-to-End Task-Oriented Dialog System with A Simple Auxiliary Task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303, Virtual Event / Punta Cana, Dominican Republic.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego California, USA.

Andrea Lin, Zhaojiang andMadotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist Transfer Learning for Task-Oriented Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016a. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 2122–2132, Austin, Texas, USA.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. 2016b. How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, pages 2122–2132, Austin, Texas, USA.

Ryan Lowe, Nissan Pow, Iulian Serban, and Joelle Pineau. 2015. The Ubuntu Dialogue Corpus: A Large Dataset for Research in Unstructured Multi-Turn Dialogue Systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294, Prague, Czech Republic.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data Using T-Sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of BLEU, Flavours of Success: The Case of MultiWOZ. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46, Online.

Jekaterina Novikova, Aparna Balagopalan, Ksenia Shkaruta, and Frank Rudzicz. 2019. Lexical Features Are More Vulnerable, Syntactic Features Have More Predictive Power. In *Proceedings of the 5th Workshop on Noisy User-generated Text, W-NUT@EMNLP 2019*, pages 431–443, Hong Kong, China.

Jekaterina Novikova, Ondrej Dusek, Amanda Cercas Curry, and Verena Rieser. 2017. Why We Need New Evaluation Metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2241–2252, Copenhagen, Denmark.

Aigul Nugmanova, Andrei Smirnov, Galina Lavrentyeva, and Irina Chernykh. 2019. Strategy of the Negative Sampling for Training Retrieval-Based Dialogue Systems. In *IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2019*, pages 844–848, Kyoto, Japan.

Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar Encoder-Decoder for Neural Conversation Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers*, pages 1329–1338, Melbourne, Australia.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002*, pages 311–318, Philadelphia, PA.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021a. SOLOIST: Building Task Bots at Scale with Transfer Learning and Machine Teaching. *Trans. Assoc. Comput. Linguistics*, 9:907–824.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2021b. RADDLE: An Evaluation Benchmark and Analysis Platform for Robust Task-oriented Dialog Systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers)*, pages 4418–4429, Virtual Event.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models Are Unsupervised Multitask Learners. Technical report, Open AI.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 8689–8696, New York, NY, USA.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for Building an Open-Domain Chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021*, pages 300–325, Online.

Peter J. Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Igor Shalyminov, Alessandro Sordoni, Adam Atkinson, and Hannes Schulz. 2020. Hybrid Generative-Retrieval Transformers for Dialogue Domain Adaptation. In *DSTC8@AAAI*, New York, NY, USA.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-Task Pre-Training for Plug-and-Play Task-Oriented Dialogue System. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022*, pages 4661–4676, Dublin, Ireland.

David Thulke, Nico Daheim, Christian Dugast, and Hermann Ney. 2021. Efficient Retrieval Augmented Generation From Unstructured Knowledge for Task-Oriented Dialog. In *DSTC9@AAAI*, Online.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30.

Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez-Moreno. 2018. Generalized End-to-End Loss for Speaker Verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018*, pages 4879–4883, Calgary, AB, Canada.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2021. Modelling Hierarchical Structure between Dialogue Policy and Natural Language Generator with Option Framework for Task-oriented Dialogue System. In *9th International Conference on Learning Representations, ICLR 2021*, Virtual Event, Austria.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Peihao Su, David Vandyke, and Steve J. Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1711–1721, Lisbon, Portugal.

Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina Maria Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve J. Young. 2017. A Network-based End-to-End Trainable Task-oriented Dialogue System. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Volume 1: Long Papers*, pages 438–449, Valencia, Spain.

Jason Weston, Emily Dinan, and Alexander H. Miller. 2018. Retrieve and Refine: Improved Sequence Generation Models For Dialogue. In *Proceedings of the 2nd International Workshop on Search-Oriented Conversational AI, SCAI@EMNLP 2018*, pages 87–92, Brussels, Belgium.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos*, pages 38–45, Online.

Chien-Sheng Wu, Steven C. H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 917–929, Online.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28-, Volume 1: Long Papers*, pages 808–819.

Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proc. IEEE*, 101(5):1160–1179.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A Dialogue Dataset With Additional Annotation Corrections and State Tracking Baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020a. Task-Oriented Dialog Systems That Consider Multiple Appropriate Responses under the Same Context. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 9604–9611, New York, NY, USA.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT : Large-Scale Generative Pre-training for Conversational Response Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020*, pages 270–278, Online.

A Model Architectures

Figure 4 shows architectures of the baseline (Gen), dual-encoder-based model (DE), and single-encoder action-aware model (AAE). See Figure 1 for details about AARGH and Section 3 for description of the models.

B Beam Search Results

See Table 6 for the results of beam search-based response generation evaluation, and compare the results with greedy decoding evaluation (see Section 5.1 and Table 2). For all models, we used beams of size 8 during the decoding

In the case of conservative α-blending, beam search decoding results in higher lexical diversity for all retrieval-augmented systems. However, the gains with respect to Inform and Success rates are mostly very small or not present at all in the case of AADE and AARGH. All BLEU scores are slightly lower which corresponds with the higher output diversity. We notice that the numbers for the baseline without a retrieval component have an opposite trend. Beam search decoding causes lower lexical diversity and higher BLEU. We attribute this to the fact that beam search decoding prefers safer responses with a higher overall probability.

When using higher α-blending, the differences become small even in the case of lexical diversity. We hypothesize that all the retrieval-based models are not substantially influenced by the particular response decoding strategy because they strongly rely on the retrieved hints and their copying.

C End-to-end Conversation

Figure 5 shows a multi-domain (restaurant and taxi) end-to-end conversation between our end-to-end retrieval-based model AARGH (See Section 3.5).

D Human Evaluation Interface

We used the graphical user interface depicted in Figure 6 for human evaluation. A full dialog context, i.e., all past utterances corresponding to the particular turn, and the number of database results were shown to participants. We asked participants to rank provided responses from the best to the worst. They evaluated only two conversations in a single run and we sampled the conversations from the test set so that all participants receive roughly the same number of turns to assess. Evaluated responses were shown side-by-side; each of them had a dedicated discrete scale from 1 to 4 where 1 was labeled as the best and 4 as the worst. More responses could receive the same ranking. Participants could move forward and backward in the conversations and they could switch to another conversation anytime.

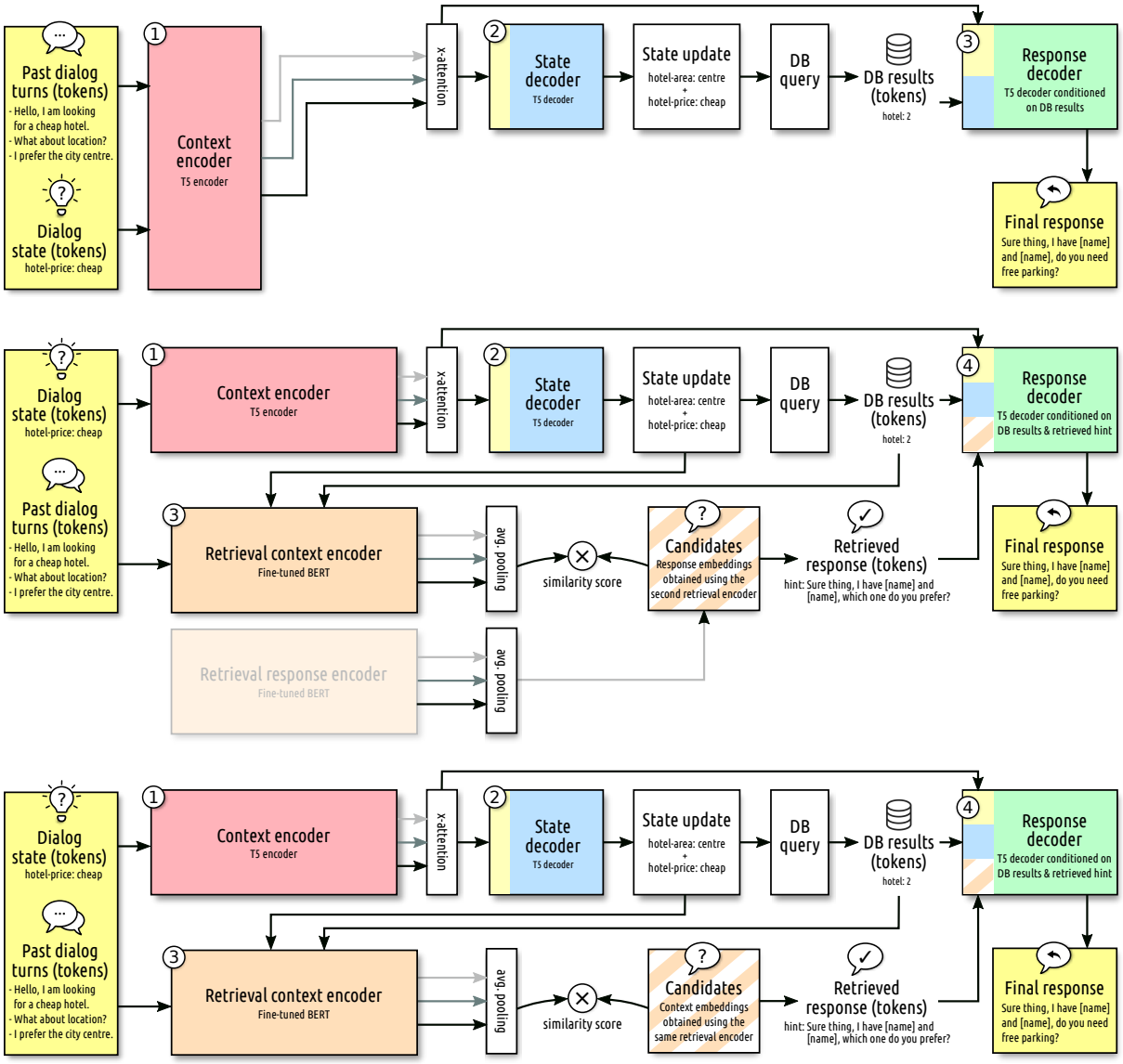


Figure 4: Architecture of the baseline (Gen, *top*), dual-encoder-based model (DE, *middle*) and single-encoder action-aware model (AAE, *bottom*). Numbers in module boxes mark the order of processing during inference.

	Setting	BLEU	Inform	Success	Num. trigrams	Bi-gram entropy	Hint-BLEU	Hint-copy
	Gen	19.1 ± 0.3	73.1 ± 1.8	63.0 ± 1.7	2683	1.81	-	-
$\alpha = 0.05$	DE	16.1 ± 0.3	81.1 ± 0.5	68.3 ± 0.8	10098	**2.49**	41.9	25.2 %
	AADE	16.0 ± 0.4	78.0 ± 1.1	65.9 ± 1.0	7378	2.33	32.6	19.2 %
	PE	15.9 ± 0.4	80.6 ± 0.9	66.9 ± 1.0	9470	2.48	40.7	24.4 %
	AAE	$\mathbf{16.4} \pm 0.4$	$\mathbf{82.5} \pm 0.8$	$\mathbf{69.8} \pm 0.6$	**10457**	2.46	44.2	29.0 %
	AARGH	16.2 ± 0.3	79.5 ± 0.5	68.0 ± 0.3	9072	2.36	36.2	22.2 %
$\alpha = 0.4$	DE	12.3 ± 0.3	87.7 ± 0.3	68.7 ± 0.6	19103	**3.28**	83.1	79.6 %
	AADE	13.7 ± 0.4	77.8 ± 0.9	63.2 ± 0.4	10997	**2.76**	55.8	50.2 %
	PE	$\mathbf{12.7} \pm 0.3$	85.9 ± 0.9	66.3 ± 0.7	17178	3.19	78.3	74.2 %
	AAE	12.0 ± 0.2	$\mathbf{90.3} \pm 0.5$	$\mathbf{71.2} \pm 0.4$	19448	3.21	92.6	90.1 %
	AARGH	12.2 ± 0.2	89.3 ± 0.3	70.0 ± 0.6	**19763**	3.22	88.6	86.0 %

Table 6: Beam search-based response generation on MultiWOZ using automatic metrics. For each model setup, We use beams of size 8 during response decoding and report results averaged over 8 random seeds. We compare the baseline (Gen) and retrieval-based generative models (See Section 3 and 4). See Section 5.2 for details about the metrics. Cf. Table 2 showing results obtained using greedy decoding.

USR Hello, I need to book a restaurant. It must be cheap and in the city centre. Can you help me?

Certainly, there are [choice] [pricerange] restaurants to choose from in the [area]. Are you in the mood for any particular type of food?

I can help you with that. There are several cheap restaurants in the centre. Are you looking for any particular type of food? AARGH

USR I would like to taste French cuisine. Do you have some restaurants with frogs and similar food?

There are no [food] restaurants located in [area]. Are you interested in another type of cuisine?

I'm sorry, there are no french restaurants in the centre. Would you like to try a different type of cuisine? AARGH

USR Ok, no problem. I actually like noodles too. What about Chinese restaurants?

There are [choice] such restaurants. I reccommend [name]. Would you like a reservation?

There are 3 such restaurants in the centre. Charlie Chan and Rice House. Would you like to book one of these? AARGH

USR Book Charlie Chan for three persons on Friday 8pm please.

Booking was successful for [bookday]. The table will be reserved for 15 minutes. Reference number is: [ref]. Can I help with anything else today?

Booking was successful. The table will be reserved for 15 minutes. Reference number is: 0000010. Can I help with anything else today? AARGH

USR I also need a taxi from El Shaddai to the restaurant.

Of course! What time would you like the taxi to leave or arrive?

What time would you like the taxi to leave or arrive by? AARGH

USR I need to be in the restaurant at 19:45.

I booked a [type] to pick you up. The contact number for the taxi is [phone].

I booked a yellow lexus to pick you up. The contact number for the taxi is 13582494566. AARGH

USR Ok thanks, that's all for today.

Thank you for allowing me to help you. Have a great day!

You're welcome. Have a great day! AARGH

Figure 5: End-to-end conversation between ● the user and ● our retrieval-based AARGH model with conservative α-blending (see Section 3). For the system turns, we show delexicalized hints proposed by the retrieval module (left boxes in italics) and the corresponding lexicalized final responses (right boxes). We highlighted ● the parts of hints present in the final texts and ● the parts of final responses newly-introduced by the model during refining.

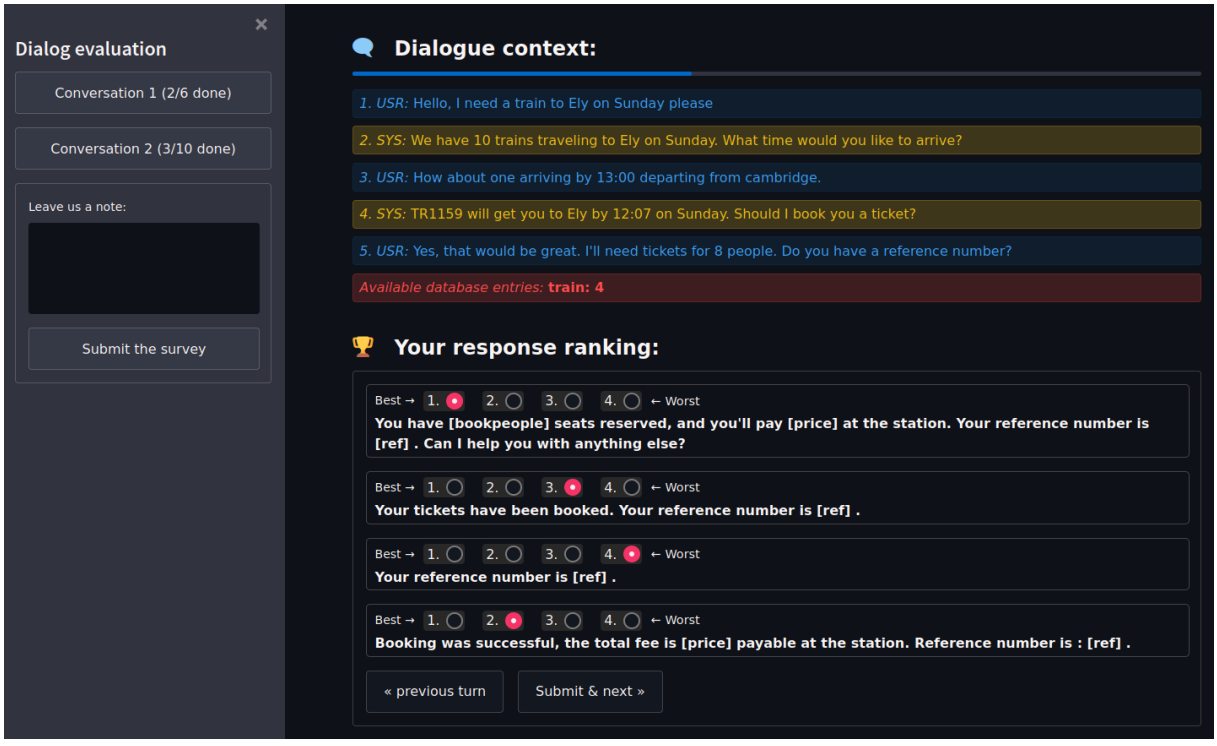


Figure 6: Our graphical user interface used for human evaluation.

A Systematic Evaluation of Response Selection for Open Domain Dialogue

Behnam Hedayatnia, Di Jin, Yang Liu, Dilek Hakkani-Tur
Amazon Alexa AI
{behnam,djinamzn,yangliud,hakkanit}@amazon.com

Abstract

Recent progress on neural approaches for language processing has triggered a resurgence of interest on building intelligent open-domain chatbots. However, even the state-of-the-art neural chatbots cannot produce satisfying responses for every turn in a dialog. A practical solution is to generate multiple response candidates for the same context, and then perform response ranking/selection to determine which candidate is the best. Previous work in response selection typically trains response rankers using synthetic data that is formed from existing dialogs by using a ground truth response as the single appropriate response and constructing inappropriate responses via random selection or using adversarial methods. In this work, we curated a dataset where responses from multiple response generators produced for the same dialog context are manually annotated as appropriate (positive) and inappropriate (negative). We argue that such training data better matches the actual use case examples, enabling the models to learn to rank responses effectively. With this new dataset, we conduct a systematic evaluation of state-of-the-art methods for response selection, and demonstrate that both strategies of using multiple positive candidates and using manually verified hard negative candidates can bring in significant performance improvement in comparison to using the adversarial training data, e.g., increase of 3% and 13% in Recall@1 score, respectively.

1 Introduction

Building an open-domain dialog system to interact with users on a variety of topics can involve building multiple response generators (RG) with different functions (Paranjape et al., 2020). These RGs can be a mixture of generative, retrieval and template based methods. A response selector is then built to re-rank response candidates produced by different applicable RGs to determine the best response for a given turn. These response selectors are based on either rule-based or model-based architectures (Papaioannou et al., 2017; Serban et al., 2017; Zhou et al., 2020; See and Manning, 2021a). Rule-based systems typically consist of manually-designed logic to rank hypotheses, whereas model-based approaches can either be conventional machine learning models or recent neural models that learn to rank candidates. As the number of RGs grows, a rule-based system can become cumbersome to maintain, whereas model-based methods can simplify the selection process as well as achieve better performance.

Latest work in model-based response selectors involves leveraging pretrained transformer models such as BERT (Devlin et al., 2019) and DialoGPT (Zhang et al., 2019). These selection models are often trained using existing dialog datasets that typically contain ground truth responses. Thus a focus of past response selection work is on the construction of inappropriate/negative responses, using methods such as random selection, utterance manipulation or leveraging user feedback (Whang et al., 2020; Han et al., 2021; Whang et al., 2021; Gu et al., 2020; Xu et al., 2020; Zhang and Zhao, 2021; Gao et al., 2020; See and Manning, 2021b; Gupta et al., 2021; Li et al., 2019). However, such synthesized datasets for response selection have the following known drawbacks. First of all, their claimed incorrect responses are not verified if they are actually incorrect. Second, these negative responses are easy to differentiate from positive ones since it is very likely that they will be on different topics from the context. Therefore, models trained on such easy negative responses will not be able to generalize to real-world settings, where multiple responses are generated given the same dialog context and many of them are strong candidates.

To resolve the aformentioned issues, we construct a new dataset (named RSD) for response selection by showing human annotators multiple response candidates produced by different RGs for

Proceedings of the SIGdial 2022 Conference, pages 298–311
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

a given turn and dialog context, and asking them to annotate all responses that are appropriate for that specific dialog context. We leverage RSD to conduct a systematic evaluation of state-of-the-art methods for response selection, including existing trained models, DialogRPT (Gao et al., 2020) and BERT-FP (Han et al., 2021), and a BERT based ranker that we trained. Our experimental results show the following findings: (1) Models trained on RSD significantly outperform those trained on existing datasets, e.g., Reddit and Ubuntu, showing the benefit of bringing in human annotated data for this task; (2) Using manually verified hard negatives greatly outperforms using adversarial negatives; (3) Training on multiple positive candidates improves performance in comparison to a single positive candidate. Though these findings are most expected, this is the first empirical study that clearly shows that constructing a more realistic dataset benefits strongly over generating synthetic examples for response selection, and we hope such results can guide future research in this direction and deployment of open domain dialog systems.

2 Related Work

Previous work in response selection has been conducted in different domains, such as chatlogs (Lowe et al., 2015), e-commerce (Zhang et al., 2018b), and open-domain dialog (Wu et al., 2017; Zhang et al., 2018a; Smith et al., 2020; See and Manning, 2021b). Our work focuses on open-domain dialog, where current systems typically consist of multiple response generators, each of which is designed to deal with a certain domain. For example, in the Alexa Prize challenge (Ram et al., 2018; Gabriel et al.), most of the participating socialbots built by university teams consist of a variety of responders that are based on retrieval-based methods, template-based methods, or generative models (Konrád et al., 2021; Saha et al., 2021; Paranjape et al., 2020; Ram et al., 2018). In order to select the final response to present to users, both rule-based or model-based ranking models have been proposed (Ram et al., 2018; Papaioannou et al., 2017; Serban et al., 2017; Zhou et al., 2020; See and Manning, 2021a; Shalyminov et al., 2018). This approach is also common in other real-world systems such as XiaoIce that employs a manually-designed set of features to rank hypotheses (Zhou et al., 2020).

For training response selection models, typically human-human dialogs are used, where positive examples are the ground truth responses and negative responses are often randomly selected or synthetically created since there are no labeled negative responses. Han et al. (2021) randomly selected responses from other dialogs or within the same dialog session. Whang et al. (2021) corrupted utterances by inserting, substituting and deleting random tokens. Xu et al. (2020) masked and shuffled utterances within a dialog. Li et al. (2019) selected negative responses from a batch based on their similarity scores from the positive response score. Gupta et al. (2021) used automatic methods such as replacing random tokens in a positive examples using a Mask-and-fill approach to create adversarial negative examples.

However, these sampling strategies do not ensure the selected negative responses are hard examples. In this work, rather than relying on approximation for negative responses, we perform turn level annotation of multiple response candidates for response appropriateness for a given dialog context. See and Manning (2021b); Gao et al. (2020) did construct hard negative examples by annotating responses from a single generative model for appropriateness; however, our work contains responses from a mixture of various RG methods.

On the other hand, open-domain dialogs can have multiple appropriate responses for a given dialog context. Previous work has augmented dialog datasets with multiple positive examples (Mizukami et al., 2015; Khayrallah and Sedoc, 2020; Gupta et al., 2019; Sai et al., 2020; Zhang et al., 2020). Within open-domain dialogs, Gupta et al. (2019); Sai et al. (2020) augmented the DailyDialog dataset (Li et al., 2017) with multiple positive human written responses. In contrast, our dataset has multiple positive responses generated from models, which reduces the cost of human annotation significantly. The closest work to ours is (Sai et al., 2020) that constructed negative examples by asking annotators to copy information from the dialog context. We do not restrict the definition of negative examples to be copying information from the dialog context, since incorrect responses in open-domain dialog can have different issues, e.g., off-topic, contradicting or repetitive responses.

3 Datasets

As described earlier, most previous work in response selection has constructed test sets that typi-

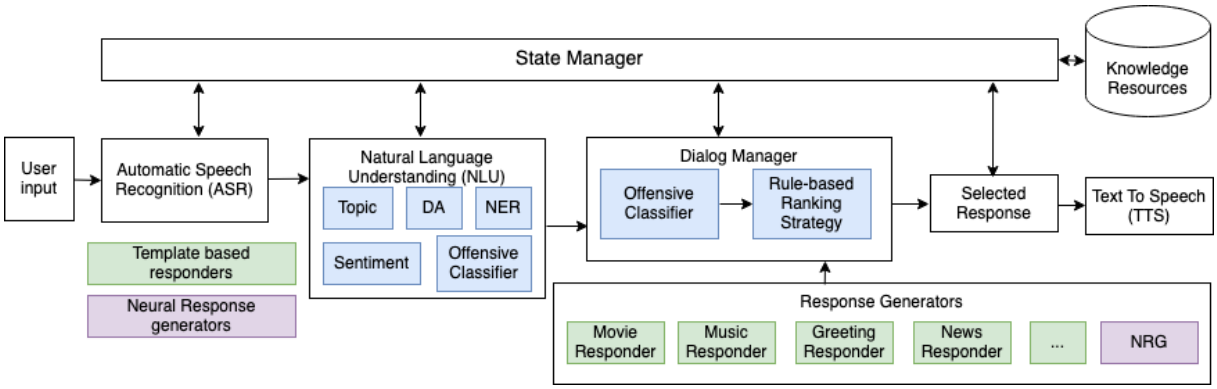


Figure 1: Architecture of our Open Domain Dialog System. NER = Named Entity Recognition, DA = Dialog Act

cally contain only one positive candidate and one or more synthetically created negative candidates. However, such negative responses may be easy for a model to detect. Additionally, in real-world open-domain dialogs there can be more than one positive response per turn. Therefore, in this work we constructed a more realistic dataset consisting of annotations for real response candidates. Our dataset consists of spoken interactions between a dialog system and real users.

3.1 Open Domain Dialog System

We first describe the open-domain dialog system used for data collection. The architecture of our dialog system is shown in Figure 1. Every user utterance in the dialog is sent into an ASR system whose output goes through a series of NLU modules that classifies topics, dialog acts, sentiment, extracts entities, and detects if user utterance is offensive. Our system then calls multiple response generators for the given dialog context and logs all the generated response candidates within the State Manager. The response presented to the user is selected by a rule-based ranker and then sent to the TTS module.

For popular topics in open domain dialogs, such as movies, music, recent news, we developed template-based response generators (highlighted in green in Figure 1) for the given dialog state. An example state and response for the movie domain is: when the user turn mentions a movie name (based on the NER result), we respond with information about the actor, the rating, or the plot of this certain movie. In addition to topic-specific template-based RGs, our system includes other template-based RGs for different dialog contexts, such as, greetings, topic switches, etc.

For every user turn, we also apply a neural network-based response generation (NRG) model to produce a response, highlighted in purple in Figure 1. Our *NRG Responder* is a GPT2-XL (Radford et al., 2019) based model trained on real user conversation data described in Section 3.2. We discuss its training details in Appendix B.

The rule-based ranker uses predefined logic and the topic extracted from the user utterance to select domain specific template-based responders. If a template-based responder is not available it will use the NRG response as a fall back. Our system has just a few template-based RGs, and uses NRG responses for almost half of all turns.

3.2 Response Selection Data (RSD)

We deploy the dialog system described above within the Alexa Prize Socialbot framework (Ram et al., 2018) to interact with real users. A user initiates an interaction with our dialog system and consents to have their data being collected. These interactions end when the user requests to stop the conversation. At the end of each interaction, users are asked to leave a rating in the range of 1 to 5. We denote this dataset as real user interactions (RUI)[1]. Our data consists of approximately 100k interactions and 2.5 million turns. For each user turn in RUI, we produced additional response candidates using variants of our *NRG Responder* to supplement the logged responses. These may be appropriate responses, or hard negative examples. The NRG variants we used include the following (Further model training details are in Appendix B).

- A GPT2-medium version of our *NRG Responder*.
- A GPT2-XL *NRG Responder* grounded on knowledge. When there is an entity in the user

[1]All interactions are in English.

Data Split	# Dialogs	# positive responses	# negative responses	Avg. # responses at each turn	# Turns with no positive responses
RSD Train	1,501	17,778	78,273	5.67	8,871
RSD Test	142	2,995	6,298	5.36	309

Table 1: Dataset Statistics. For our experiments, we conduct 5 fold cross validation on all our training datasets and therefore do not have a dedicated development set.

turn, we search Wikipedia to find the article related to the entity, and perform knowledge selection and knowledge-grounded response generation.

- A GPT2-medium *NRG Responder* grounded on dialog acts (DA) (Hedayatnia et al., 2020).
- A GPT2-XL based sentiment controlled *NRG Responder*. When the user's utterance shows some negative sentiment (e.g., when a person says "I'm depressed"), the NRG model generates a response conditioned on this emotion.

We worked with internal human annotators to set up an annotation pipeline. These internal annotators are not experts in the dialog domain; however, we worked closely with them to ensure they have a clear understanding of the task provided to them. In our annotation pipeline, for each turn in a dialog, we showed internal human annotators all the available responses produced by the template based generators and various NRG models[2], and asked them whether each response candidate is appropriate given the certain dialog context. An annotator can label multiple responses or none of them as appropriate. To determine if a response is appropriate we ask annotators to see if the response is relevant to the dialog context and that it does not contradict what was said in previous dialog system's responses. For data annotation we randomly sampled a subset of RUI that contain dialogs with more than 5 turns and fewer than 30 turns. A snapshot of the interface for the annotation task can be found in Appendix C.

We randomly split the annotated conversations into training and test sets. Table 1 shows the statistics of our annotated response selection data, denoted as RSD. Due to user privacy constraints, we cannot release this data. Note that we assume our response selector must always choose a response and therefore we drop turns where none of the responses are labeled as appropriate, and for each turn, we may have multiple positive and negative responses.

[2]Note that for the NRG models, we only use responses produced within a pre-defined timeout period.

3.3 RSD Training Variations

To show the importance of using hard negative and multiple positive candidates for response selection, we have also created five variations of the train set of RSD. In our experiments, for each variation we ran random sampling five times and report the average results.

- RSD Train with one positive candidate (denoted as "RSD 1 Pos."). Based on the original RSD Train, we sample only one positive candidate for each turn from the multiple positive candidates, and keep all the annotated negative responses. This leads to 8,046 positive and 78,273 negative candidates.

- Synthetic Inter-Random. Based on the abovementioned "RSD 1 Pos." set, we further remove the human annotated negative candidates, and instead use five randomly selected responses from other dialogs and deem these as the new negative candidates. There are 8,046 positive and 40,230 negative candidates in this set. This approach to constructing negative candidates is commonly used in the literature. We experimented with different number of negative candidates and found sampling 5 negative candidates at each turn had the best results.

- Synthetic Intra-Random. Similar to the above set, we use one positive example and four randomly selected responses as negative, two drawn from a random different dialog and the other two from the same dialog as the candidate we are training on. This set contains 8,046 positive and 32,184 negative candidates. This approach to constructing negative candidates is proposed by (Han et al., 2021). We experimented with different number of negative candidates and found sampling 4 negative candidates at each turn had the best results.

- Synthetic Adversarial. Based on the abovementioned "RSD 1 Pos." set, we further create negative candidates using the Mask-and-Fill

approach from (Gupta et al., 2021). This approach uses the hierarchical masking function from (Donahue et al., 2020) to replace spans in a positive example with blank tokens that will be replaced with tokens predicted from an Infilling Language Model from (Donahue et al., 2020). For every turn, an average of 28.22 negative candidates were constructed using this approach. We experimented with different number of negative candidates and found sampling 10 negative candidates at each turn had the best results. In total, we have 8,046 positive and 76,307 negative candidates.

- Synthetic Retrieval. In this approach, we generate negative examples that are semantically similar to the positive example. This approach to constructing negative candidates is proposed by (Li et al., 2019). The motivation behind this approach is to create negative candidates that are somewhat similar to the positive candidate and use these as hard examples for the model to train on. Specifically, we use the *all-MiniLM-L6-v2* model (Wang et al., 2020) from HuggingFace[3] and create a sentence embedding for each response in our dataset. At each turn we compute the cosine similarity between the positive candidate and all the other responses in the dataset. We then take responses that have a cosine similarity between 0.8 and 0.95 as a negative candidate. We experimented with different thresholds and found this had the best results. Using these thresholds we get an average of 2.2 negative candidates per turn. In total we have 8,046 positive and 17,778 negative candidates.

4 Response Selection Models

We have adopted two state-of-the-art methods for response selection and adapted them to our new dataset for a comprehensive empirical evaluation.

4.1 DialogRPT (Gao et al., 2020)[4]

DialogRPT is initialized with DialoGPT (Zhang et al., 2019) and trained using a contrastive loss function to predict a higher score for the positive response given the dialog context and a pair of one positive and one negative response. Trained on the Reddit dataset, five different ranker models are proposed by training DialogRPT on different synthesized labels (see the original paper for details).

4.2 BERT Models

We experiment with two different BERT model variants for response selection:

BERT-FP (Han et al., 2021)[5]: BERT-FP has achieved high scores on the Ubuntu Dialogue Corpus test set (Lowe et al., 2015). The authors post-train the Masked Language Model (MLM) head and Next Sentence Prediction (NSP) head of a BERT-base model (Devlin et al., 2019) on the Ubuntu corpus via unsupervised learning. Given a dialog context and a response, the NSP head is trained to predict whether a response is either: the ground truth, from a random dialog, or from a random turn in the same dialog. After post-training, the model is further fine-tuned on downstream data for response selection, where given a dialog context and a system response, the model classifies whether this is the correct response or not.

BERT-Ranker: We directly fine-tune a BERT-base (Devlin et al., 2019) model without the above-mentioned post-training step. We denote this model as BERT-Ranker.

Figure 2 illustrates the fine-tuning stage for both BERT models. To construct our input, we concatenate the dialog context with a system response and follow the same training procedure used by (Han et al., 2021), which uses the pooled output representation by the BERT model, passes it through a linear layer followed by a sigmoid function, and minimizes the binary cross-entropy function to predict whether the given system response is positive or negative.

5 Experiments

5.1 Experimental Setup

Following the previous work (Whang et al., 2020; Han et al., 2021; Whang et al., 2021; Gu et al., 2020; Xu et al., 2020; Zhang and Zhao, 2021), for evaluation metrics, we use MRR (mean reciprocal rank) and Recall at k (R@k), which is defined as the correct answer existing among the top-k candidates.

For DialogRPT, we run their five different rankers out of the box over RSD Test in a zero-shot fashion and find that the human vs random ranker scores the highest for both MRR and Recall,

[3]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[4]https://github.com/golsun/DialogRPT/

[5]https://github.com/hanjanghoon/BERT_FP

Model	Train Data	MRR	R@1	R@2	R@3	R@4	R@5
DialogRPT	Reddit	0.681	0.481	0.730	0.868	0.939	0.988
BERT-FP	Ubuntu	0.684	0.486	0.742	0.864	0.930	0.979
DialogRPT	RSD Train	0.787	0.647	0.834	0.910	0.979	0.992
BERT-FP	RSD Train	0.795	0.657	0.841	0.931	0.973	0.994
BERT-R	RSD Train	**0.796**	**0.659***	**0.843***	**0.936***	**0.980***	**0.995***
BERT-R	RSD 1 Pos.	0.762(0.06)	0.628(0.05)	0.806(0.06)	0.894(0.07)	0.941(0.07)	0.958(0.07)
BERT-R	Synthetic IE	0.688(0.01)	0.488(0.01)	0.741(0.00)	0.880(0.00)	0.949(0.00)	0.984(0.00)
BERT-R	Synthetic IA	0.698 (0.00)	0.506 (0.00)	0.750 (0.00)	0.879 (0.00)	0.948 (0.00)	0.983 (0.00)
BERT-R	Synthetic Adv	0.712 (0.00)	0.532 (0.00)	0.753 (0.00)	0.884 (0.00)	0.950 (0.00)	0.987 (0.00)
BERT-R	Synthetic Ret	0.718 (0.00)	0.533 (0.01)	0.776 (0.00)	0.902 (0.00)	0.961 (0.00)	0.990 (0.00)

Table 2: Model results on RSD Test. Results for BERT-R (Ranker) using Synthetic datasets are computed by sampling candidates with five different seeds and averaging the model prediction results across those runs. Standard deviations are in parentheses. IE (Inter-Random), IA (Intra-Random), Adv (Adversarial), Ret (Retrieval) are the four different ways of creating negative examples described in Section 3.3. Recall numbers marked with * mean that the improvement is statistically significant compared with Synthetic Ret (mcnemar with p-value < 0.05).

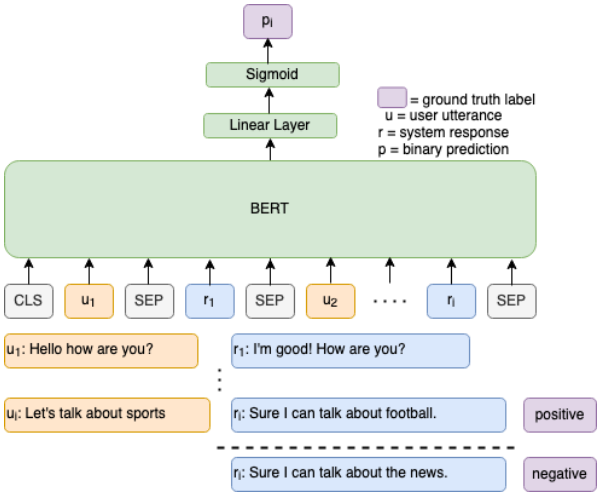


Figure 2: Model architecture of BERT-Ranker and BERT-FP.

therefore we fine-tune this model on RSD Train following the same training approach in the original paper. Since we have p positive and n negative candidates for each turn, we can obtain $p \times n$ example pairs. For our BERT models, we finetune both BERT-FP and BERT-Ranker on RSD Train. To evaluate the effect of positive and negative examples, we finetune the BERT-Ranker using different RSD training variations described in Section 3.3.

We also implemented model ensembling for all the methods. We first divide the training set into five folds, and each time we choose four of them for model training and the remaining one for validation. In this way, we obtain five trained models, and then average their prediction probability outputs on the test set to get the final prediction scores. Further training details are provided in Appendix A.

5.2 Results

Table 2 shows the results on RSD Test using different models and training configurations. From the table, we have the following findings:

- We observe that there is no performance improvement when training BERT-FP on RSD Train versus BERT-Ranker on RSD Train. Therefore the post-training process via optimizing the MLM and NSP objectives proposed in the BERT-FP model does not bring an extra advantage.
- By comparing DialogRPT trained on both Reddit and RSD Train as well as BERT-FP trained on Ubuntu and RSD Train, we can see that the same models trained on our labeled data lead to much better performance, because of the matched training and testing setup.
- We observe that training BERT-Ranker on adversarially created negatives (Synthetic Adv.) and (Synthetic Ret.) outperforms using random negatives within the same dialog (Synthetic IA), achieving Recall@1 scores 0.532 and 0.506, respectively. However, training on adversarial examples (Synthetic Adv.) and (Synthetic Ret.) still significantly underperforms training on human-verified hard negatives (RSD Train), which achieved a Recall@1 score of 0.659.
- We see the benefit of leveraging multiple positive responses, i.e., BERT-Ranker (RSD Train) outperforms BERT-Ranker (RSD 1 Pos.) with Recall@1 scores of 0.659 and 0.628, respectively.

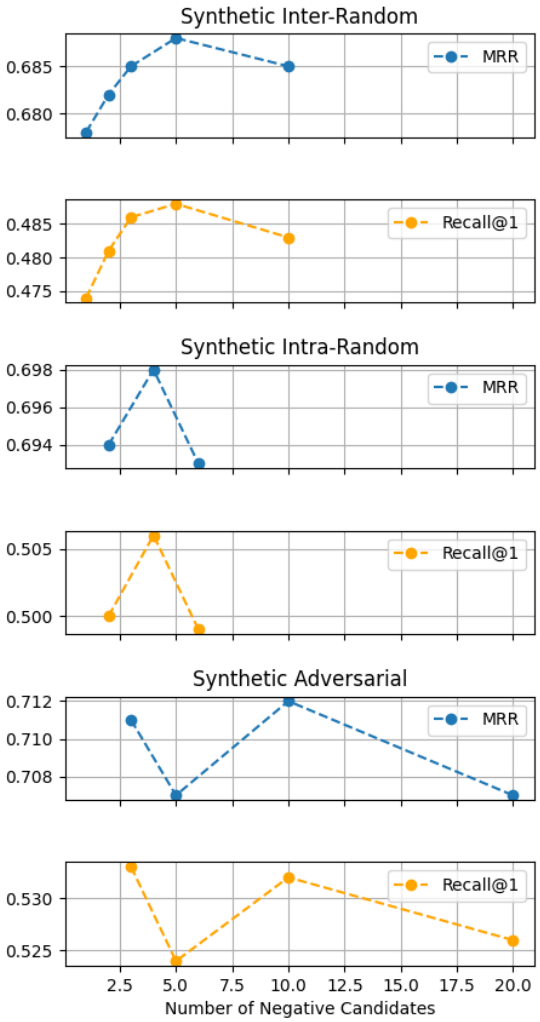


Figure 3: Results when increasing the number of negative candidates in the Synthetic Datasets Training set. See Recall@2-5 results in Appendix D

Example 1
Dialog History: **System:** If you don't mind me asking, what type of music are you into ? **User**: i like rock my favorite band is the arctic monkeys
BERT Ranker (RSD Train): Thanks for sharing! If you don't mind me asking, what is your favorite rock song ?
BERT Ranker (Synthetic Inter-Random): i see, you like rock your favorite band is the arctic monkeys. i like rock too. who is your favorite artist?

Example 2
Dialog History: **System:** i'm happy to talk about movies. so, have you seen a good movie lately? **User**: oh yeah
BERT Ranker (RSD Train): nice. what movie did you see?
BERT Ranker (Synthetic Adversarial): tom hanks is a great actor. are you a fan?

Example 3
Dialog History: **System:** so do you have any pets at home? **User**: yeah i've got two **System:** cool what is it? **User**: it's a dog
BERT Ranker (RSD Train): wow! that's great! i like dogs! what is its breed?
BERT Ranker (Synthetic Retrieval): cool! what kind of pet?

Figure 4: Example predictions of BERT Ranker (RSD Train) outperforming BERT Ranker trained on Synthetic Datasets. Due to privacy concerns, these example dialogs are from an internal author.

Example 1
Dialog History: **System:** How about we talk about movies, news or music. So, what topic would you like to talk next ? **User**: tell me about
Model selected response: ok. i can fetch ratings from rotten tomatoes, check for fresh movies in the theatres, tell you movie plot and reviews, and more.
A correct response: i didnt hear you clearly, could you please repeat?
Issue: User utterance 'tell me about' was incomplete due to ASR error. The model did not recognize the error and focused on the dialog topic instead which was movies.

Example 2
Dialog History: **System:** so, tell me, what was the last movie you liked? **User**: i don't remember
Model selected response: i don't know this movie. space odyssey 2001 is great, in my opinion. it is a long-standing classic, one of the first films about artificial intelligence. tell me, what movie genres do you prefer?
A correct response: i have found some popular new movies: uncut gems and the rest of us. do you want to know details about uncut gems?
Issue: The model assumes the user fulfilled the system's question by providing a movie even though the user didn't.

Figure 5: Example predictions of BERT Ranker (RSD Train). Due to privacy concerns, these example dialogs are from an internal author.

5.3 Analysis

The advantage of creating negative examples via random or synthetic approaches is the ability to automatically increase the number of training examples. To further evaluate this, we vary the number of negative candidates in Synthetic Inter-Random, Intra-Random, and Adversarial, and report the corresponding MRR and Recall@1 scores, in Figure 3. We see that for our Synthetic Datasets increasing the number of negative candidates to a certain point improves performance for both MRR and Recall@1, after which the performance will degrade.

Increasing the number of negative candidates for (Synthetic Inter-Random) and (Synthetic Intra-Random) increases the likelihood of retrieving a candidate that is a false negative. This can bring noise and confusion to the model during training time. Increasing the size of the corpus could mitigate this issue; however, it can be expensive to collect a large enough dataset to see its benefits.[6] The advantage of (Synthetic Adv.) is the ability to create a large number of negative candidates without collecting more data; however, as seen in Figure 3 the decrease in MRR and Recall@1 when sampling more candidates may be due to false negatives and therefore still need to be manually verified.

[6]Large datasets such as Reddit are known to be noisy and could degrade performance.

5.4 Qualitative Examples

We provide examples of our BERT-Ranker models in Figure 4. In Example 1 both responses selected by the models acknowledge the user's artist preference; however, BERT-Ranker (Synthetic Inter-Random) chooses a response that repeats the question already answered by the user while BERT-Ranker (RSD Train) does not. In Example 2, BERT-Ranker (RSD Train) provides a more coherent response versus BERT-Ranker (Synthetic Adversarial) which has an abrupt topic change. In Example 3, BERT Ranker (Synthetic Retrieval) repeats the same question asked in the dialog history.

Figure 5 shows two typical erroneous examples. In the examples we also provide an explanation for the errors. It is worth pointing out that incorrect ASR output (word errors or end point detection errors such as the first example) is a source of errors to confuse our models. Gopalakrishnan et al. (2020) has observed similar issues for the task of response generation in speech-based dialog systems. Future work such as training on synthetic/actual ASR errors is needed to improve the robustness of models for such ASR issues.

5.5 Limitations

Our evaluation is done on a dialog dataset that contains a limited number of responders and only GPT2 is used as a neural response generation model. Synthetically created examples may perform better on datasets with a wider variety of neural response generation models. Future work would involve collecting response selection data annotated with a wider variety of responders.

6 Conclusion

In this work, we have curated a new dataset for response selection, which contains multiple positive responses and human verified hard negatives. We conducted a comprehensive evaluation of SOTA response selection models and various techniques to construct negative candidates to demonstrate the benefit of the dataset. Even though RSD requires manual annotation we see that training on our dataset greatly outperforms methods that use only one positive example and generate adversarial negative candidates.

7 Ethics and Broader Impact

Our work involves re-ranking responses from a dialog system. We acknowledge that we are using data from real users who have not been paid for these interactions. We also acknowledge there may be biases in the demographics of the user population.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501.

Raefer Gabriel, Yang Liu, Anna Gottardi, Mihail Eric, Anju Khatri, Anjali Chadha, Qinlang Chen, Behnam Hedayatnia, Pankaj Rajan, Ali Binici, et al. Further advances in open domain dialog systems in the third alexa prize socialbot grand challenge.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and William B Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 386–395.

Karthik Gopalakrishnan, Behnam Hedayatnia, Longshaokan Wang, Yang Liu, and Dilek Hakkani-Tür. 2020. Are neural open-domain dialog systems robust to speech recognition errors in the dialog history? an empirical study.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Prakhar Gupta, Shikib Mehri, Tiancheng Zhao, Amy Pavel, Maxine Eskenazi, and Jeffrey P Bigham. 2019. Investigating evaluation of open-domain dialogue systems with human generated multiple references. *arXiv preprint arXiv:1907.10568*.

Prakhar Gupta, Yulia Tsvetkov, and Jeffrey P Bigham. 2021. Synthesizing adversarial negative responses for robust response ranking and evaluation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3867–3883.

Janghoon Han, Taesuk Hong, Byoungjae Kim, Youngjoong Ko, and Jungyun Seo. 2021. Fine-grained post-training for improving retrieval-based dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1549–1558, Online. Association for Computational Linguistics.

Behnam Hedayatnia, Karthik Gopalakrishnan, Seokhwan Kim, Yang Liu, Mihail Eric, and Dilek Hakkani-Tur. 2020. Policy-driven neural response generation for knowledge-grounded dialog systems. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 412–421.

Huda Khayrallah and João Sedoc. 2020. Smrter chatbots: Improving non-task-oriented dialog with simulated multi-reference training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4489–4505.

Gary King and Langche Zeng. 2001. Logistic regression in rare events data. *Political analysis*, 9(2):137–163.

Jakub Konrád, Jan Pichl, Petr Marek, Petr Lorenc, Van Duy Ta, Ondřej Kobza, Lenka Hỳlová, and Jan Šedivỳ. 2021. Alquist 4.0: Towards social intelligence using generative models and dialogue personalization. *arXiv preprint arXiv:2109.07968*.

Jia Li, Chongyang Tao, Wei Wu, Yansong Feng, Dongyan Zhao, and Rui Yan. 2019. Sampling matters! an empirical study of negative sampling strategies for learning of matching models in retrieval-based dialogue systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1291–1296, Hong Kong, China. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Stefano Mezza, Alessandra Cervone, Evgeny Stepanov, Giuliano Tortoreto, and Giuseppe Riccardi. 2018. Iso-standard domain-independent dialogue act tagging for conversational agents. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3539–3551.

Masahiro Mizukami, Hideaki Kizuki, Toshio Nomura, Graham Neubig, Koichiro Yoshino, Sakriani Sakti,

Tomoki Toda, and Satoshi Nakamura. 2015. Adaptive selection from multiple response candidates in example-based dialogue. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 784–790. IEEE.

Ioannis Papaioannou, Amanda Cercas Curry, Jose L Part, Igor Shalyminov, Xinnuo Xu, Yanchao Yu, Ondrej Dušek, Verena Rieser, and Oliver Lemon. 2017. Alana: Social dialogue using an ensemble model and a ranker trained on user feedback.

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Ashwin Ram, Rohit Prasad, Chandra Khatri, Anu Venkatesh, Raefer Gabriel, Qing Liu, Jeff Nunn, Behnam Hedayatnia, Ming Cheng, Ashish Nagar, et al. 2018. Conversational ai: The science behind the alexa prize. *arXiv preprint arXiv:1801.03604*.

Sougata Saha, Souvik Das, Elizabeth Soper, Erin Pacquetet, and Rohini K Srihari. 2021. Proto: A neural cocktail for generating appealing conversations. *arXiv preprint arXiv:2109.02513*.

Ananya B Sai, Akash Kumar Mohankumar, Siddhartha Arora, and Mitesh M Khapra. 2020. Improving dialog evaluation with a multi-reference adversarial dataset and large scale pretraining. *Transactions of the Association for Computational Linguistics*, 8:810–827.

Abigail See and Christopher Manning. 2021a. Understanding and predicting user dissatisfaction in a neural generative chatbot. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–12, Singapore and Online. Association for Computational Linguistics.

Abigail See and Christopher D Manning. 2021b. Understanding and predicting user dissatisfaction in a neural generative chatbot. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–12.

Iulian V Serban, Chinnadhurai Sankar, Mathieu Germain, Saizheng Zhang, Zhouhan Lin, Sandeep Subramanian, Taesup Kim, Michael Pieper, Sarath Chandar, Nan Rosemary Ke, et al. 2017. A deep reinforcement learning chatbot. *arXiv preprint arXiv:1709.02349*.

Igor Shalyminov, Ondřej Dušek, and Oliver Lemon. 2018. Neural response ranking for social conversation: A data-efficient approach. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 1–8.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and Heuiseok Lim. 2020. An effective domain adaptive post-training method for bert in response selection. In *Interspeech*.

Taesun Whang, Dongyub Lee, Dongsuk Oh, Chanhee Lee, Kijong Han, Dong-hun Lee, and Saebyeok Lee. 2021. Do response selection models really know what's next? utterance manipulation strategies for multi-turn response selection.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. Transfertransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint arXiv:1901.08149*.

Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505.

Ruijian Xu, Chongyang Tao, Daxin Jiang, Xueliang Zhao, Dongyan Zhao, and Rui Yan. 2020. Learning an effective context-response matching model with self-supervised tasks for retrieval-based dialogues. *arXiv preprint arXiv:2009.06265*.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9604–9611.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. DialoGPT: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, and Hai Zhao. 2018b. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 3740–3752.

Zhuosheng Zhang and Hai Zhao. 2021. Structural pre-training for dialogue comprehension. *arXiv preprint arXiv:2105.10956*.

Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.

Naitian Zhou and David Jurgens. 2020. Condolences and empathy in online communities. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 609–626.

A Response Selection Model Training Details

All our BERT-base (Devlin et al., 2019) models are trained with a batch size of 32 on 1 NVIDIA V100 GPU with 16GB memory. We use the Adam optimizer with a learning rate of 1e-5 and the model is trained for 2 epochs. We use a sequence length of 256 tokens. To deal with the label imbalance, we compute a weighted loss where the loss for a positive candidate is up-weighted by a factor of α and the loss for a negative candidate is down-weighted by a factor of β. We follow (King and Zeng, 2001) and compute α by taking the sum of the number of positive and negative candidates and divide by the number of labels times the number of positive candidates. The same is done for β but we divide by the number of negative candidates instead. In our experiments $\alpha = 5.35$ and $\beta = 0.55$.

For the DialogRPT-human vs ranker model, we train with a batch size of 4 on 8 NVIDIA V100 GPUs with 16GB memory each. We use the Adam optimizer with a learning rate of 3e-5. We use a sequence length of 50 tokens and the model is trained for 3 epochs.

B NRG Training Details

We train all our NRG models on the RUI dataset described in Section 3.2. This dataset is split into a 90/10/10 train, valid, test split. All of our models are initialized with GPT2 (Radford et al., 2019) based models and were trained with a batch size of 2 on 8 NVIDIA A100 GPUS with 32GB memory each. We use the Adam optimizer and a learning rate of 6.25e-5. Each model is trained for 3 epochs and we finetune both the Language Modeling Head and Multiple Choice Head of GPT2 in a Transfer-Transfo fashion (Wolf et al., 2019). The Multiple Choice Head is finetuned with 1 randomly selected negative candidate. We leverage the HuggingFace's transformers library for all our models.[1] Detailed descriptions of our NRG variants are provided as below.

NRG Responder: Is a GPT2-XL model where the input is the dialog context which is truncated to 64 tokens.

NRG Responder GPT2-medium: Is a GPT2-medium model where the input is the dialog context which is truncated to 64 tokens.

NRG Responder grounded on knowledge: Is a GPT2-XL model where the dialog context is truncated to 256 tokens and a single knowledge sentence is truncated to 32 tokens. The dialog context and knowledge sentence are concatenated together to be used as input into the model.

NRG Responder grounded on dialog acts (DA): Is a GPT2-XL model where the dialog context is truncated to 64 tokens and each dialog act has it's own embedding that is randomly initialized and updated during finetuning. The dialog context and DA are concatenated together to be used as input into the model. When training this model we automatically label the RUI dataset with a dialog act tagger [2] and use those DAs as the ground truth. The DA labels used are from (Mezza et al., 2018) e.g. Feedback, Yes-No question, Statement.

During inference, a sequence of dialog acts are determined using a rule-based dialog policy which are used as input into the model to control the generated response. For example, a Yes-No question dialog act will cause the model response to generate a question (Hedayatnia et al., 2020).

NRG Responder grounded on sentiment: Is a GPT2-XL model where the dialog context is truncated to 64 tokens. There is an embedding representing negative sentiment that is randomly initialized and updated during finetuning. The dialog context and negative sentiment are concatenated together to be used as input into the model to control the generated response. This controllability allows the model is able to generate a sympathetic response when the user expresses negative sentiment. When training such a model, we automatically label the RUI dataset with an off the shelf sentiment classifier (Zhou and Jurgens, 2020) and use those sentiment tags as the ground truth.

C Response Selection Annotation Details

Our annotation framework is shown in Figure C.1. A human annotator is shown a dialog context and a set of response candidates are shown below. The annotator can then check off however many responses they deem as appropriate with respect to the dialog context. All responses not selected are considered inappropriate.

[1] https://github.com/huggingface/transformers

[2] We annotate a subset of the RUI dataset for dialog acts and train an RNN model on these annotations

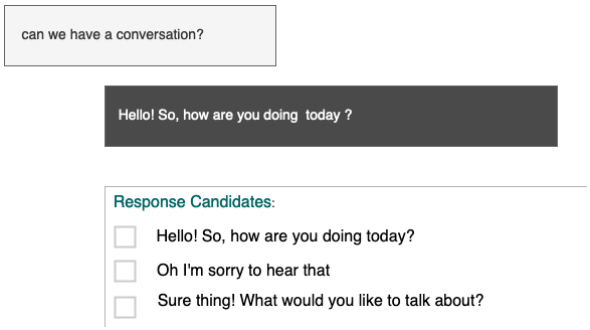


Figure C.1: Annotation framework to collect RSD Train/Test. Due to privacy concerns, this example dialog is from an internal author.

D Evaluation Metrics for Synthetic Datasets

In Figures D.2, D.3 and D.4 we show all the metrics for our BERT Ranker model trained on each of our Synthetic Datasets with different number of sampled negative candidates. We see for all metrics as the number of negative candidates increase results either degrade or taper off.

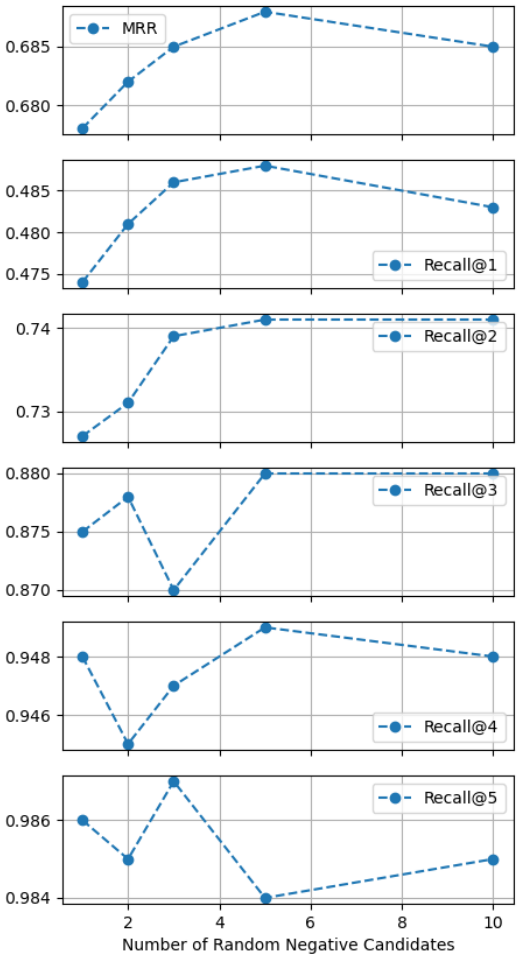


Figure D.2: Increasing the number of randomly sampled negative candidates in the Synthetic Inter-Random Training set.

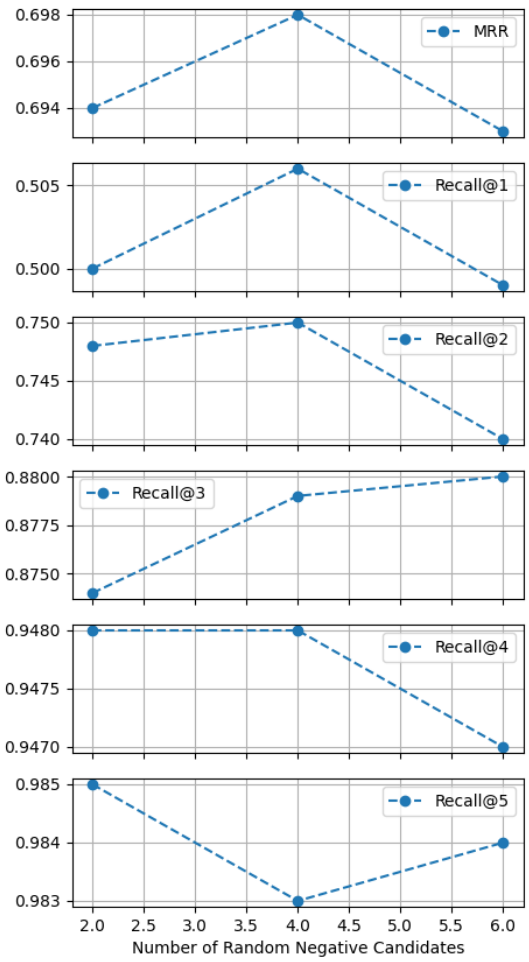


Figure D.3: Increasing the number of randomly sampled negative candidates in the Synthetic Intra-Random Training set.

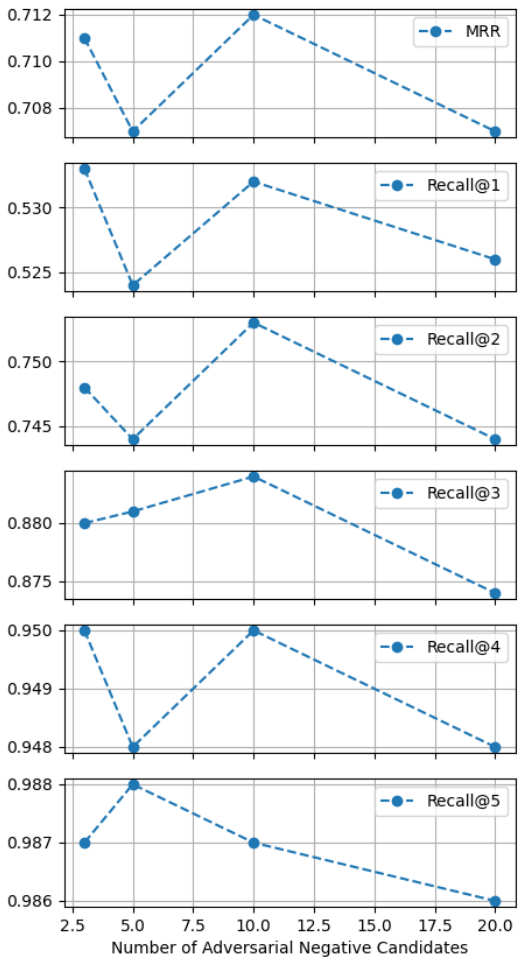


Figure D.4: Increasing the number of randomly sampled negative candidates in the Synthetic Adversarial Training set.

Inferring Ranked Dialog Flows from Human-to-Human Conversations

Javier Miguel Sastre Martínez, Aisling Nugent
Accenture The Dock, R&D Global Innovation Center,
7 Hanover Quay, Grand Canal Dock, Dublin, Ireland
{j.sastre.martinez, a.nugent}@accenture.com

Abstract

We present a novel technique to infer ranked dialog flows from human-to-human conversations that can be used as an initial conversation design or to analyze the complexities of the conversations in a call center. This technique aims to identify, for a given service, the most common sequences of questions and responses from the human agent. Multiple dialog flows for different ranges of top paths can be produced so they can be reviewed in rank order and be refined in successive iterations until additional flows have the desired level of detail. The system ingests historical conversations and efficiently condenses them into a weighted deterministic finite-state automaton, which is then used to export dialog flow designs that can be readily used by conversational agents. A proof-of-concept experiment was conducted with the MultiWoz data set, a sample output is presented and future directions are outlined.

1 Introduction

Virtual assistants are an attractive solution to customer service automation. While their language understanding capabilities and general knowledge of the world is limited in comparison with human agents, they can provide relatively simple services to an unlimited number of concurrent customers when coupled with cloud technologies. Additionally, they ensure an homogeneous experience, according to their programming. It is common practice to program a virtual assistant to fall back to a human agent whenever it detects it cannot provide a service, combining the strengths of both human and machine. Another usage of virtual assistants is to suggest to human agents a list of potential answers during a conversation with a customer, providing the agent potential useful information from previous interactions with the customer or from the customer profile, but letting the human decide the final answer.

Nowadays there exists a wide range of platforms for implementing virtual assistants, such as Google DialogFlow,[1] Amazon Lex,[2] Microsoft Bot Framework,[3] and RASA.[4] However, implementing a virtual assistant or extending it to support new services is not a trivial task. For the case of new services, one has to imagine how the conversations for that given service will be, or run a Wizard of Oz experiment with potential customers to gather examples of conversations. Once a conversational agent is deployed, it is often necessary to review its performance and adapt it to the actual conversations. For the case of services that are already being provided by human agents (e.g. in a call center), it is possible to review the conversation recordings in order to design a virtual assistant that will be better suited when first deployed. However, manually reviewing the call recordings can be time consuming.

In this paper we propose a technique to extract the most common workflows or dialog flows human agents follow when providing a specific service, once the calls are segregated by service.[5] The types of agent questions and responses are first identified and labeled (e.g."Where are you going?" → "Destination request"), for which we use proprietary software. Once the dialog utterances are replaced by the labels, hundreds of conversation paths can be condensed and ranked in seconds as a weighted finite-state automaton. Different ranges of best paths in the automaton can then be exported as a succession of manageable-size dialog flows for their manual review (examples in the supplementary material). The conversational designer can then review them in rank order and decide when to stop, taking into account the added value of each successive dialog flow and the time available.

[1] https://cloud.google.com/dialogflow
[2] https://aws.amazon.com/lex/
[3] https://dev.botframework.com/
[4] https://rasa.com/
[5] A potential approach to segregation by service is discussed in Chatterjee and Sengupta (2020)

Proceedings of the SIGdial 2022 Conference, pages 312–324
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

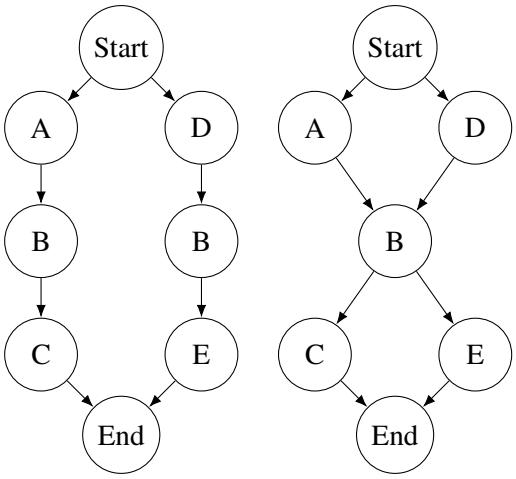


Figure 1: Example of 2 dialogs ABC and DBE (left) leading to overgeneration of sequences ABE and DBC (right) when only taking into account consecutive sequences of 2 dialog phases

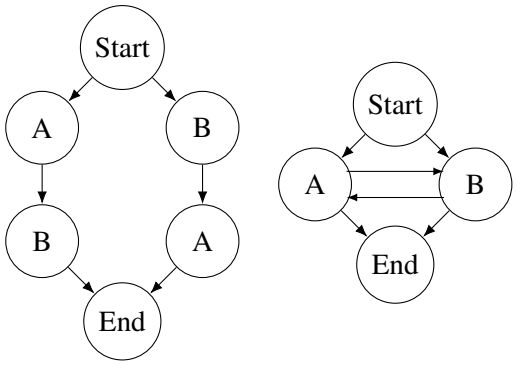


Figure 2: Example of 2 dialogs (left) leading to a loop (right) when only taking into account consecutive sequences of 2 dialog phases

2 Related Work

Bouraoui et al. (2019) present Graph2Bots, a tool that also aims to assist conversational agent designers. Similarly to us, they first identify types of utterances or dialog turns, which they call dialog phases. Then they build a graph with all possible dialog phases as nodes, and all possible transitions between consecutive dialog phases in the dialogs. Frequencies of dialog phases and transitions can then be used in order to filter out less frequent portions of the graph. We have also experimented with this kind of dialog phase graph and found several inconveniences we aim to overcome, namely

1. big convoluted graphs that, although they can be filtered, they are not partitioned so one can examine successive and manageable subsets of paths, one subset at a time,

2. the resulting graph represents concatenations of consecutive subsequences of 2 dialog phases from multiple dialogs, resulting in paths that do not actually exist in the dataset and produce confusion (Figure 1), and

3. the overgeneration of paths results in loops (Figure 2), which prevent the dialog flows from being loaded into conversational agent platforms as initial designs.

Qiu et al. (2020) propose an unsupervised approach to dialog structure inference based on a variational recurrent neural network with a structured attention layer that supports both 1 to 1 and multiparty conversations. However the reported times to train these models are in the order of hours, which in our use case would be impractical.

Zhai and Williams (2014) and Paul (2012) combine Hidden Markov Models and topic modeling to model the dialog structures as conversation states with probabilities to shift to other states, where each state models the potential language or topics in that state.

3 Rationale

This work builds on top of the output of a proprietary suite of tools for the analysis of call center conversations. This output comprises a set of dialog transcripts segregated by intent (e.g. booking a restaurant), where the utterances have been labeled by speaker role (agent/customer), classified into question/response/other, and then grouped into clusters of semantically equivalent questions/responses.[6] For each group of questions or responses, a canonical form of the question or response is provided to serve as the normalized version, analogous to the dialog phases in Bouraoui et al. (2019). Our goals are:

1. to find the most frequent sequences of questions and responses human agents follow, and

2. to compile them into a succession of dialog flows containing ranges of top-ranked sequences so that a conversational designer can visualize any number of them, starting from the highest ranked ones.

Also, as additional requirements,

[6]Other utterance types (e.g. greetings) are ignored

1. the paths in these dialog flows should come from actual dialogs and not be concatenations of subsequences from different dialogs (e.g. Figures 1 and 2), in order to avoid confusion and loops,

2. the size of individual dialog flows (number of paths) should be limited by means of a parameter, and

3. the dialog flows should also include some examples of potential customer utterances that may appear before and after each agent question or response so that one can determine the triggers of specific questions and responses as well as potential customer responses.

We do not intend to determine exact types of customer utterances but to provide a variety of examples since customer utterances tend to be more varied than agent utterances: whereas customers may request a service just a few times and may have no prior knowledge of the service protocols, agents deliver the same services multiple times to multiple customers and must adhere to established protocols and regulations.

4 Methodology

Overall, our proposed approach consists of 6 main steps:

1. building a non-deterministic finite-state automaton (NFA) representing all possible sequences of normalized agent questions and responses,

2. minimizing the NFA in order to obtain an equivalent but compact deterministic finite-state automaton (DFA),

3. annotating the DFA with question/response frequencies as well as with customer utterances

4. ranking the DFA paths and transitions and pruning it to a desired number of paths

5. selecting a maximum number of customer utterance examples before and after each agent utterance, discarding the rest, and

6. exporting consecutive ranges of ranked paths into separate dialog flows for their manual review.

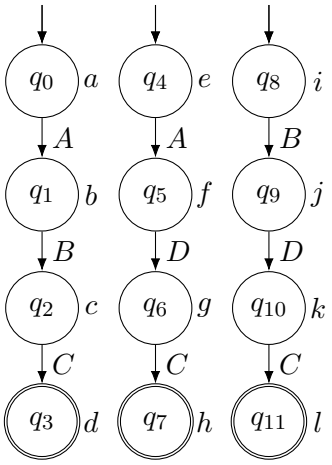


Figure 3: Example of NFA representing 3 dialogs $aAbBcCd, eAfDgCh$ and $iBjDkCl$

4.1 Building the NFA

For simplicity, let A, B, C and D be types of either agent questions or responses (their normalized versions or dialog phases). Let $a, b, \ldots l$ be specific examples of customer utterances (non-normalized). We build an NFA as depicted in Figure 3, with a linear sequence of states (nodes) and transitions (edges) for each dialog, where transitions are annotated with the normalized agent utterances and states with the customer utterance examples. Note we only consider agent questions and responses, and other kinds of agent utterances are simply ignored (e.g. greetings). Consecutive sequences of customer utterances between 2 agent utterances are simply concatenated and treated as a single utterance. Consecutive sequences of agent utterances with no customer utterances in between result in a state that is annotated with no customer utterance (a state may have no customer utterance).

Formally, an NFA is defined as a 5-tuple $(Q, \Sigma, \delta, Q_I, F)$ with

- $Q = \{q_0, q_1, \ldots, q_{|Q|-1}\}$, as a finite set of states,
- $\Sigma = \{\sigma_0, \sigma_1, \ldots, \sigma_{|\Sigma|-1}\}$, as an either finite or potentially infinite input alphabet (normalized agent utterances in our case),
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ as a finite and partial transition function where $\varepsilon \notin \Sigma$ is the empty symbol and $\mathcal{P}(\cdot)$ represents the set of all subsets of a given set,
- $Q_I \subseteq Q$ as the set of initial states (represented as nodes pointed by an arrow coming from

nowhere), and

- $F \subseteq Q$ as the set of final states (represented as double-circled nodes).

A path in the automaton is an alternation of states and input symbols $q_i, \sigma_i, q_{i+1}, \sigma_{i+1}, \ldots$ starting and ending with a state, where for every subsequence $q_j, \sigma_j, q_j + 1$ there is a transition $\delta(q_j, \sigma) = q_{j+1}$. We say an automaton recognizes, represents or accepts an input sequence $\sigma_i \ldots \sigma_{i+n}$ iff there exists at least one path from an initial state to a final state with the same sequence of input symbols. We say an automaton is not deterministic iff it contains at least 2 paths starting from an initial state and labeled with the same sequence of input symbols (multiple states can be reached by consuming the same input sequence). Note having more than one initial state is sufficient for being non-deterministic.

We define the partial map $\zeta_c : Q \to \Gamma$ of states to customer utterances (Γ being the set of all customer utterances) to capture the customer utterance that may appear between 2 agent utterances, if any.

4.2 Minimizing the NFA

Minimizing an NFA results in an equivalent deterministic finite-state automaton (DFA) that represents the exact same set of input sequences but with a minimum set of states (see Figure 4). While this does not necessarily imply that the resulting automaton will have less transitions, this is usually the case for the NFAs that we build. Note for the sake of minimization, customer utterances are ignored (we only care about producing the same sequences of agent questions and responses). Formally, we define a DFA as a 5-tuple $(Q, \Sigma, \delta, q_I, F)$, where each element is defined in the same manner than for NFAs except for

- q_I, which is a unique initial state instead of a set of possible states, and
- $\delta : (Q \times \Sigma) \to Q$, which does not allow for empty symbols or more than one target state for the same source state and input symbol.

NFA minimization can be achieved by reversing the automaton, determinizing it, reversing it again and determinizing it a second time (van de Snepscheut, 1985). Reversing an automaton can be achieved by reversing the transitions, making initial states final, and final states initial. Since the NFAs we produce do not use empty input symbols, we can use a simpler algorithm for determinizing them. Let A be one of these NFAs, Algorithm 1 (in the appendix) traverses all paths in A starting from its initial states, generating a DFA A' that contains a single state for each set of states that can be reached by consuming the same input sequence, and adding the corresponding transitions between the states in A'. It builds a map ζ_m of sets of states in A to states in A' to keep track of these correspondences and to avoid generating more than one state in A' for the same set of states in A. The algorithm starts by creating a single initial state q_I in A' corresponding to the set of initial states Q_I in A, and places the pair (Q_I, q_I) in a queue E of states to explore. As long as E is not empty, the next pair (Q_s, r_s) is dequeued and Algorithm 2 (in the appendix) is used to explore all the transitions coming from any state in Q_s, returning a map ζ_t of input symbols σ to sets of target states Q_t that can be reached from any state in Q_s by consuming σ. For each σ and Q_t, the corresponding state r_t in A' is either created or retrieved from ζ_m if already existed, and transition $\delta'(r_s, \sigma) = r_t$ is added to A'. Each time a state r_t is created for a given set of states Q_t, r_t is made final iff there is at least one final state in Q_t. Finally, whenever a new r_t is to be created due to the lack of a map $\zeta_m(Q_t)$, the map is added and (Q_t, r_t) is enqueued for further exploration of A.

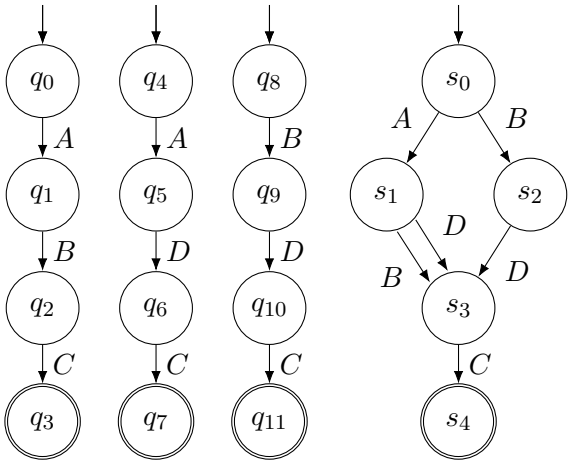


Figure 4: Example of NFA representing 3 dialogs (left) and DFA resulting from the NFA minimization (right)

4.3 Annotating the DFA

Let A be an NFA and A_{min} the resulting DFA upon minimization, since both machines are equivalent they recognize the exact same sequences. In the same way that during minimization we generate

states of a DFA that correspond to sets of states in an NFA, there is a correspondence between states in A_{min} and states in A, as well as between transitions in A_{min} and transitions in A.

Given a map ζ_c of states in A to customer utterances (1 or none per state), Algorithm 3 (in the appendix) annotates the states in A_{min} with the sets of all customer utterances of the corresponding states in A (map ζ'_c), and annotates the transitions in A_{min} with the count of all equivalent transitions in A (map ζ'_f). An example is given in Figure 5. The algorithm also requires a topological sort of A_{min} as an input,[7] which can be computed with Kahn's (1962) algorithm. Algorithm 3 (in the appendix) explores both A and A_{min} synchronously, while computing the map ζ_m^{-1} of states in A_{min} to states in A. It starts by mapping the initial state of A_{min} to the set of initial states in A. Then explores the states of A_{min} by following the provided topological sort. For each state s_s in the sort, it retrieves the corresponding set of states $\zeta_m^{-1}(s_s) = Q_s$, and annotates s_s with the union of customer utterances in Q_s. Then for each transition $\delta'(\sigma, s_s) = s_t$ in A_{min} finds all the corresponding transitions $\delta(\sigma, q_s) = q_t$ in A, adding all the states q_t found to the mapping $\zeta_m^{-1}(s_t)$, and incrementing the count of transitions $\zeta'_f(s_s, \sigma, s_t)$ for each equivalent transition found in A. The topological sort is needed so that when exploring a next state s_s in the sort, we are sure the map $\zeta_m^{-1}(s_s)$ contains every possible corresponding state q_s in A, which will be the case since A and A_{min} are equivalent.

Apart from transition counts or frequencies, transitions of A_{min} can also be annotated with probabilities by normalizing the frequencies: for each set of transitions outgoing from the same source state, we compute the sum of frequencies of the transitions in the set, then divide the frequencies of these transitions by the sum. Log-probabilities can also be added in order to optimize the computation of top-scoring paths in the next section. A path score is the aggregation of the transition weights in the path, let it be the sum of frequencies, the product of probabilities, or the sum of log-probabilities.

4.4 Ranking and pruning

Given an annotated DFA A and a maximum desired number k of paths to keep (or carve), we use a Viterbi-like (1967) algorithm to efficiently compute the top-scoring paths, rank them (from 1^{st} to k^{th}) and annotate the transitions in A with the set of ranks of top paths they belong to. Transition rank annotations are used in the export step to generate the dialog flows for desired ranges of best paths. States and transitions that do not belong to any top-k path are removed in order to limit the execution time of the algorithm. Whereas this also limits the ranges of best paths that it will later be possible to export, in practice this limit can be much higher than the number of paths a conversational designer would deem necessary (e.g. 500), while keeping the execution time in the order of seconds. The algorithm is divided in 4 parts, which we detail in the following subsections: DFA preparation, forward propagation of weights, backward propagation of ranks, and DFA clean up. The first 3 parts also make use of a topological sort that is to be previously computed; the same topological sort used for the DFA annotation can be reused here. We use DFA in Figure 6 as an example. Note customer utterances are omitted since they are not relevant for the sake of ranking and pruning (the algorithms ignore map ζ'_c).

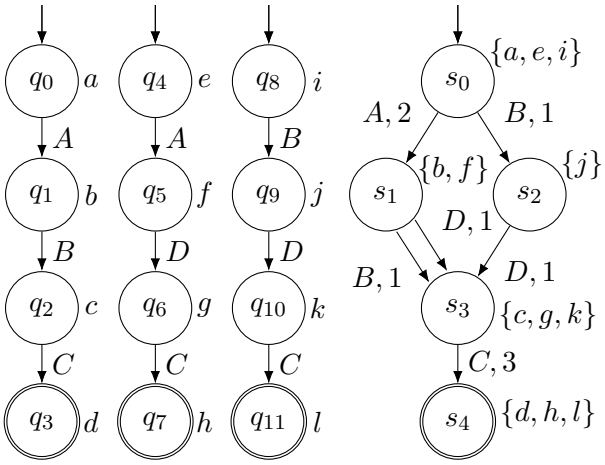


Figure 5: Example of NFA representing 3 dialogs (left) and equivalent DFA after state and transition annotation (right)

4.4.1 DFA preparation

The ranking and pruning algorithm computes the best paths between an initial and a final state of a DFA A. Whereas a DFA can only have a single initial state, it may have more than one final state. In order to take into account all possible paths in the DFA, we modify A as illustrated in Figure 7 so it contains a single and new final state s_f, with ε-

[7] An ordering of all the states in A_{min} such that, for every transition in A_{min}, target states always come after source states in the ordering. This is the same problem as finding an ordering in a dependency graph.

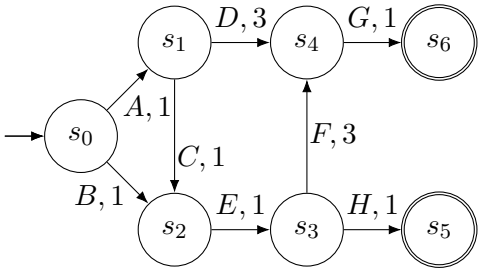


Figure 6: DFA with top-3 sequences ACEFG (1+1+1+3+1=7), BEFG (1+1+3+1=6) and ADG (1+3+1=5)

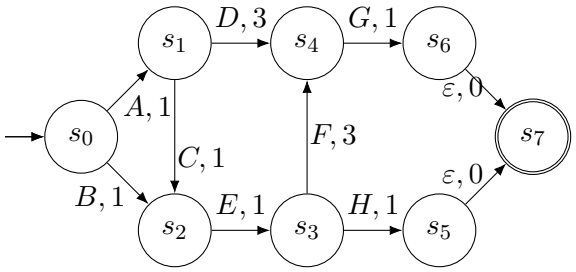


Figure 7: DFA after carving preparation

transitions arriving to it from each former final state and annotated with neutral weights (0 for frequencies and log-probabilities or 1 for probabilities). Strictly speaking, adding ε-transitions to A make it non-deterministic, however they will be removed during the clean up of A. Finally, the topological sort of A is to be updated by appending s_f at the end. In Figure 7, it ends up being $s_0, s_1, \ldots, s_7$.

4.4.2 Forward propagation of weights

For each state s_t in a carving-prepared DFA A, Algorithm 4 (in the appendix) computes the list L_t of k best possible aggregated weights that can be produced by reaching s_t from the initial state, and annotates s_t with this list (map $\zeta_L(s_t) = L_t$). An example of the computed lists is given in Figure 8 (lists above or below the states). Each element of L_t is a triplet (w, σ, s_s), with w being a top aggregated weight, and (σ, s_s) the symbol and source state of the previous transition that allowed for that best weight (transition $\delta(\sigma, s_s) = s_t$). The algorithm starts by initializing the lists L_t of all the states as empty lists. Then initializes $\zeta_L(q_I)$ with triplet $(w_{init}, \varepsilon, \perp)$, a initial aggregated weight (0 for frequencies and log probabilities, 1 for probabilities), and a non-transition (there is no transition before q_I). For each state s_s in the provided topological sort, except the last state s_f added during carving preparation, the algorithm propagates the corresponding top weights in $\zeta_L(s_s)$ towards the

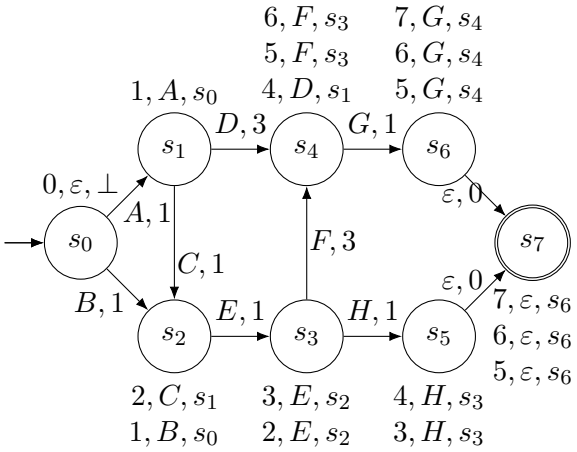


Figure 8: DFA after forward propagation

lists L_t of the corresponding target states s_t. Given a list L_s of $n \leq k$ elements, the n weights are combined with the weight of each transition from s_s, and the resulting aggregated weight is added to the list L_t of the corresponding target state s_t along with the corresponding transition symbol and source state s_s. Lists of top weights are sorted lists of at most k triplets, so when a list overflows the excess can be easily removed from its end. Thanks to the topological sort, whenever propagating the top weights of a state s_s we make sure all possible paths that reach s_s from q_I have been explored, and the list contains the top weights only (excess of weights will have been removed).

4.4.3 Backward propagation of ranks

Once the lists of top weights and last transitions have been computed, we can proceed to rank the final top weights in $\zeta_L(s_f)$ and propagate these ranks backwards, following the last transitions in the corresponding triplets of best weights. Algorithm 5 (in the appendix) starts by creating a list of sets of ranks for state s_f (map ζ_{SR} standing for *state ranks*), one set of ranks per triplet in $\zeta_L(s_f)$. The first set of ranks is $\{1\}$ (first rank), the second is $\{2\}$ (second rank), and so forth (see list below s_7 in Figure 9; we replaced top weights with rank sets to save space). Then these ranks are propagated backwards by following a reverse of the topological sort, excepting the initial state. Given a state s_t in the topological sort, the algorithm first computes a map ζ_{BR} (backwards ranks) of backwards transitions to the list of all possible sets of ranks in $\zeta_L(s_t)$. For instance, in Figure 9 the 3 top backwards transitions of s_f are the same, so ζ_{BR} contains in this case a single map $\zeta_{BR}(\varepsilon, s_6) = [\{1\}, \{2\}, \{3\}]$. For each

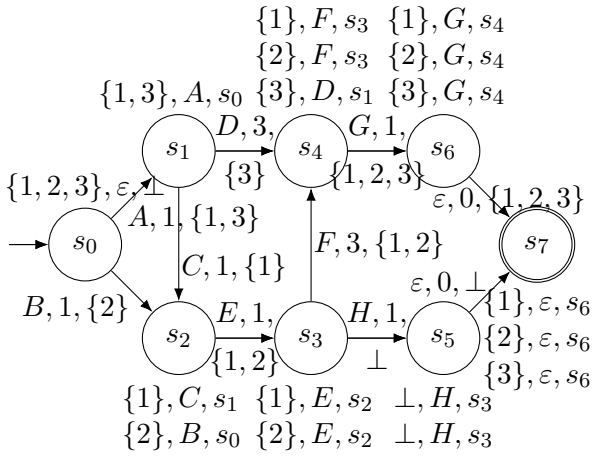


Figure 9: DFA after backward propagation

map $\zeta_{BR}(\sigma, s_s) = BR$, transition $\delta(s_s, \sigma) = s_t$ is annotated with the union of all the sets of ranks in BR (e.g. in Figure 9, transition $\delta(s_6, \varepsilon) = s_7$ gets ranks $\{1, 2, 3\}$. Map ζ_{TR} is used to annotate the transition ranks. Furthermore, for each list of sets of ranks $\zeta_{BR}(\sigma, s_s) = [R_1, R_2, \ldots]$, the list is propagated backwards towards $\zeta_{SR}(s_s)$ by computing the pairwise union of sets of ranks of $\zeta_{SR}(s_s)$ with $\zeta_{BR}(\sigma, s_s)$. For instance, in Figure 9 list $[1, 2, 3]$ below s_7 gets propagated as is to the list above s_6, since s_7 is the only contributor of ranks for s_6. States that get ranks are part of the top k paths and are marked as useful (states to be keep during clean up). For instance, no ranks get propagated to s_5 (symbol $\bot$ represents null), hence it will not be marked and will be removed during clean up. Ranks $[\{1\}, \{2\}]$ above state s_4 correspond to transition $\delta(s_3, F) = s_4$, hence get propagated to state s_3 (ranks below s_3). However rank $[\{3\}]$ of s_4 for transition $\delta(s_1, D) = s_4$ gets propagated to ranks of state s_1. Rank $[\{1\}]$ of s_2 for transition $\delta(s_1, C)$ also gets propagated to s_1, resulting in ranks $[\{1, 3\}]$ (pairwise union of sets of ranks). Once the algorithm ends, the ranks $[\{1\}, \{2\}, \{3\}]$ of s_7 have travelled back through the top paths, annotating the corresponding transitions and states, and arriving to state s_0 as $[\{1, 2, 3\}]$.

4.4.4 DFA clean up

Algorithm 6 (in the appendix) undoes the changes done to the DFA during carving preparation, and deletes every unmarked state (e.g. s_5) and unranked transition (e.g. $\delta(s3, H) = s_5$). The lists of state top weights and ranks are no longer needed and can be discarded; we just need to keep map ζ_{TR} of transition ranks. Figure 10 illustrates the resulting automaton for our example. In order to avoid potential data corruption, the algorithm deletes the states and transitions in a proper order, starting with transitions in $\zeta_L(s_f)$; these are ranked transitions but are added during carving preparation. Then states s_s are scanned in topological sort except for s_f. For each s_s, transitions from s_s with no ranks are removed. Then s_s is removed if it's not marked. Note that by following a topological sort, all transitions incoming to and outgoing from an unmarked state are removed before removing the state. Finally, s_f is removed unconditionally without scanning it, since it has no outgoing transitions and it was added during carving preparation. We no longer need the topological sort, so it can be discarded.

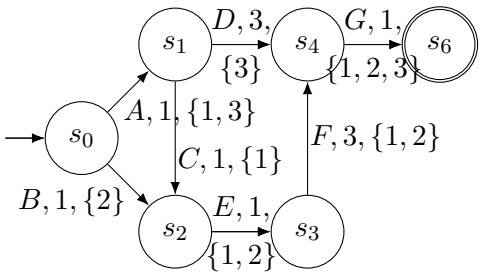


Figure 10: DFA after clean up

4.5 Selecting customer utterances

Due to the potential big number of customer utterances that might be annotated on the remaining DFA states, we want to select a limited number n of different examples per state and delete the rest so that the exported dialog flows are not overcrowded. For each set of customer utterances, we first compute the corresponding sentence embeddings (Cer et al., 2018; Reimers and Gurevych, 2019; Yang et al., 2020). Then we clusterize the sentences into groups of semantically similar ones using DBSCAN (Ester et al., 1996). We select the n biggest clusters and, for each one, we find the vector closest to the cluster centroids. Finally, we retrieve the sentences that correspond to those vectors, and delete all the rest.

4.6 Exporting dialog flows

Once the DFA is pruned, the transitions ranked, and the customer utterances filtered, generating a dialog flow for an arbitrary range of best paths is straightforward: we simply traverse the automaton starting from the initial state and following every transition that has at least one rank within the range,

until no more states are found. Transition rank sets ζ_{TR} are sorted data structures (e.g. sorted lists or binary trees) so one can efficiently evaluate whether the intersection of the set with the range of ranks is empty or not. As states and transitions are traversed, the corresponding nodes and edges of the dialog flow can be exported to the desired format, e.g. DOT (Gansner and North, 2000) in order to create dialog flow visualizations, or some format of a conversational agent platform.

5 Methodology extension

An inconvenience of the method described above is that all the customer utterances that may start a conversation get grouped together in the DFA initial state (e.g. utterances a, e, and i of state s_0 in Figure 5). We would like to split this group into potential utterances that may precede each first agent utterance, so that we can also determine what triggered each first agent utterance. This can be achieved by modifying the way in which the NFA is built, as illustrated in Figure 11: we simply duplicate the first transition of each individual dialog, leaving the new initial states with no customer utterances. Upon minimization, the new first agent utterances will only allow for grouping the first customer utterances that are followed by the same agent utterance. Upon exporting the dialog flows, these first agent utterances are simply to be ignored.

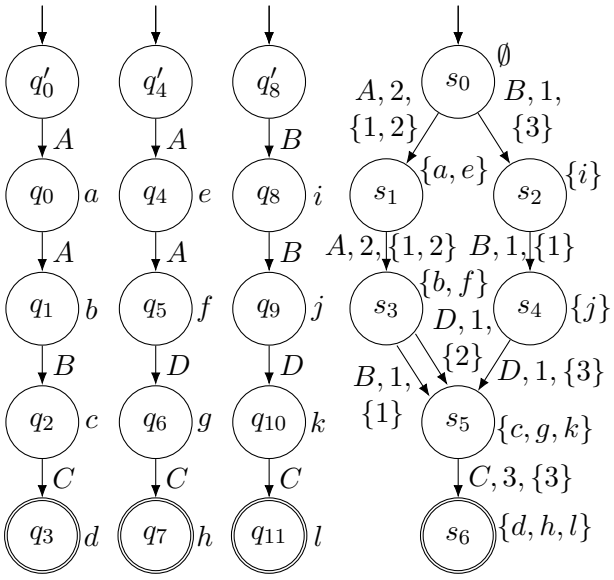


Figure 11: NFA with duplicated first transitions (left) and resulting NFA after minimization, annotation and ranking (right)

6 Results

We have tested this methodology with a sample of 492 restaurant booking dialogs from MultiWOZ (Han et al., 2020). On a MacBook Pro (2018), it took 4.3 seconds to run the extended method from NFA building (4083 states and 3591 transitions) to DFA ranking and pruning (1704 states, 2176 transitions) for a big enough k so all paths (488) were ranked and kept. Filtering the customer utterances took 36.8 additional seconds, though taking into account that this process included computing multilingual sentence embeddings (Yang et al., 2020) for all the customer utterances, it could be considerably reduced by using a GPU. Exporting a dialog flow of 50 paths into SVG with GraphViz (Gansner and North, 2000) took 2.5 seconds. Two flows are shown as supplementary material, and a wide range of flows has been provided as accompanying materials. Ranking criterion is frequency aggregation so longer paths are produced. For simplicity, only the top rank of each transition is shown. The process factors out prefixes and suffixes of agent utterance sequences, which to some extent allows for identifying the most common full sequences. The flows exactly reflect what is found in the data, which is what we initially intended.

7 Conclusion and future work

This paper presented a novel and efficient method for inferring ranked dialog flows from human-to-human conversations in seconds. This method converts the dialogs into summarised and digestible artefacts, in the form of weighted finite-state automata with ranked transitions. The method is intended to be used together with a semi-supervised iterative process of identification of types of agent utterances, hence the quick generation of the ranked dialog flows is a must.

Future work includes 1) splitting the customer bubbles across the entire dialog flows to have separate groups of examples of customer utterances before each agent bubble, 2) to identify dialog substructures such as subsequences of agent questions and responses that may appear in any order, so they can be replaced by a subautomaton call and allow for further path collapsing, and 3) to allow for a controlled amount of overgeneration/noise in the automaton that maximizes the number of collapsed paths (adding missing subsequences that allow for further minimization).

8 Acknowledgements

We thank Paul A. Walsh and the SIGDIAL 2022 reviewers for their feedback.

References

Jean-Leon Bouraoui, Sonia Le Meitour, Romain Carbou, Lina M. Rojas Barahona, and Vincent Lemaire. 2019. Graph2Bots, unsupervised assistance for designing chatbots. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 114–117, Stockholm, Sweden. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.

Ajay Chatterjee and Shubhashis Sengupta. 2020. Intent mining from past conversations for conversational agent. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4140–4152, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

Emden R. Gansner and Stephen C. North. 2000. An open graph visualization system and its applications to software engineering. *Softw. Pract. Exper.*, 30(11):1203—1233.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2020. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. *arXiv preprint arXiv:2010.05594*.

Arthur B. Kahn. 1962. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562.

Michael J. Paul. 2012. Mixed membership Markov models for unsupervised conversation modeling. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 94–104, Jeju Island, Korea. Association for Computational Linguistics.

Liang Qiu, Yizhou Zhao, Weiyan Shi, Yuan Liang, Feng Shi, Tao Yuan, Zhou Yu, and Song-Chun Zhu. 2020. Structured attention for unsupervised dialogue structure induction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1889–1899, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Jan L. A. van de Snepscheut. 1985. *Trace theory and VLSI design*, volume 200 of *Lecture Notes in Computer Science*. Springer-Verlag. PhD thesis, Eindhoven University of Technology.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online. Association for Computational Linguistics.

Ke Zhai and Jason D. Williams. 2014. Discovering latent structure in task-oriented dialogues. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–46, Baltimore, Maryland. Association for Computational Linguistics.

A Appendix

A.1 Algorithms

Algorithm 1 nfa_determinize(A)

Input: $A = (Q, \Sigma, \delta, Q_I, F)$, a NFA
Output: $A' = (Q', \Sigma, \delta', q_I, F')$, a DFA equivalent to A

initialize A' as a DFA with a single and initial state q_I and no final states or transitions
if $Q_I \cap F \neq \emptyset$ **then**
    $F' \leftarrow F' \cup \{q_I\}$
end if
$\zeta_m(Q_I) \leftarrow q_I$ ▷ state equivalence map
$E \leftarrow \{(Q_I, q_I)\}$ ▷ equivalent-pairs queue
while $E \neq \emptyset$ **do**
    $(Q_s, r_s) \leftarrow \text{dequeue}(E)$
    $\zeta_t \leftarrow \text{nfa\_recognize\_every\_symbol}(Q_s)$
    for each $(\sigma, Q_t) : \zeta_t(\sigma) = Q_t$ **do**
        $r_t \leftarrow \zeta_m(Q_t)$
        if $r_t = \perp$ **then**
            make new state $r_t \in Q'$
            if $Q_t \cap F \neq \emptyset$ **then**
                $F' = F' \cup \{r_t\}$
            end if
            $\zeta_m(Q_t) \leftarrow r_t$
            $E \leftarrow E \cup (Q_t, r_t)$
        end if
        $\delta'(r_s, \sigma) \leftarrow r_t$
    end for
end while

Algorithm 2 nfa_recognize_every_symbol(Q_s)

Input: Q_s, a source set of states
Output: $\zeta_t : \Sigma \rightarrow \mathcal{P}(Q)$, a map of input symbols to target sets of states such that $\zeta_t(\sigma) = \bigcup_{q_s \in Q_s} \delta(q_s, \sigma)$

initialize ζ_t as an empty map of $\Sigma \rightarrow \mathcal{P}(Q)$
for each $q_s \in Q_s$ **do**
    for each $(\sigma, q_t) : \delta(q_s, \sigma) = q_t$ **do**
        if $\zeta_t(\sigma) = \perp$ **then**
            $\zeta_t(\sigma) \leftarrow \emptyset$
        end if
        $\zeta_t(\sigma) \leftarrow \zeta_t(\sigma) \cup \{q_t\}$
    end for
end for

Algorithm 3 dfa_annotate(A, A_{min}, ζ_c)

Input: $A = (Q, \Sigma, \delta, Q_I, F)$, a NFA
$\zeta_c : Q \rightarrow C$, map of states in A to customer utterances
$A_{min} = (Q', \Sigma, \delta', q'_I, F')$, DFA result of minimizing A
$A_{min\_sort} : (Q \times Q \times \ldots)$, a topological sort of A_{min}
Output: $\zeta'_c : Q' \rightarrow \mathcal{P}(C)$, map of A_{min} states to sets of customer utterances
$\zeta'_f : (Q' \times \Sigma \times Q') \rightarrow \mathbb{N}_0$, map of A_{min} transitions to frequencies

$\zeta_m^{-1}(q_I) \leftarrow Q_I$ ▷ Inverse equivalent state map
for each $s_s \in A_{min\_sort}$ **do**
    $Q_s \leftarrow \zeta_m^{-1}(s_s)$
    $\zeta'_c(s_s) \leftarrow \bigcup_{q_s \in Q_s} \{\zeta_c(q_s)\}$
    for each $(\sigma, s_t) : \delta'(s_s, \sigma) = s_t)$ **do**
        if $\zeta_m^{-1}(s_t) = \perp$ **then**
            $\zeta_m^{-1}(s_t) \leftarrow \emptyset$
        end if
        $Q_t \leftarrow \bigcup_{q_s \in Q_s} \delta(q_s, \sigma)$
        $\zeta_m^{-1}(s_t) \leftarrow \zeta_m^{-1}(s_t) \cup Q_t$
        $\zeta'_f(s_s, \sigma, s_t) \leftarrow |Q_t|$
    end for
end for

Algorithm 4 dfa_carving_forward_prop(A, $A_{sort}, \varepsilon, \zeta_w, w_{init}, \bullet, \prec, k$)

Input: $A = (Q, \Sigma, \delta, q_I, F)$, carving-prep. DFA,
$A_{sort} : Q^{|Q|-1}$, a topological sort of A
ε, a special symbol not in Σ to denote the empty input
$\zeta_w : (Q \times \Sigma \cup \{\varepsilon\}) \times Q) \rightarrow W$, map of A transitions to weights
w_{init}, the initial weight
$\bullet$, the weight aggregation operator
$\prec$, the weight comparison operator
k, the number of paths to carve
Output: ζ_L, a map of states $s_t \in Q$ to sorted lists of triplets $(w, \sigma, s_s) \in (W \times (\Sigma \cup \{\varepsilon\}) \times Q)$, each representing a top-k best weight w produced by reaching s_t through a last transition $\delta(s_s, \sigma) = s_t$

for each $s \in Q$ **do**
    $\zeta_L(s) \leftarrow$ empty list
end for
append $(w_{init}, \varepsilon, \perp)$ to $\zeta_L(q_I)$

5: **for each** $s_s \in A_{sort}$ except last **do**
6: $L_s \leftarrow \zeta_L(s_s)$
7: **for each** $(\sigma, s_t) : \delta(s_s, \sigma) = s_t$ **do**
8: $w \leftarrow \zeta_w(s_s, \sigma, s_t)$
9: $L_t \leftarrow \zeta_L(s_t)$
10: **for each** $(w_s, \sigma', s_b) \in L_s$ **do**
11: $w_t \leftarrow w_s \bullet w$
12: insert (w_t, σ, s_s) in L_t maintaining $\prec$ weight order
13: **if** size of L_t > k **then**
14: remove last triplet from L_t
15: **end if**
16: **end for**
17: **end for**
18: **end for**

Algorithm 5 dfa_carving_backward_prop(A, A_{sort}, ζ_L)

Input: $A = (Q, \Sigma, \delta, q_I, F)$, carving-prep. DFA
$A_{sort} : Q^{|Q|-1}$, topological sort of A
ζ_L, map of states to top backwards trans.
Output: A with states to keep marked
ζ_{TR}, map of transitions (s_s, σ, s_t) in A to sets of ranks in $\mathcal{P}(N)$

1: $s_f \leftarrow$ last state in A_{sort}
2: $k' \leftarrow |\zeta_L(s_f)|$ ▷ number of top paths found
3: **for each** $i = 1 \ldots k'$ **do** ▷ init. s_f rank sets
4: $\zeta_{SR}(s_f)[i] \leftarrow \{i\}$
5: **end for**
6: **for each** $s_t \in \text{reverse}(A_{sort})$ except last **do**
7: $SR_t \leftarrow \zeta_{SR}(s_t)$
8: **if** $SR_t \neq \perp$ **then** ▷ no ranks for s_t
9: **continue** ▷ skip s_t rank propagation
10: **end if**
11: mark s_t ▷ s_t is to be kept
12: $L \leftarrow \zeta_L(s_t)$
13: init. ζ_{BR} as an empty map of s_t backwards transitions in (Σ, Q) to lists of rank sets
14: **for each** $i \in 1 \ldots |SR_t|$ **do**
15: $(w, \sigma, s_s) \leftarrow L[i]$
16: **if** $\zeta_{BR}(\sigma, s_s) = \perp$ **then**
17: $\zeta_{BR}(\sigma, s_s) \leftarrow$ empty list
18: **end if**
19: append $SR_t[i]$ to $\zeta_{BR}(\sigma, s_s)$
20: **end for**
21: **for each** $(\sigma, s_s, BR) : \zeta_{BR}(\sigma, s_s) = BR$ **do**
22: $\zeta_{TR}(s_s, \sigma, s_t) \leftarrow \bigcup_{R \in BR} R$
23: **if** $\zeta_{SR}(s_s) = \perp$ **then**
24: $\zeta_{SR}(s_s) = \emptyset$
25: **end if**
26: $SR_s \leftarrow \zeta_{SR}(s_s)$
27: **for each** $i = 1 \ldots |SR_t| - |SR_s|)$ **do**
28: append $\emptyset$ to SR_s
29: **end for**
30: **for each** $i = 1 \ldots |SR_s|$ **do**
31: $SR_s[i] = SR_s[i] \cup SR_t[i]$
32: **end for**
33: **end for**
34: **end for**
35: mark q_I

Algorithm 6 dfa_carving_cleanup(A, A_{sort}, ζ_L, ζ_{TR})

Input: $A = (Q, \Sigma, \delta, q_I, F)$, a DFA that underwent carving backward propagation
$A_{sort} : Q^{|Q|-1}$, topological sort of A
ζ_L, map of states to top back. transitions
ζ_{TR}, map of transitions to rank sets
Output: A after clean up

1: $s_f \leftarrow$ last state in A_{sort}
2: **for each** $(w, \sigma, s_s) \in \zeta_L(s_f)$ **do** ▷ note $\sigma = \varepsilon$
3: remove transition $\delta(s_s, \sigma) = s_f$
4: $F \leftarrow F \cup \{s_s\}$
5: **end for**
6: **for each** $s_s \in A_{sort}$ **do**
7: **if** s_s is marked **then**
8: **for each** $(\sigma, s_t) : \delta(s_s, \sigma) = s_t$ and $\zeta_{TR}(s_s, \sigma, s_t) = \perp$ **do**
9: remove transition $\delta(s_s, \sigma) = s_t$
10: **end for**
11: **else** remove s_s from A along with all transitions from s_s
12: **end if**
13: **end for**
14: remove s_f from A

B Supplementary Material

Figure 12: Dialog flow for top 3 restaurant booking paths. Bubble colors are: purple for the dialog start (initial state), blue for customer utterances (DFA states), and gray/green for agent questions/responses (DFA transitions).

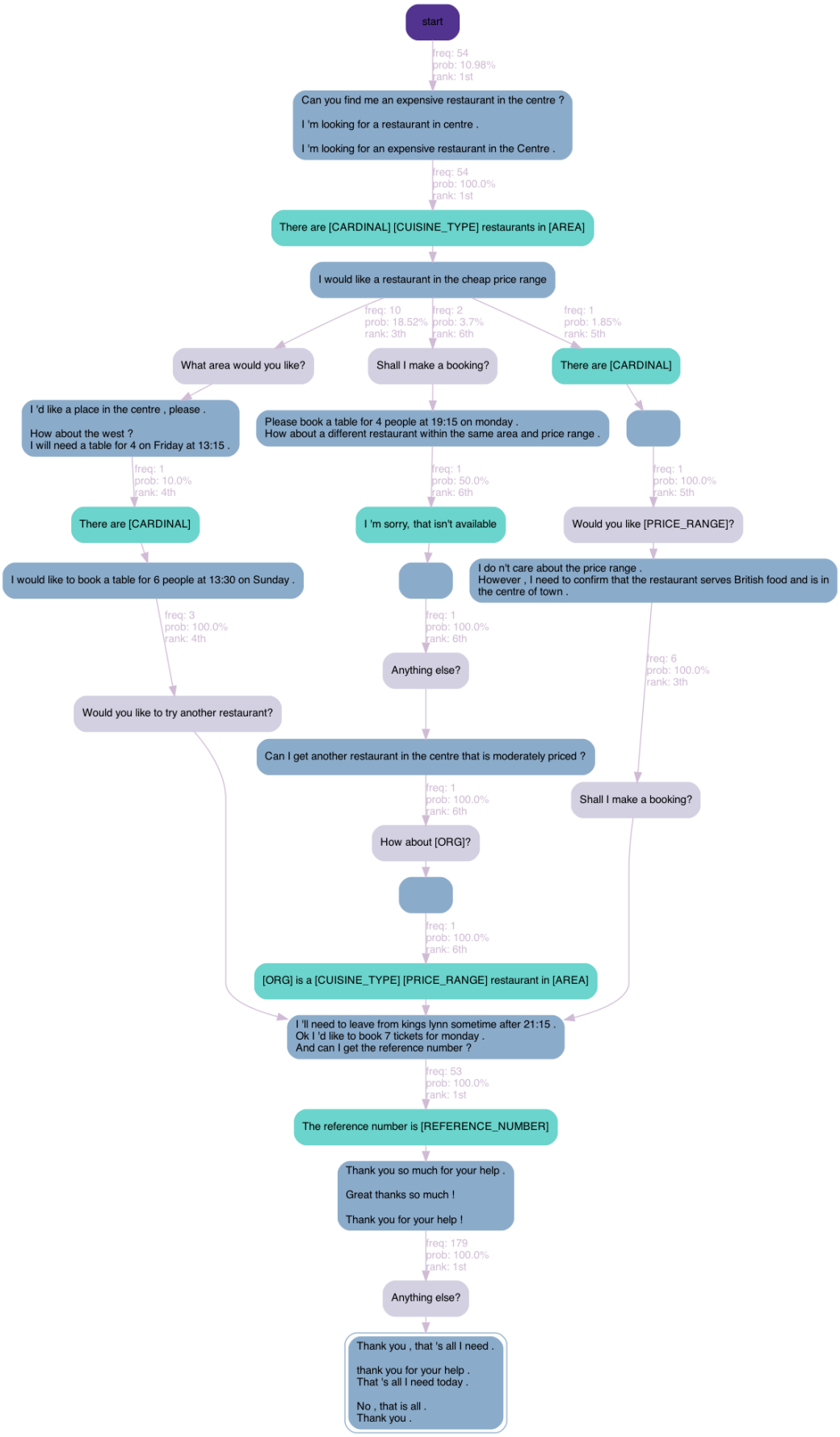


Figure 13: Dialog flow for top restaurant booking paths 4 to 6. Bubble colors are: purple for the dialog start (initial state), blue for customer utterances (DFA states), and gray/green for agent questions/responses (DFA transitions).

Structured Dialogue Discourse Parsing

Ta-Chung Chi
Language Technologies Institute
Carnegie Mellon University
tachungc@andrew.cmu.edu

Alexander I. Rudnicky
Language Technologies Institute
Carnegie Mellon University
air@cs.cmu.edu

Abstract

Dialogue discourse parsing aims to uncover the internal structure of a multi-participant conversation by finding all the discourse *links* and corresponding *relations*. Previous work either treats this task as a series of independent multiple-choice problems, in which the link existence and relations are decoded separately, or the encoding is restricted to only local interaction, ignoring the holistic structural information. In contrast, we propose a principled method that improves upon previous work from two perspectives: encoding and decoding. From the encoding side, we perform structured encoding on the adjacency matrix followed by the matrix-tree learning algorithm, where all discourse links and relations in the dialogue are jointly optimized based on latent tree-level distribution. From the decoding side, we perform structured inference using the modified Chiu-Liu-Edmonds algorithm, which explicitly generates the labeled multi-root non-projective spanning tree that best captures the discourse structure. In addition, unlike in previous work, we do not rely on hand-crafted features; this improves the model's robustness. Experiments show that our method achieves new state-of-the-art, surpassing the previous model by 2.3 on STAC and 1.5 on Molweni (F1 scores). [1]

1 Introduction

Discourse parsing is a series of tasks that consist of elementary discourse unit (EDU) segmentation, relation directionality classification (optional), and relation type classification between EDUs (Jurafsky and Martin, 2021). It serves as the first step of many downstream applications (Meyer and Popescu-Belis, 2012; Jansen et al., 2014; Narasimhan and Barzilay, 2015; Bhatia et al., 2015; Ji et al., 2016; Asher et al., 2016; Ji and Smith, 2017; Li et al., 2020a), and it can be categorized into three major discourse formalisms: RST (Mann and Thompson, 1988), PDTB (Prasad et al., 2008), and SDRT (Lascarides and Asher, 2008) styles. Considering that SDRT-style formalism is used to label the STAC (Asher et al., 2016) and Molweni (Li et al., 2020a) dialogue corpora and the increasing importance of dialogue discourse parsers trained on them (Ouyang et al., 2021; Feng et al., 2021; Jia et al., 2020; Chen and Yang, 2021), we focus on designing an SDRT-style dialogue discourse parser using the two corpora in this work. Figure 1 presents an example of a dialogue session in the STAC corpus (Asher et al., 2016) annotated with its discourse structure. The annotation is often encoded in two components: *links* and *relations*. The goal of a dialogue discourse parser is to extract them accurately at the same time.

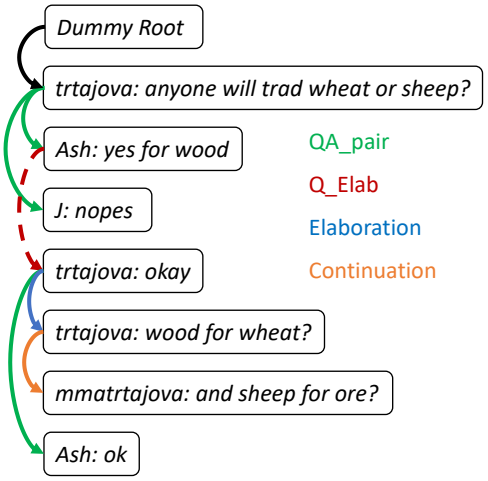


Figure 1: This is an example dialogue session. The ultimate goal of a dialogue discourse parser is to predict all the links (arrows) and relations (color of arrows) shown in this figure. Note that the Q_Elab arrow (dashed) can cross the QA_pair one, making it non-projective.

One straightforward solution to this problem is to transform the parsing structure into a series of local pairwise link prediction problems. In other words, the model is expected to compute some

[1] Code released at https://github.com/chijames/structured_dialogue_discourse_parsing.

Proceedings of the SIGdial 2022 Conference, pages 325–335
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Models	Encoding	Decoding	Link & Relation Prediction	Use Feature
MST (2015)	*local, edge-wise*	*partial MST*	*separate*	Y
ILP (2016)	*local, edge-wise*	*ILP*	*separate*	Y
Deep-Seq. (2019)	*global, two-staged*	*indp. multiple choice*	*separate*	Y
Struct-Aware (2021)	*global, fully-connected*	*indp. multiple choice*	*separate*	Y
Hierarchical (2021)	*hierarchical*	*indp. multiple choice*	*separate*	N
This Work	*global, structured*	*full MST*	*joint*	N

Table 1: This is the comparison between different dialogue discourse parsers. Our method is designed with structured encoding and decoding processes. Furthermore, the links and relations are learned and predicted jointly. Finally, our method does not rely on human-designed features, hence enjoys better robustness.

local potentials between each pair of utterances, and predict the relation type of that link if it exists. However, this formulation does not take the global structural information into account, leading to inferior parsing performance.

In contrast to previous work, our core observation is that by adding a dummy utterance at the beginning of the dialogue, the overall structure closely resembles a **labeled multi-root non-projective spanning tree**. In light of this observation, we propose a principled dialogue discoursing parser that encodes structural inductive biases during training and inference.

The essential elements of our method are the novel structured parameterization of the adjacency matrix, the directed version of matrix-tree theorem (Tutte, 1984; Koo et al., 2007), and the modified directed spanning tree inference algorithm. To the best of our knowledge, this is the first time that the labeled multi-root non-projective spanning tree is applied to the analysis of dialogue discourse structure. In summary, the contributions of this paper are:

- We propose a principled method for the dialogue discourse parsing task, where structural inductive biases for both encoding and decoding processes are introduced.
- We jointly predict discourse *links* and *relations* in a unified space.
- We propose a padding method that allows batchwise variable-length determinant calculation.
- Experimental results demonstrate state-of-the-art discoursing parsing performance on two datasets.

2 Task Background

We are given a dialogue session D and the links and relations between pairs of utterances labeled using the 17 discourse relations defined in Asher et al. (2016). All the utterances, links, and relations constitute a graph $G(V, E, R)$, where V represents the set of utterances, E represents the links connecting them, and R represents the edge labels. The goal of a discourse parser is to predict E and R given V.

There are five existing dialogue discourse parsers to the best of our knowledge (Afantenos et al., 2015; Perret et al., 2016; Shi and Huang, 2019; Wang et al., 2021; Liu and Chen, 2021). We compare them against each other in detail in the following subsections and provide a summary in Table 1.

2.1 Encoding

Afantenos et al. (2015); Perret et al. (2016) use a MaxEnt (Ratnaparkhi, 1997) model to parameterize local pairwise scores between utterance pairs. Therefore, global and contextual information are not taken into account during the encoding process. Liu and Chen (2021) improve upon them by using a hierarchical encoder that models the contextual information. Shi and Huang (2019) inject more structural information by first predicting all the links, followed by a global structured encoding module. However, the predicted links are discrete, making this two-staged solution not end-to-end trainable. To connect the two stages, Wang et al. (2021) instead use a fully connected graph between all utterances. While being fully end-to-end, useful structured bias is not encoded anymore. Based on the drawbacks of previous parsers, we propose a fully end-to-end encoder while maintaining structured information at the same time.

2.2 Decoding

Shi and Huang (2019); Wang et al. (2021); Liu and Chen (2021) treat the links and relations decoding tasks as a series of independent multiple-choice problems. In other words, the existence of

one link has nothing to do with other links. In contrast, Perret et al. (2016) find the structure by solving an integer linear programming problem, but it needs a set of complicated human-designed decoding constraints. Afantenos et al. (2015) is the closest approach to this work, where they run the maximum spanning tree decoding algorithm on the predicted edges only to find the tree structure (links). However, the relations are not jointly decoded. Instead, we run the modified spanning tree decoding algorithm on the unified link and relation space.

2.3 Link and Relation Prediction

All previous work treat the prediction of links and relations as a two-stage process. That is, they first predict the existence of a link, and the relation is predicted only if the link exists. This decouples the joint learning of links and relations. We mitigate this issue by unifying the prediction space of links and relations, making it a three-dimensional tensor.

2.4 Feature Usage

Finally, all previous work execpt (Liu and Chen, 2021) utilize some hand-crafted features. To name a few, they explicitly model if two utterances are spoken by the same *speaker*, or if they belong to the same *turn*. These features are useful but also make the baseline parsers deeply coupled with them, which might limit the parsing performance if applied to a new dataset. For example, if the new dataset is a transcript of a teleconference or radio exchange, it is likely that we only have the utterances recorded as it is expensive and hard to obtain all the speaker and turn information. In contrast, since our model does not rely on such explicitly modeled feature, the performance drop is less than the ones that use them when the speaker and turn information are removed.

3 Structure Formulation

The graph G defined in § 2 can theoretically be any directed acyclic graph, which is generally difficult to optimize. Fortunately, we find that by discarding only a small fraction of the edges, which is 6% for the STAC corpus and 0% for the molweni corpus, we can recover a spanning tree-like structure that permits efficient learning and structure inference. For the nodes having more than one parent, we keep only the latest one.[2] In addition, for dangling utterances that do not have any parents, we connect them to the dummy root utterance, so we are in fact optimizing a *multi-root* tree during training time. Finally, note that our tree structure allows different links to *cross* each other (Figure 1) and each edge also has a relation *label*, $G(V, E, R)$ is a labeled directed multi-root non-projective spanning tree, which is referred to as tree for conciseness hereinafter [3].

Several questions naturally arise:

- How to parametrize the tree? We will model the pairwise potential scores by an adjacency matrix, where a cell represents the relevance score of a pair of utterances. See § 4.1.
- How to learn the correct tree? We calculate the probability of the correct tree among all possible trees encoded by the adjacency matrix, and that probability is maximized. This is similar to softmax attention using trees as basic units instead of tokens. See § 4.2.
- How to perform inference? Given the learned three-dimensional adjacency matrix, we can run the modified maximum spanning tree induction algorithm to induce the tree structure. See § 4.4.

4 Proposed Framework

4.1 Model Parameterization

Given n utterances (aka EDUs) $\{U_i\}_{i=1}^{n}$ in a dialogue session D, we define a *discourse pair* in D as a 3-tuple $(h, m, r), h < m, r \in [1, 17]$ where $h \in [0 \dots n]$ is the index of the parent utterance, $m \in [1 \dots n]$ is the index of the children utterance, and r is one of the 17 relations. Note that we add a special *root* utterance $h = 0$ to be the shared pseudo parent for the first utterances. Note that this root utterance can be chosen arbitrarily, and we use the utterance "This is the start of a dialogue" in this work.

To parameterize the tree, we first model the d-dimensional pairwise representation between a pair of utterances. This can be expressed compactly by a 3-order tensor, which is an adjacency matrix with each element $V_{h,m}$ being a d-dimensional feature vector, hence $V \in \mathbb{R}^{(n+1)\times(n+1)\times d}$. Each $V_{h,m}$ is calculated using a BERT (Devlin et al., 2019) model as the encoder. BERT takes a pair of utterances as input, and a special [CLS] token

[2] This strategy is adopted by all baselines as well.

[3] There are four types of spanning trees investigated in the dependency parsing domain (McDonald et al., 2005; Koo et al., 2007)

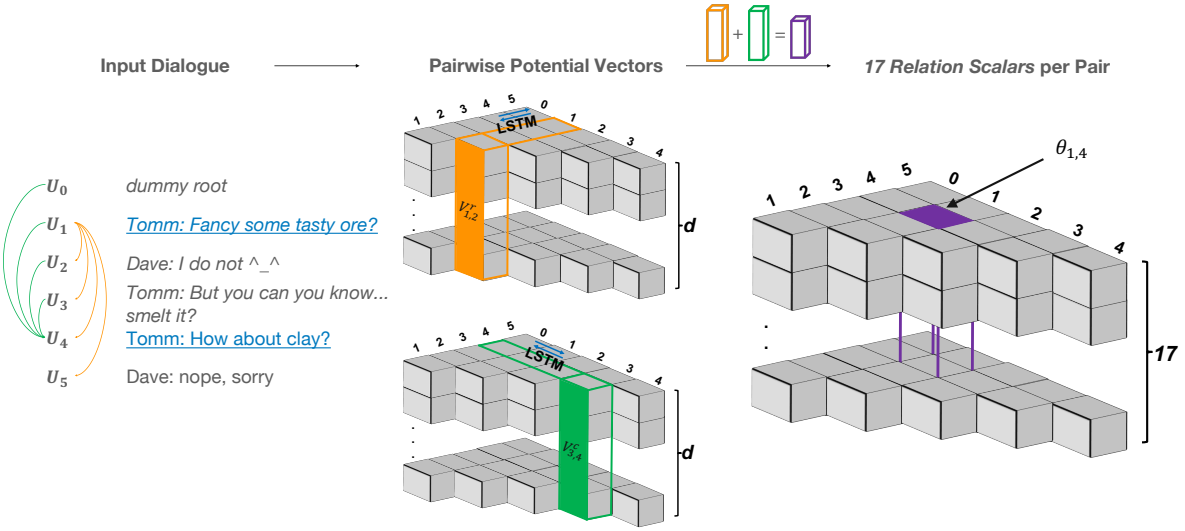


Figure 2: The contextual encoding process. Row numbers from 0 to 4 represent h, and column numbers from 1 to 5 represents m. In this example, the 1-st utterance can connect to the 2-nd, 3-rd or 5-th utterances when predicting the $V_{1,4}$ cell. This is represented by the orange rectangles. Similarly, the 4-th utterance can have 0, 1, or 3-rd utterances as its parent, represented by the green rectangles. We use two LSTMs for two directions (orange and green). Finally, we add the contextualized vectors together to get the purple vector $\text{Linear}(V^r_{1,4} + V^c_{1,4}) = \theta_{1,4}$.

is prepended before the concatenation of the two utterances. The representation of the [CLS] token is further used to calculate d-dimensional pairwise representation:

$$V_{h,m} = \text{BERT}_{CLS}(U_h, U_m) \quad (1)$$

One immediate drawback of eq. (1) is that the pairwise scores are calculated independently. We alleviate this issue by using a bidirectional LSTM to encode the contextual information. For the h-th row, we obtain the hidden states for all timestep t:

$$\{V^{\text{r}}_{h,t}\}^n_{t=h+1} = \text{LSTM}(\{V_{h,t}\}^n_{t=h+1}) \quad (2)$$

The underlying idea is that to accurately decide if U_h should point to U_m, we should collect the information of connecting U_h to all the other utterances that appear later chronologically. Similarly, for the m-th column, all the hidden states are:

$$\{V^{\text{c}}_{t,m}\}^{m-1}_{t=0} = \text{LSTM}(\{V_{t,m}\}^{m-1}_{t=0}) \quad (3)$$

The final context-aware potential score is:

$$\tilde{V}_{h,m} = V^{\text{r}}_{h,m} + V^{\text{c}}_{h,m} \quad (4)$$

$\tilde{V} \in \mathbb{R}^{(n+1)\times(n+1)\times 2d}$. Every pairwise score is now aware of neighboring pairs. It still remains to convert $\tilde{V}$ to individual score of a discourse pair. We do so by simply passing $\tilde{V}$ through a linear transformation layer:

$$\theta_{h,m} = \text{Linear}(\tilde{V}) \quad (5)$$

where $\theta \in \mathbb{R}^{(n+1)\times(n+1)\times 17}$. Note that there is no activation function after the linear layer since we assume θ to be in log space. Another important property of θ is that it is a strictly *upper-triangular* matrix due to the $h < m$ constraint. In practice, this can be enforced by setting the lower triangular and diagonal elements to -inf. We illustrate the overall idea in Figure 2.

4.2 Learning the Tree

The parameterization we define in eq. (5) still does not impose any structural constraints. Based on our conclusion in § 3, we would like to impose a non-projective multi-root spanning tree constraint in the learning and inference process. We define a tree T to be the collection of discourse pairs $\{(h, m, r)\}$. We use $\mathcal{T}(D)$ to denote all possible trees of a dialogue session D. During learning, the reference tree structure $\bar{T} \in \mathcal{T}(D)$ is given. If the score of $\bar{T}$ and the summation of the scores of all trees in $\mathcal{T}(D)$ is tractable, we can obtain the probability of $\bar{T}$ and optimize it using gradient descent. The challenge lies in the exponentially many candidates of $\mathcal{T}(D)$, which is computationally infeasible to naively enumerate. Fortunately, we will see that the Matrix-Tree Theorem (Tutte, 1984; Koo et al., 2007) permits efficient calculation of the summation we need.

4.2.1 Matrix-Tree Theorem

Before we dive into the details of the Matrix-Tree Theorem, we have to give enough credits to Tutte

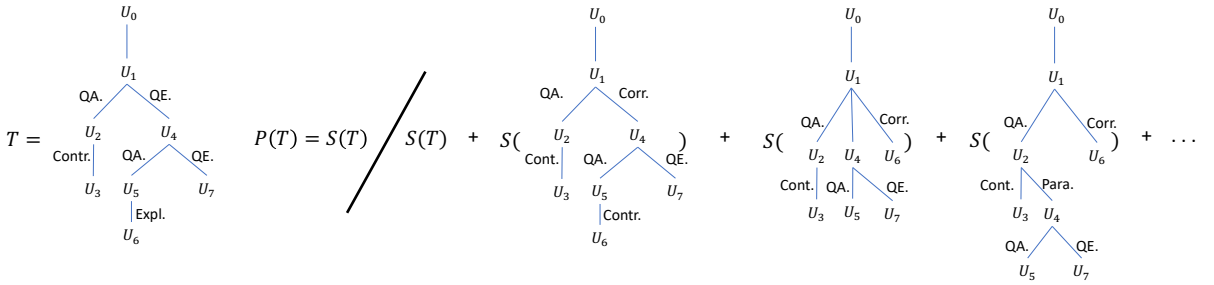


Figure 3: This is the graphical illustration of of how we perform tree-structured learning. Note that we are summing the scores of all possible labeled trees as the denominator, which is eq. (11).

(1984); Koo et al. (2007) as we are applying their proposed theorem. The probability of the reference tree $\bar{T}$, which is to be optimized, can be defined as:

$$\mathbb{P}(\bar{T}) = \frac{s(\bar{T})}{Z(\theta)}, \quad Z(\theta) = \sum_{T \in \mathcal{T}(D)} s(T) \tag{6}$$

$Z(\theta)$ is also known as the partition function. The numerator $s(T)$ of any tree T is defined to be:

$$s(T) = \prod_{(h,m,r) \in T} \exp(\theta_{h,m,r}), \tag{7}$$

With this definition, the score is merely the product of corresponding cells in $\exp(\theta)$ (θ from eq. (5)).

Next, we need to find an efficient way to compute the partition function as there are exponentially many candidate trees. The first step is to calculate the exponential matrix $A_{h,m,r}$ from eq. (5):

$$A_{h,m,r}(\theta) = \begin{cases} 0, & \text{if } h \geq m \\ \exp(\theta_{h,m,r}), & \text{otherwise} \end{cases} \tag{8}$$

Note that the first $0 = \exp(-inf)$ condition applies to all $h \geq m$ cells as θ is an upper-triangular matrix described earlier. To account for edge labels, we have to marginalize r out :

$$A_{h,m}(\theta) = \sum_{r} A_{h,m,r}(\theta) \tag{9}$$

Now, we are ready to calculate $Z(\theta)$. The first step is to calculate the graph Laplacian matrix, which is the difference between the degree matrix and adjacency matrix:

$$L_{h,m}(\theta) = \begin{cases} \sum_{i'=1}^{n} A_{i',m}(\theta), & \text{if } h = m \\ -A_{h,m}(\theta), & \text{otherwise} \end{cases} \tag{10}$$

Then the minor[4] $L^{(0,0)}(\theta)$ is equal to the sum of the weights of all directed spanning trees rooted at the dummy root utterance (Tutte, 1984):

$$Z(\theta) = \det(\hat{L}), \tag{11}$$

whre $\hat{L}$ is defined to be the submatrix constructed by removing the first row and column from L. The computational complexity of eq. (11) is determined by the determinant operation, which is $O(n^3)$. While the cubic time complexity might seem scary at the first glance, it does not incur significant computational overhead in our experiments, where the time to compute the determinant is negligible (< 1%) compared to BERT encoding in eq. (1).

4.2.2 Efficient GPU Implementation

The equations derived so far work well for a single training instance. However, it becomes problematic if we want to perform batchwise training on GPUs, which was not addressed in Koo et al. (2007). The main challenge is the variable-length padding. In particular, we have to calculate batchwise determinants in eq. (11) with different sizes of $\hat{L}$. The naive option is to pad the extra rows and columns with zeros. Unfortunately, this would result in a singular matrix and give erroneous partition results. To circumvent the padding issue, we can use the cofactor expansion formula. Concretely, all the diagonal elements of the padding part should be 1, while others should be 0. We illustrate the padding strategy in Figure 4. Note that this strategy holds whether the size of $\hat{L}$ is odd or even.

4.3 Optimization of Tree

Since we are given the reference tree $\bar{T}$, we can directly maximize the log probability of eq. (6) using any gradient-descent based algorithms, which is also equivalent to minimizing the KL-divergence between the predicted and reference tree distributions.

[4] A minor $L^{(x,y)}$ is the determinant of a submatrix constructed by removing the x-th row and y-th column of L.

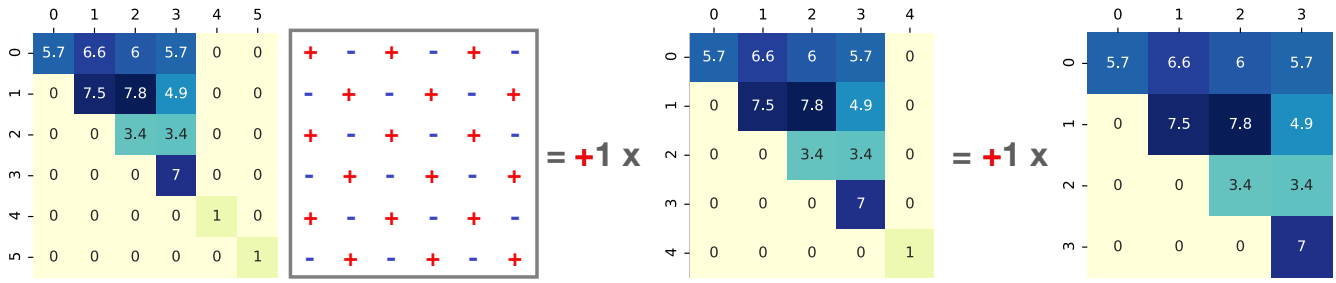


Figure 4: This is an efficient padding for calculating batch determinant. The original 4×4 matrix is expanded to a 6×6 (leftmost) one for padding. Note that the last two diagonal elements are all ones. The second matrix encodes the coefficients for multiplying sub-matrices. After a series of cofactor expansions, we can see that the determinant of the padded 6×6 matrix is equivalent to the original unpadded 4×4 matrix.

4.4 Inference of Tree

There is a well-known algorithm - Chiu-Liu-Edmonds (CLE) (Edmonds, 1967; Chu, 1965) that can find the directed spanning tree $\tilde{T}$ with maximum weight given $A(\theta)$ derived in eq. (8). However, we cannot directly apply the CLE algorithm as the original version does not accept labeled trees. To solve this problem, we have to first pick the highest-scoring relation for each edge $A'_{h,m} = \max_r A_{h,m,r}$ to get $A' \in \mathbb{R}^{(n+1)\times(n+1)}$. Now we can feed this standard form into the CLE algorithm: $\tilde{T} = \text{CLE}(A')$. The correctness of this approach can be proved easily by contradiction: suppose the optimal tree includes one edge that is not the highest-scoring one among $\{A_{h,m,r}\}_{r=1}^{17}$, we can always substitute that edge with the highest-scoring one to get a better tree (contradiction). Note that for a pair of utterances, we only allow one direction of link (θ is strictly upper triangular) so the CLE algorithm in fact degenerates to its undirected version known as Prim/Kruskal's algorithm (Prim, 1957; Kruskal, 1956).

5 Experiments

5.1 Datasets

There are two datasets for us to train the discourse parser, one of which is the STAC (Asher et al., 2016) corpus, which is a multi-party dialogue corpus collected from an online game, and the other is the Ubuntu IRC corpus (Li et al., 2020a), which compiles technical discussions about Linux. The differences between these two datasets were analyzed in Liu and Chen (2021), where the takeaway messages are: 1) there is no significant difference in their average EDU numbers, 2) the lexical distributions are significantly different sharing only a small portion of common tokens, 3) relation distributions are similar.

5.2 Hyperparameters

Following Liu and Chen (2021), we use the Roberta-Base uncased pretrained checkpoint for a fair comparison. The max utterance length is set to 28. The initial learning rate is set to 2e-5 with a linear decay to 0 for 4 epochs. The batch size is 4. The first 10% of training steps is the warmup stage. For all baselines using large pretrained models, we always use the same model checkpoint and tune the learning rate and batch size for them for a fair comparison.

5.3 Metrics

We follow the baselines to use two metrics for evaluation:

- Unlabeled Attachment Score (UAS): We only care about the existence of a discourse link. In other words, discourse relations do not affect the results. (Also known as Link F1 score)
- Labeled Attachment Score (LAS): It is much harder as it requires both discourse links and relations to be correct. We focus primarily on this metric since it is more informative and is used in downstream applications. (Also known as Link & Rel F1 score)

5.4 Main Results

We present the results in Table 2. The left part of the table focuses on in-domain training and testing, which is the standard setting. Bearing in mind that discourse parsers are often used as the first stage of downstream applications, we follow (Liu and Chen, 2021) to benchmark the performance of all parsers in the cross-domain setting. Note that this is an extremely challenging setting as the domains are completely different (gaming vs Linux technical forum).

	STAC / STAC		MOL / MOL		STAC / MOL		MOL / STAC	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
MST (Afantenos et al., 2015)	69.6	52.1	69.0	48.7	61.5	24.0	**60.5**	14.8
ILP (Perret et al., 2016)	69.0	53.1	67.3	48.3	57.0	24.1	60.4	14.5
Deep-Seq. (Shi and Huang, 2019)	73.2	54.4	76.1	53.3	53.5	21.6	42.7	15.7
Hierarchical (Liu and Chen, 2021)	73.1	57.1	80.1	56.1	60.1	32.1	48.9	26.8
Struct-Aware (Wang et al., 2021)	73.4	57.3	81.6	58.4	57.0	32.9	44.7	26.1
This Work	**74.4**	**59.6**	**83.5**	**59.9**	**64.5**	**38.0**	50.6	**31.6**

Table 2: STAC / MOL means the training dataset is STAC and the testing dataset is MOL. LAS is the harder setting used for downstream applications. Results are the average of three runs. Note that speaker information is still used in this set of experiments, except our parser does not need to model their relations explicitly as described in § 2.4.

In-domain We first take a look at the in-domain results. Our proposed parser is the best among all parsers, surpassing the previous state-of-the-art by 2.3 on STAC and 1.5 (F1 scores) on Molweni under the LAS setting. The trend is similar for the UAS setting. We also want to highlight the improved performance can NOT be attributed to using a pretrained language model as *Struct-Aware* and *Hierarchical* (Liu and Chen, 2021) both utilize the same or comparable pretrained model.

Cross-domain We shift gear to the cross-domain setting where the parser is trained on one dataset and tested on the other (Liu and Chen, 2021). We can see that our parser is the best under the LAS setting, substantially outperforming the best candidates by 5.1 on STAC/MOL and 4.8 points on MOL/STAC. However, the best-performing model under the UAS setting is the oldest model (Afantenos et al., 2015; Perret et al., 2016). This can be explained by the inclination of a large pretrained model to overfit on the training domain, which was corroborated by Liu and Chen (2021) as well. Readers might wonder why the same phenomenon does not happen under the UAS setting of STAC/MOL, and the speculated reason is STAC has a much larger linguistic diversity (Liu and Chen, 2021), thereby alleviating the model overfitting issue. In other words, we might want to train the dialogue discourse parser on a linguistically diverse dataset if the goal is domain generalization.

5.5 Additional Analyses

Dialogue Length Robustness We hypothesize that our parser is likely to perform better when the dialogue becomes longer, and the reason is that our parser models the overall dialogue structure using tree distributions. This lowers the burden of the parser to predict long-range links. We focus on the in-domain setting and plot the results in Figure 5. As we can see, the performance of our parser drops less than baselines when the dialogue becomes longer, highlighting the benefit of global structured learning and inference.

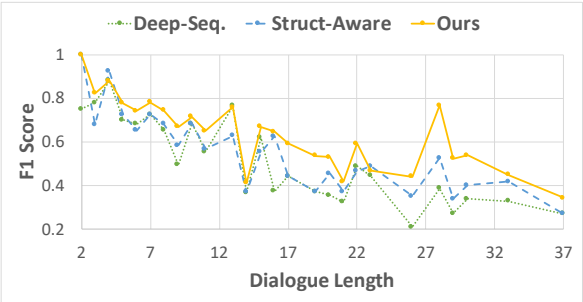


Figure 5: Parsing performance w.r.t dialogue lengths. As we can see, the performance difference is larger when the dialogue becomes longer, demonstrating the length robustness of our parser.

Relation Performance Breakdown In order to know what kinds of relations benefit the most from our proposed parser, we count the number of correct relation predictions and plot them in Figure 6 and 7.[5] The baseline parser we compare with is the Hierarchical model (Liu and Chen, 2021) as it can be viewed as the non-structured version of our parser with the same pretrained model backbone. We can see that our parser outperforms the baseline on certain relations like *Comment* and *QA pair* on STAC and *QA pair* and *Clarification Question* on Molweni. However, there is still a large room for improvement as demonstrated by the gap between our parser and the ground truth. Another observation is that both parsers struggle to predict

[5]Note that this implies the link predictions of these correct relations are also correct.

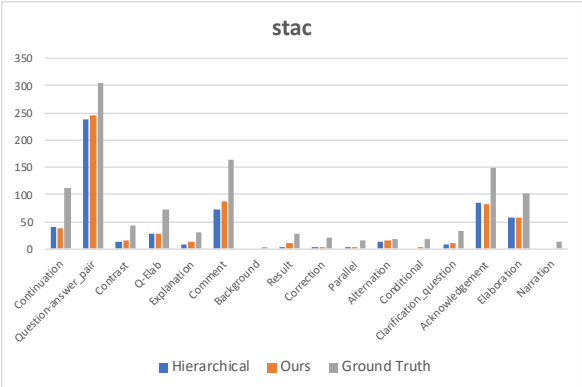


Figure 6: STAC relation performance breakdown.

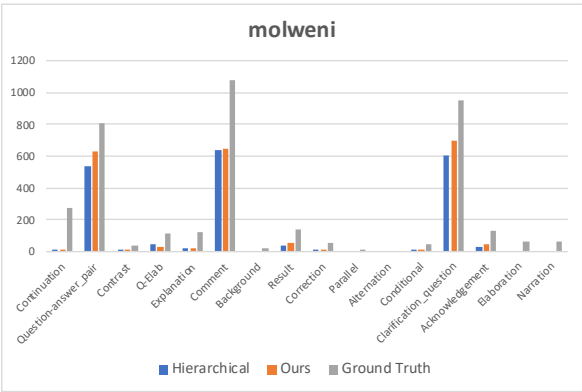


Figure 7: Molweni relation performance breakdown.

low-resource relations, marking an important direction for future work.

Speaker and Turn Feature Robustness We experiment with removing speaker and turn information used in baselines. The performance drop (LAS of STAC) of our parser (59.6 → 54.4) is less than that of the best baseline (Struct-Aware) (57.3 → 47.8), demonstrating the robustness of our parser.

6 Related Work

Discourse Parsing As discussed in the Introduction section, there are three types of discourse parsing formalisms: RST (Mann and Thompson, 1988), PDTB (Prasad et al., 2008), and SDRT (Lascarides and Asher, 2008; Asher et al., 2016). For the first two tasks, there are transition-based (Li et al., 2014; Braud et al., 2017; Yu et al., 2018) and CKY-based methods (Joty et al., 2015; Li et al., 2016; Liu and Lapata, 2017) in the literature. In this work, we assume that the EDUs are already given. In practice, there are papers working on segmenting EDUs (Subba and Di Eugenio, 2007; Li et al., 2018) before feeding them to the discourse parser.

Dialogue Disentanglement Clustering utterances in a conversation into threads is studied extensively by previous work (Shen et al., 2006; Elsner and Charniak, 2008; Wang and Oard, 2009; Elsner and Charniak, 2011; Jiang et al., 2018; Kummerfeld et al., 2019; Zhu et al., 2020; Li et al., 2020b; Yu and Joty, 2020). They predict the *reply-to* links independently and run a connected component algorithm to construct the threads. This is similar to the UAS setting in this work.

Structured Learning Algorithms Natural language is highly structured suggesting that the introduction of structural bias will facilitate learning. Previous work have studied dependency-tree like structures extensively (Koo et al., 2007; McDonald et al., 2005; McDonald and Satta, 2007; Niculae et al., 2018; Paulus et al., 2020). Several works propose to incorporate such inductive bias into intermediate layers of modern NLP models (Kim et al., 2017; Chen et al., 2017; Liu and Lapata, 2018; Choi et al., 2018). In our work, the induced structure is not only implicitly learned, it is also used to directly decode the labeled tree structure, which is our ultimate goal.

Dependency Parsing Our work can also be viewed as extending token-level dependency parsing (Mel'cuk et al., 1988; Koo et al., 2007; Smith and Eisner, 2008; Koo and Collins, 2010; Chen and Manning, 2014; Dozat and Manning, 2017; Qi et al., 2018; Choi and Palmer, 2011) to utterance-level. Another important difference is that our tree is labeled, which means we have to additionally predict the type of tree edges.

7 Conclusion

In this paper, we propose a principled method for dialogue discourse parsing. From the encoding side, we introduce a structurally-encoded adjacency matrix followed by the matrix-tree theorem, which is used to holistically model all utterances as a tree. From the decoding side, we apply the modified CLE algorithm for maximum spanning tree induction. Our method achieves state-of-the-art performance on two benchmark datasets. We also benchmark the cross-domain parser performance, and find our parser performs the best in the most-commonly used and harder LAS setting. We believe that the techniques described in this work pave the way for more structured analyses of dialogue and interesting research problems in the field

of dialogue discourse parsing.

References

Stergos Afantenos, Eric Kow, Nicholas Asher, and Jérémy Perret. 2015. Discourse parsing for multi-party chat dialogues. Association for Computational Linguistics (ACL).

Nicholas Asher, Julie Hunter, Mathieu Morey, Farah Benamara, and Stergos Afantenos. 2016. Discourse structure and dialogue acts in multiparty dialogue: the stac corpus. In *10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2721–2727.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2212–2218, Lisbon, Portugal. Association for Computational Linguistics.

Chloé Braud, Maximin Coavoux, and Anders Søgaard. 2017. Cross-lingual RST discourse parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 292–304, Valencia, Spain. Association for Computational Linguistics.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Huadong Chen, Shujian Huang, David Chiang, and Jiajun Chen. 2017. Improved neural machine translation with a syntax-aware encoder and decoder. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1936–1945, Vancouver, Canada. Association for Computational Linguistics.

Jiaao Chen and Diyi Yang. 2021. Structure-aware abstractive conversation summarization via discourse and action graphs. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1380–1391, Online. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jinho D Choi and Martha Palmer. 2011. Getting the most out of transition-based dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 687–692.

Yoeng-Jin Chu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Micha Elsner and Eugene Charniak. 2008. You talking to me? a corpus and algorithm for conversation disentanglement. In *Proceedings of ACL-08: HLT*, pages 834–842.

Micha Elsner and Eugene Charniak. 2011. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1179–1189.

Xiachong Feng, Xiaocheng Feng, Bing Qin, and Xinwei Geng. 2021. Dialogue discourse-aware graph model and data augmentation for meeting summarization. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3808–3814. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 977–986.

Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse-driven language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 332–342, San Diego, California. Association for Computational Linguistics.

Yangfeng Ji and Noah A. Smith. 2017. Neural discourse structure for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1005, Vancouver, Canada. Association for Computational Linguistics.

Qi Jia, Yizhu Liu, Siyu Ren, Kenny Zhu, and Haifeng Tang. 2020. Multi-turn response selection using dialogue dependency relations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1911–1920, Online. Association for Computational Linguistics.

Jyun-Yu Jiang, Francine Chen, Yan-Ying Chen, and Wei Wang. 2018. Learning to disentangle interleaved conversational threads with a siamese hierarchical network and similarity ranking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1812–1822.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 41(3):385–435.

Daniel Jurafsky and James H. Martin. 2021. *22*, 3 edition.

Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. 2017. Structured attention networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1–11.

Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. Structured prediction models via the matrix-tree theorem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150.

Joseph B Kruskal. 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50.

Jonathan K. Kummerfeld, Sai R. Gouravajhala, Joseph J. Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. 2019. A large-scale corpus for conversation disentanglement. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3846–3856, Florence, Italy. Association for Computational Linguistics.

Alex Lascarides and Nicholas Asher. 2008. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer.

Jiaqi Li, Ming Liu, Min-Yen Kan, Zihao Zheng, Zekun Wang, Wenqiang Lei, Ting Liu, and Bing Qin. 2020a. Molweni: A challenge multiparty dialogues-based machine reading comprehension dataset with discourse structure. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2642–2652, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jing Li, Aixin Sun, and Shafiq R Joty. 2018. Segbot: A generic neural text segmentation model with pointer network. In *IJCAI*, pages 4166–4172.

Qi Li, Tianshi Li, and Baobao Chang. 2016. Discourse parsing with attention-based hierarchical neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 362–371.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35.

Tianda Li, Jia-Chen Gu, Xiaodan Zhu, Quan Liu, Zhen-Hua Ling, Zhiming Su, and Si Wei. 2020b. Dialbert: A hierarchical pre-trained model for conversation disentanglement. *arXiv preprint arXiv:2004.03760*.

Yang Liu and Mirella Lapata. 2017. Learning contextually informed representations for linear-time discourse parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1289–1298.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

Zhengyuan Liu and Nancy Chen. 2021. Improving multi-party dialogue discourse parsing via domain integration. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 122–127, Punta Cana, Dominican Republic and Online. Association for Computational Linguistics.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of human language technology conference and conference on empirical methods in natural language processing*, pages 523–530.

Ryan McDonald and Giorgio Satta. 2007. On the complexity of non-projective data-driven dependency parsing. In *Proceedings of the Tenth International Conference on Parsing Technologies*, pages 121–132.

Igor Aleksandrovic Mel'cuk et al. 1988. *Dependency syntax: theory and practice*. SUNY press.

Thomas Meyer and Andrei Popescu-Belis. 2012. Using sense-labeled discourse connectives for statistical machine translation. In *Proceedings of the EACL2012 Workshop on Hybrid Approaches to Machine Translation (HyTra)*, CONF.

Karthik Narasimhan and Regina Barzilay. 2015. Machine comprehension with discourse relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1253–1262.

Vlad Niculae, Andre Martins, Mathieu Blondel, and Claire Cardie. 2018. Sparsemap: Differentiable sparse structured inference. In *International Conference on Machine Learning*, pages 3799–3808. PMLR.

Siru Ouyang, Zhuosheng Zhang, and Hai Zhao. 2021. Dialogue graph modeling for conversational machine reading. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3158–3169, Online. Association for Computational Linguistics.

Max B. Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J. Maddison. 2020. Gradient estimation with stochastic softmax tricks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Jérémy Perret, Stergos Afantenos, Nicholas Asher, and Mathieu Morey. 2016. Integer linear programming for discourse parsing. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 99–109.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Robert Clay Prim. 1957. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal Dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Adwait Ratnaparkhi. 1997. A simple introduction to maximum entropy models for natural language processing. *IRCS Technical Reports Series*, page 81.

Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–42.

Zhouxing Shi and Minlie Huang. 2019. A deep sequential model for discourse parsing on multi-party dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7007–7014.

David A Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 145–156.

Rajen Subba and Barbara Di Eugenio. 2007. Automatic discourse segmentation using neural networks. In *Proc. of the 11th Workshop on the Semantics and Pragmatics of Dialogue*, pages 189–190.

WT Tutte. 1984. Graph theory, encyclopedia of mathematics and it applications.

Ante Wang, Linfeng Song, Hui Jiang, Shaopeng Lai, Junfeng Yao, Min Zhang, and Jinsong Su. 2021. A structure self-aware model for discourse parsing on multi-party dialogues. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3943–3949. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Lidan Wang and Douglas W Oard. 2009. Context-based message expansion for disentanglement of interleaved text conversations. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 200–208. Citeseer.

Nan Yu, Meishan Zhang, and Guohong Fu. 2018. Transition-based neural rst parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570.

Tao Yu and Shafiq Joty. 2020. Online conversation disentanglement with pointer networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6321–6330, Online. Association for Computational Linguistics.

Henghui Zhu, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Who did they respond to? conversation structure modeling using masked hierarchical transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9741–9748.

"Do you follow me?": A Survey of Recent Approaches in Dialogue State Tracking

Léo Jacqmin[1,2] **Lina M. Rojas-Barahona**[1] **Benoit Favre**[2]
[1]Orange Innovation, Lannion, France
[2]Aix-Marseille Université / CNRS / LIS, Marseille, France
{leo.jacqmin, linamaria.rojasbarahona}@orange.com
benoit.favre@lis-lab.fr

Abstract

While communicating with a user, a task-oriented dialogue system has to track the user's needs at each turn according to the conversation history. This process called dialogue state tracking (DST) is crucial because it directly informs the downstream dialogue policy. DST has received a lot of interest in recent years with the text-to-text paradigm emerging as the favored approach. In this review paper, we first present the task and its associated datasets. Then, considering a large number of recent publications, we identify highlights and advances of research in 2021-2022. Although neural approaches have enabled significant progress, we argue that some critical aspects of dialogue systems such as generalizability are still underexplored. To motivate future studies, we propose several research avenues.

1 Introduction

Since human conversation is inherently complex and ambiguous, creating an open-domain conversational agent capable of performing arbitrary tasks is an open problem. Therefore, the practice has focused on building task-oriented dialogue (TOD) systems limited to specific domains such as flight booking. These systems are typically implemented through a modular architecture that provides more control and allows interaction with a database, desirable features for a commercial application. Among other components, dialogue state tracking (DST) seeks to update a representation of the user's needs at each turn, taking into account the dialogue history. It is a key component of the system: the downstream dialogue policy uses this representation to predict the next action to be performed (e.g. asking for clarification). A response is then generated based on this action. Beyond processing isolated turns, DST must be able to accurately accumulate information during a conversation and adjust its prediction according to the observations to provide a summary of the dialogue so far.

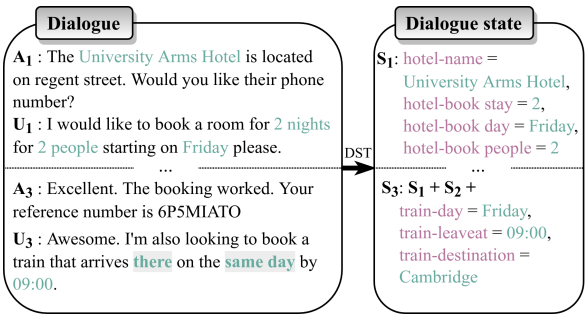


Figure 1: Dialogue state tracking (DST) example taken from MultiWOZ. DST seeks to update the dialogue state at each turn as (slot, value) pairs. Current DST models struggle to parse U_3, which requires world knowledge and long-term context awareness.

Several papers have surveyed DST from the perspective of a specific paradigm or period. Young et al. (2013) give an overview of POMDP-based statistical dialogue systems. These generative approaches model the dialogue in the form of a dynamic bayesian network to track a belief state. They made it possible to account for the uncertainty of the input related to speech recognition errors and offered an alternative to conventional deterministic systems which were expensive to implement and often brittle in operation. However, POMDP-based systems presuppose certain independencies to be tractable and were gradually abandoned in favor of discriminative approaches that directly model the dialogue state distribution. These methods, surveyed by Henderson (2015), rely on annotated dialogue corpora such as those from the Dialogue State Tracking Challenge, the first standardized benchmark for this task (Williams et al., 2016). The past years have been marked by the use of neural models that have allowed significant advances, covered by Balaraman et al. (2021). Since then, many works dealing with key issues such as adaptability have been published. The first contribution of this paper is a synthesis of recent advances in order to identify the major achievements in the field.

Proceedings of the SIGdial 2022 Conference, pages 336–350
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

The recent development of neural models has addressed fundamental issues in dialogue systems. DST models can now be decoupled from a given domain and shared across similar domains (Wu et al., 2019). They even achieve promising results in zero-shot scenarios (Lin et al., 2021b). Despite these advances, modern approaches are still limited to a specific scenario as dialogues in most DST datasets consist of filling in a form: the system asks for constraints until it can query a database and returns the results to the user. Moreover, these data-driven approaches are rigid and do not correspond to the need for fine-grained control of conversational agents in an application context. In a real-world scenario, access to annotated data is limited but recent DST models appear to have poor generalization capacities (Li et al., 2021a). These limitations show that there remain major challenges to the development of versatile conversational agents that can be adopted by the public. The second contribution of this paper is to propose several research directions to address these challenges.

2 Dialogue State Tracking

A successful TOD system makes it possible to automate the execution of a given task in interaction with a user. The last few years have seen an increase in research in this field, which goes hand in hand with the growing interest of companies in implementing solutions based on TOD systems to reduce their customer support costs.

A typical modular architecture for these systems is composed of the following components: natural language understanding (NLU), DST, dialogue policy, and natural language generation (NLG). NLU consists of two main subtasks, namely, intent detection which consists in identifying the user's intent, such as booking a hotel, and slot filling which consists in identifying relevant semantic concepts, such as price and location. DST aims to update the dialogue state, a representation of the user's needs expressed during the conversation. The dialogue policy predicts the system's next action based on the current state. Along with DST, it forms the dialogue manager which interfaces with the ontology, a structured representation of the back-end database containing the information needed to perform the task. Finally, NLG converts the system action into natural language.

In the case of a spoken dialogue system, automatic speech recognition (ASR) and speech synthesis components are integrated to move from speech to text and back. NLU then operates on the ASR hypotheses and is denoted SLU for spoken language understanding.[1] Traditionally, DST operates on the output of SLU to update the dialogue state by processing noise from ASR and SLU. For example, SLU may provide a list of possible semantic representations based on the top ASR hypotheses. DST manages all these uncertainties to update the dialogue state. However, recent DST datasets are collected in text format with no consideration for noisy speech inputs, which has caused slot filling and DST to be studied separately.

Note that such modular architecture is one possible approach to conversational AI and others have been considered. Another approach is end-to-end systems which generate a response from the user's utterance using a continuous representation. Such systems have been studied extensively for open-domain dialogue (Huang et al., 2020) and have also been applied to TOD (Bordes et al., 2017).

2.1 Dialogue State

Consider the example dialogue shown in Figure 1. Here, the agent acts as a travel clerk and interacts with the user to plan a trip according to the user's preferences. The dialogue state is a formal representation of these constraints and consists of a set of predefined slots, each of which can take a possible value, e.g. `book day` = `Friday`. At a given turn t, the goal of DST is to extract values from the dialogue context and accumulate information from previous turns as the dialogue state S_t, which is then used to guide the system in choosing its next action towards accomplishing the task, e.g. finding a suitable hotel. Traditionally, informable slots (constraints provided by the user, e.g. `price range`) are distinguished from requestable slots (information that the user can request, e.g. `phone number`). Informable slots can take a special value: `don't care` when the user has no preference and `none` when the user has not specified a goal for the slot. The current dialogue state can be predicted either from the previous turn using the previous dialogue state S_{t-1} as context, or from the entire dialogue history, predicting a new dialogue state S_t at each turn.

The dialogue state representation as (*slot, value*) pairs is used to deal with a single domain. When

[1] Another possibility is end-to-end approaches which seek to obtain a semantic representation directly from speech.

dealing with a multidomain corpus, this approach is usually extended by combining domains and slots to extract (*domain-slot*, *value*) pairs.[2]

2.2 Datasets

Many public datasets have been published to advance machine learning approaches for DST. The evolution of these datasets is marked by increasing dialogue complexity, especially with the advent of multidomain corpora. We distinguish two main approaches for collecting dialogue datasets: (i) Wizard-of-Oz (WOZ or H2H for *human-to-human*) approaches where two humans (asynchronously) play the roles of user and agent according to a task description. This approach allows for natural and varied dialogues, but the subsequent annotation can be a source of errors. (ii) Simulation-based approaches (M2M for *machine-to-machine*) where two systems play the roles of user and agent and interact with each other to generate conversation templates that are then paraphrased by humans. The advantage of this method is that the annotations are obtained automatically. However, the complexity of the task and linguistic diversity are often limited because the dialogue is simulated.

Table 1 lists the main datasets as well as recent datasets relevant to the problems discussed in Section 4. In addition to the datasets mentioned, there are other less recent corpora listed in the survey of Balaraman et al. (2021) that we do not include here for the sake of clarity. What follows is a description of the main datasets. Recent datasets, namely SMCalFlow[3] (Andreas et al., 2020), ABCD[4] (Chen et al., 2021), SIMMC 2.0[5] (Kottur et al., 2021a), BiToD[6] (Lin et al., 2021d) and DSTC10[7] (Kim et al., 2021), will be discussed in Section 4.

DSTC2[8] (Henderson et al., 2014a) Early editions of the Dialogue State Tracking Challenge (DSTC) introduced the first shared datasets and evaluation metrics for DST and thus catalyzed research in this area. The corpus of the second edition remains a reference today. It includes dialogues between paid participants and various telephone dialogue systems (H2M collection). The user needs to find a restaurant by specifying constraints such as the type of cuisine and can request specific information such as the phone number.

MultiWOZ[9] (Budzianowski et al., 2018) The first large-scale multidomain corpus and currently the main benchmark for DST. It contains dialogues between a tourist and a travel clerk that can span several domains. A major problem related to the way the data was collected is the inconsistency and errors of annotation, which was crowdsourced. Four more versions were later released to try and fix these errors. (Eric et al., 2019; Zang et al., 2020; Han et al., 2021; Ye et al., 2021a). Multilingual versions[10] were obtained by a process of machine translation followed by manual correction (Gunasekara et al., 2020; Zuo et al., 2021).

SGD[11] (Rastogi et al., 2020b) The Schema-Guided Dataset was created to elicit research on domain independence through the use of schemas. Schemas describe domains, slots, and intents in natural language and can be used to handle unseen domains. The test set includes unseen schemas to encourage model generalization. SGD-X is an extension designed to study model robustness to different schema wordings (Lee et al., 2021b).

2.3 Evaluation Metrics

As there is a strict correspondence between dialogue history and dialogue state, DST models' performance is measured with accuracy metrics. Two metrics introduced by Williams et al. (2013) are commonly used for joint and individual evaluation.

Joint goal accuracy (JGA) Main metric referring to the set of user goals. JGA indicates the performance of the model in correctly predicting the dialogue state at a given turn. It is equivalent to the proportion of turns where all predicted values for all slots exactly match the reference values. A similar metric is the average goal accuracy which only evaluates predictions for slots present in the reference dialogue state (Rastogi et al., 2020a).

[2] For simplicity, in the rest of this paper, we use the term "slot" to refer to "domain-slots"

[3] https://microsoft.github.io/task_oriented_dialogue_as_dataflow_synthesis/

[4] https://github.com/asappresearch/abcd

[5] https://github.com/facebookresearch/simmc2

[6] https://github.com/HLTCHKUST/BiToD

[7] https://github.com/alexa/alexa-with-dstc10-track2-dataset

[8] https://github.com/matthen/dstc

[9] https://github.com/budzianowski/multiwoz

[10] Mandarin, Korean, Vietnamese, Hindi, French, Portuguese, and Thai.

[11] https://github.com/google-research-datasets/dstc8-schema-guided-dialogue

	DSTC2	MultiWOZ	SGD	SMCalFlow	ABCD	SIMMC2.0	BiToD	DSTC10
Language(s)	en	en, 7 lang.*	en, ru, ar, id, sw	en	en	en	en, zh	en
Collection	H2M	H2H	M2M	H2H	H2H	M2M	M2M	H2H
Modality	speech	text	text	text	text	text, image	text	speech‡
No. domains	1	7	16	4	30	1	5	3
No. dialogues	1612	8438	16,142	41,517	8034	11,244	5787	107
No. turns	23,354	115,424	329,964	155,923	177,407	117,236	115,638	2292
Avg. turns / dial.	14.5	13.7	20.4	8.2	22.1	10.4	19.9	12.6
Avg. tokens / turn	8.5	13.8†	9.7†	10.2	9.2	12.8	12.2†	17.8
Dialogue state	slot-val	slot-val	slot-val	program	action	slot-val	slot-val	slot-val
Slots	8	25	214	-	231	12	68	18
Values	212	4510	14,139	-	12,047	64	8206	327

Table 1: Characteristics of the main (left) and recent (right) datasets available for dialogue state tracking. *Machine translated. † Average for English. ‡ In the form of ASR hypotheses.

Slot accuracy Unlike JGA, this metric evaluates the predicted value of each slot individually at each turn. It is computed as a macro-averaged accuracy: $SA = \frac{\sum_i^n acc_i}{n}$, where n represents the number of slots. For a more detailed evaluation, it can be decomposed according to slot type (e.g. requested slots, typically measured with the F1 score).

Alternative metrics Though these two metrics are widely used, they can be difficult to interpret. Slot accuracy tends to overestimate performance as most slots are not mentioned in a given turn while JGA can be too penalizing as a single wrong value prediction will result in wrong dialogue states for the rest of the conversation when accumulated through subsequent turns. Two new metrics have been recently proposed to complement existing metrics and address these shortcomings: Relative slot accuracy ignores unmentioned slots and rewards the model for correct predictions (Kim et al., 2022). Flexible goal accuracy is a generalized version of JGA which is more tolerant of errors that stem from an earlier turn (Dey et al., 2022).

2.4 Modern Approaches

One feature that categorizes DST models is the way they predict slot values. The prediction can be made either from a predefined set of values (*fixed ontology*) or from an open set of values (*open vocabulary*). What follows is a description of these approaches. A selection of recent models is presented in Table 2 based on this taxonomy.

Fixed ontology Following the discriminative approaches that preceded them, traditional neural approaches are based on a fixed ontology and treat DST as a multiclass classification problem (Henderson et al., 2014b; Mrkšić et al., 2017). Predictions for a given slot are estimated by a probability distribution over a predefined set of values, restricting the prediction field to a closed vocabulary and thus simplifying the task considerably. The performance of this approach is therefore relatively high (Chen et al., 2020), however, its cost is proportional to the size of the vocabulary as all potential values have to be evaluated. In practice, the number of values can be large and a predefined ontology is rarely available.

Open vocabulary To overcome these limitations, approaches to predict on an open set of values have been proposed. The first method consists in extracting values directly from the dialogue history, e.g. by formulating DST as a reading comprehension task (Gao et al., 2019). This method depends solely on the dialogue context to extract value spans, however, slot values can be implicit or have different wordings (e.g. the value "expensive" may be expressed as "high-end"). An alternative is to generate slot values using an encoder-decoder architecture. For instance, TRADE uses a copy mechanism to generate a value for each slot based on a representation of the dialogue history (Wu et al., 2019). A common current approach is to decode the dialogue state using a pretrained autoregressive language model (Hosseini-Asl et al., 2020).

Hybrid methods A trade-off seems to exist between the level of value independence in a model and DST performance. Some works have sought to combine fixed ontology approaches with open vocabulary prediction to benefit from the advantages of both methods. This approach is based on the distinction between categorical slots for which a set of values is predefined, and non-categorical slots with an open set of values (Goel et al., 2019; Zhang et al., 2020; Heck et al., 2020).

Model	Decoder	Context	Extra supervision	Ontology	MWOZ2.1
TRADE (Wu et al., 2019)	Generative	Full history	-	✗	45.60
TOD-BERT (Wu et al., 2020)	Classifier	Full history	Pretraining	✓	48.00
NADST (Le et al., 2020)	Generative	Full history	-	✗	49.04
DS-DST (Zhang et al., 2020)	Extract. + classif.	Previous turn	-	Cat. slots	51.21
SOM-DST (Kim et al., 2020)	Generative	History + prev. state	-	✗	52.57
MinTL (Lin et al., 2020)	Generative	History + prev. state	Response generation	✗	53.62
SST (Chen et al., 2020)	Classifier	Prev. turn and state	Schema graph	✓	55.23
TripPy (Heck et al., 2020)	Extractive	Full history	-	✗	55.30
SimpleTOD (Hosseini-Asl et al., 2020)	Generative	Full history	TOD tasks	✗	55.76
Seq2Seq-DU (Feng et al., 2021)	Generative	Full history	Schema	Cat. slots	56.10
SGP-DST (Lee et al., 2021a)	Generative	Full history	Schema	Cat. slots	56.66
SOLOIST (Peng et al., 2021a)	Generative	Full history	Pretraining	✗	56.85
PPTOD (Su et al., 2022)	Generative	Full history	Pretrain + TOD tasks	✗	57.45
D3ST (Zhao et al., 2022)	Generative	Full history	Schema	✗	57.80
TripPy + SCoRe (Yu et al., 2021)	Extractive	Full history	Pretraining	✗	60.48
TripPy + CoCo (Li et al., 2021a)	Extractive	Full history	Data augmentation	✗	60.53
TripPy + SaCLog (Dai et al., 2021)	Extractive	Full history	Curriculum learning	✗	60.61
DiCoS-DST (Guo et al., 2022)	Extract. + classif.	Relevant turns	Schema graph	✓	61.02

Table 2: Characteristics of recent DST models and performance in terms of joint goal accuracy on MultiWOZ 2.1 (Eric et al., 2019). "Ontology" denotes access to a predefined set of values for each slot.

3 Recent Advances

In recent years, several important problems have been addressed, notably through the use of pretrained language models (PLMs) trained on large amounts of unannotated text. This section summarizes the advances in 2021-2022.

3.1 Modeling Slot Relationships

The methods mentioned so far treat slots individually without taking into account their relations. However, slots are not conditionally independent, for instance, slot values can be correlated, e.g. hotel stars and price range. An alternative is to explicitly consider these relations using self-attention (Ye et al., 2021b). In a similar vein, Lin et al. (2021a) adopt a hybrid architecture to enable sequential value prediction from a GPT-2 model while modeling the relationships between slots and values with a graph attention network (GAT).

Rather than learning these relationships automatically, another line of work uses the knowledge available from the domain ontology, for example by taking advantage of its hierarchical structure (Li et al., 2021c). These relationships can also be represented in a graph with slots and domains as nodes. Chen et al. (2020) first build a schema graph based on the ontology and then use a GAT to merge information from the dialogue history and the schema graph. Feng et al. (2022) extend this approach by dynamically updating slot relations in the schema graph based on the dialogue context. Guo et al. (2022) incorporate both dialogue turns and slot-value pairs as nodes to consider relevant turns only and solve implicit mentions.

3.2 Adapting PLMs to Dialogues

Though now commonly used for DST, existing PLMs are pretrained on free-form text using language modeling objectives. Their ability to model dialogue context and multi-turn dynamics is therefore limited. It has been shown that adapting a PLM to the target domain or task by continuing self-supervised learning can lead to performance gains (Gururangan et al., 2020). This method has been applied to TOD systems and DST.

There are two underlying questions with this approach: the selection of adaptation data and the formulation of self-supervised training objectives to learn better dialogue representations for the downstream task. Wu et al. (2020) gather nine TOD corpora and continue BERT's pretraining with masked language modeling and next response selection. The obtained model TOD-BERT provides an improvement over a standard BERT model on several TOD tasks including DST. With a similar setup, Zhu et al. (2021) contrast these results and find that such adaptation is most beneficial when little annotated data is available. Based on TOD-BERT, Hung et al. (2022) show that it is advantageous not only to adapt a PLM to dialogues but also to the target domain. To do so, they use conversational data from Reddit filtered to contain terms specific to the target domain. Finally, Yu et al. (2021) introduce two objective functions designed to inject inductive biases into a PLM in order to jointly represent

dynamic dialogue utterances and ontology structure. They evaluate their method on conversational semantic parsing tasks including DST.

3.3 Mitigating Annotated Data Scarcity

The lack of annotated data hinders the development of efficient and robust DST models. However, the data collection process is costly and time-consuming. One approach to address this problem is to train a model on resource-rich domains and apply it to an unseen domain with little or no annotated data (cross-domain transfer; Wu et al., 2019). Dingliwal et al. (2021) adopt meta-learning and use the source domains to meta-learn the model's parameters and initialize fine-tuning for the target domain. Works around schema-based datasets (Rastogi et al., 2020a) use slot descriptions to handle unseen domains and slots (Lin et al., 2021c; Zhao et al., 2022). A drawback of these approaches is that they rely on the similarity between the unseen domain and the initial fine-tuning domains.

Another set of approaches tries to exploit external knowledge from other tasks with more abundant resources. Hudeček et al. (2021) use FrameNet semantic analysis as weak supervision to identify potential slots. Gao et al. (2020); Li et al. (2021b); Lin et al. (2021b) propose different methods to pretrain a model on reading comprehension data before applying it to DST. Similarly, (Shin et al., 2022) reformulate DST as a dialogue summarization task based on templates and leverage external annotated data.

Note that the PLM adaptation approaches seen above allow for more efficient learning when little data is available and are also a potential solution to the data scarcity problem. Along this line, Mi et al. (2021) present a self-learning method complementary to TOD-BERT for few-shot DST.

3.4 Prompting Generative Models to Address Unseen Domains

A recent paradigm tackles all text-based language tasks with a single model by converting them into a text-to-text format (Raffel et al., 2020). This approach relies on textual instructions called prompts that are prepended to the input to condition the model to perform a task. It has been successfully applied to DST, not only closing the performance gap between classification and generation methods but also opening opportunities to address unseen domains using schemas (cfr. Section 2). In addition to slot and domain names, Lee et al. (2021a) include slot descriptions in a prompt to independently generate values for each slot. They also add possible values for categorical slots. Zhao et al. (2021, 2022) expand on this approach by generating the entire dialogue state sequentially, as illustrated in Figure 2. In an analysis of prompt formulation, Cao and Zhang (2021) find that a question format yields better results for prompt-based DST.

Hand-crafting textual prompts may lead to sub-optimal conditioning of a PLM. Instead of using discrete tokens, we can optimize these instructions as continuous embeddings, a method called prompt tuning (Lester et al., 2021), which has been applied to DST for continual learning, adding new domains through time (Zhu et al., 2022). Alternatively, Yang et al. (2022) reverse description-based prompts and formulate a prompt based on values extracted from the utterance to generate their respective slot. They argue that slots that appear in a small corpus do not represent all potential requirements whereas values are often explicitly mentioned. Rather than using descriptions that indirectly convey the semantics of a schema, others have sought to prompt a model with instructions, i.e. in-context learning, in which the prompt consists of a few example input-output pairs (Hu et al., 2022; Gupta et al., 2022).

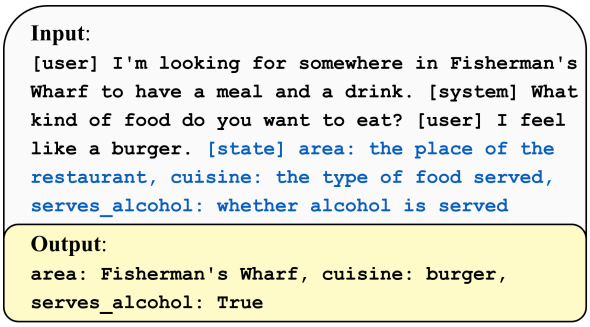


Figure 2: An example of text-to-text DST from SGD (Rastogi et al., 2020b). Including slot descriptions as a prompt makes it possible to address unseen domains.

3.5 Going Beyond English

Until recently, most works around DST were confined to English due to the lack of data in other languages, preventing the creation of truly multilingual models. In recent years, several works have addressed this issue. A DSTC9 track studied cross-lingual DST. For this purpose, a Chinese version of MultiWOZ and an English version of CrossWOZ were obtained by a process of machine translation followed by manual correction (Gunasekara et al., 2020). Zuo et al. (2021) used the same method to

translate MultiWOZ in seven languages.

A problem with machine translations is that they lack naturalness and are not localized. Two Chinese datasets were obtained by H2H collection: CrossWOZ and RiSAWOZ (Zhu et al., 2020; Quan et al., 2020), however, this type of collection is expensive. The M2M approach makes it possible to obtain adequate multilingual corpora by adapting conversation templates according to the target language. Lin et al. (2021d) took advantage of this method to create BiToD, a bilingual English-Chinese corpus. Majewska et al. (2022) used schemas from SGD to create conversation templates that were then adapted into Russian, Arabic, Indonesian and Swahili. Similarly, Ding et al. (2022) translated templates from the MultiWOZ test set in Chinese, Spanish and Indonesian and localized them. Lastly, to overcome the lack of multilingual data, Moghe et al. (2021) leverage parallel and conversational movie subtitle data to pretrain a cross-lingual DST model.

4 Challenges and Future Directions

Despite recent advances, there are still many challenges in developing DST models that can accurately capture user needs dynamically and in a variety of scenarios. There are many interesting paths, this section articulates the needs around three axes: generalization, robustness and relevance of models.

4.1 Generalizability

TOD systems are intended to be deployed in dynamic environments that may involve different settings. In practice, the application domains of these systems are numerous and varied (e.g. customer service in telecommunications, banking, technical support, etc.), which makes manual annotation of corpora for each domain difficult or impossible. This reduces the effectiveness of traditional supervised learning and is one of the reasons why most systems in production are rule-based.

Learning with little or no new annotated data offers an alternative to take advantage of the capabilities of neural networks to guarantee the flexibility of the systems. The importance of this aspect is reflected by the numerous recent works that address this problem, as presented in Section 3. Although significant progress has been made, these approaches remain limited. Models that rely on existing DST resources are unable to handle domains whose distribution deviates from that of the training data (Dingliwal et al., 2021). Models that use external knowledge offer a more generic approach but achieve relatively poor performance (Lin et al., 2021b). Learning generalizable models thus remains an open problem. An interesting avenue is continual learning, which allows new skills to be added to a system over time after deployment. Without retraining with all the data, the model must be able to accumulate knowledge (Madotto et al., 2021; Liu et al., 2021; Zhu et al., 2022).

In a real scenario, entities that are not observed during training are bound to appear. A DST model must be able to extrapolate from similar entities that have been seen. However, MultiWOZ has been shown to exhibit an entity bias, i.e. the slot values distribution is unbalanced. When a generative DST model was evaluated on a new test set with unseen entities, its performance dropped sharply, hinting at severe memorization (Qian et al., 2021). Similarly, by its constrained nature, a TOD system may be perturbed by out-of-domain utterances. It is desirable to be able to recognize such utterances in order to provide an appropriate response. The ability of a model to generalize to new scenarios is therefore related to its robustness.

4.2 Robustness

Since users may express similar requirements in different ways, a DST model must be able to interpret different formulations of a request in a consistent manner. In other words, it must be robust to variations in the input. Analytical work has shown that models' performance drops when they are confronted with realistic examples that deviate from the test set distribution (Huang et al., 2021; Li et al., 2021a). Recent work has tried to address this issue through regularization techniques (Heck et al., 2022). Related to this, an understudied aspect is dealing with utterances that deviate from the norm in the case of a written dialogue system.

A DST model must be able to take into account all the history and adjust its predictions of the dialogue state using all available information. Many works have found that performance degrades rapidly as the dialogue length increases. Another critical aspect is therefore efficiently processing long dialogues (Zhang et al., 2021). The dialogue state condenses important information, but correcting an error made in an earlier turn may be difficult. To overcome this error propagation issue, Tian et al. (2021a) use a two-pass dialogue state generation

to correct potential errors, while Manotumruksa et al. (2021) propose a turn-based objective function to penalize the model for incorrect prediction in early turns. Despite this need, Jakobovits et al. (2022) have shown that popular DST datasets are not conversational: most utterances can be parsed in isolation. Some works simulate longer and more realistic dialogues by inserting chit-chat into TODs (Kottur et al., 2021b; Sun et al., 2021).

Another point that has not been studied much in recent approaches is robustness to speech inputs (Faruqui and Hakkani-Tür, 2021). For spoken dialogue systems, new challenges arise such as ASR errors or verbal disfluencies. Early editions of DSTC provided spoken corpora that included a transcript and ASR hypotheses. Since then, DST datasets have been primarily text-based. A DSTC10 track considered this aspect again and proposed a DST task with validation and test sets containing ASR hypotheses (Kim et al., 2021).

Learning robust models requires diverse datasets that represent real-world challenges. In this sense, several evaluation benchmarks have been published to study TOD systems' robustness (Lee et al., 2021b; Peng et al., 2021b; Cho et al., 2021). Data augmentation is a potential solution to the lack of variety in datasets (Campagna et al., 2020; Li et al., 2021a; Aksu et al., 2022), especially for simulating ASR errors (Wang et al., 2020; Tian et al., 2021b).

4.3 Relevance

As we have seen, existing datasets do not really reflect real-world conditions resulting in a rather artificial task. It is important to keep a holistic view of the development of DST models to ensure the relevance of their application in a dialogue system.

Since the first TOD systems, the dialogue state has been considered as a form to be filled in as slot-value pairs. This fixed representation is suitable for simple tasks like flight booking but is limited in domains with rich relational structures and a variable number of entities. Indeed, composition is not possible (e.g. "itinerary for my next meeting") and knowledge is not directly shared between slots. Section 3 presented approaches that attempt to address the latter point by using graphs for dialogue state representation. To promote work on more realistic scenarios, some have proposed richer representations with an associated corpus. Andreas et al. (2020) encode the dialogue state as a dataflow graph and introduce the SMCallFlow corpus. Cheng et al. (2020) propose a tree structure along with the TreeDST corpus. ThingTalk is another alternative representation of the dialogue state which was successfully applied to MultiWOZ (Lam et al., 2022; Campagna et al., 2022). In the ABCD corpus, Chen et al. (2021) adopt a representation of the procedures that a customer service employee must follow in accordance with company policies.

These approaches still rely on specific database schemas and are limited to one modality. For more capable virtual agents, we can extend the scope of dialogues to a multimodal world. Emerging efforts on multimodal DST seek to track the information of visual objects based on a multimodal context. With SIMMC 2.0, Kottur et al. (2021a) introduced a multimodal TOD dataset based on virtual reality rendered shopping scenes. Similarly, Le et al. (2022) proposed a synthetic dataset of dialogues consisting of multiple question-answer pairs about a grounding video input. The questions are based on 3D objects. In both these datasets, the system has to track visual objects in the dialogue state in the form of slot-value pairs. For a broader background on multimodal conversational AI, we refer the reader to Sundar and Heck (2022)'s survey.

Dialogue is dynamic: in a real scenario, an erroneous prediction of the dialogue state would have deviated the course of the conversation from the reference dialogue. However, most studies evaluate models in isolation, assuming that it is always possible to assemble a set of well-performing components to build a good TOD system. The overall performance of a system is rarely taken into account as evaluating the system as a whole is complicated and requires human evaluation. Moreover, it can be difficult to identify which component of the system is problematic and needs to be improved. Despite these hurdles, it is important to consider the impact that DST can have on the dialogue system as a whole. Takanobu et al. (2020) conducted automatic and human evaluations of dialogue systems with a wide variety of configurations and settings on MultiWOZ. They found a drop in task success rate using DST rather than NLU followed by a rule-based dialogue state update. They explain this result by the fact that NLU extracts the user's intentions in addition to the slot-value pairs. Another work showed how uncertainty estimates in belief tracking can lead to a more robust downstream policy (van Niekerk et al., 2021). These studies are rare in their kind and call for more similar work.

5 Conclusion

Dialogue state tracking is a crucial component of a conversational agent to identify the user's needs at each turn of the conversation. A growing body of work is addressing this task and we have outlined the latest developments. After giving an overview of the task and the different datasets available, we have categorized modern neural approaches according to the inference of the dialogue state. Despite encouraging results on benchmarks such as MultiWOZ, these systems lack flexibility and robustness, which are critical skills for a dialogue system. In recent years, many works have sought to address these limitations and we have summarized the advances. However, there are still significant challenges to be addressed in the future. There are many interesting avenues and we have proposed three key features of DST models to guide future research: generalizability, robustness and relevance.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and suggestions for further improvement.

References

Ibrahim Aksu, Zhengyuan Liu, Min-Yen Kan, and Nancy Chen. 2022. N-shot learning for augmenting task-oriented dialogue state tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1659–1671, Dublin, Ireland. Association for Computational Linguistics.

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-Oriented Dialogue as Dataflow Synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Vevake Balaraman, Seyedmostafa Sheikhalishahi, and Bernardo Magnini. 2021. Recent Neural Methods on Dialogue State Tracking for Task-Oriented Dialogue Systems: A Survey. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 239–251, Singapore and Online. Association for Computational Linguistics.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Pawe\l Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica Lam. 2020. Zero-Shot Transfer Learning with Synthesized Data for Multi-Domain Dialogue State Tracking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 122–132, Online. Association for Computational Linguistics.

Giovanni Campagna, Sina Semnani, Ryan Kearns, Lucas Jun Koba Sato, Silei Xu, and Monica Lam. 2022. A Few-Shot Semantic Parser for Wizard-of-Oz Dialogues with the Precise ThingTalk Representation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4021–4034, Dublin, Ireland. Association for Computational Linguistics.

Jie Cao and Yi Zhang. 2021. A Comparative Study on Schema-Guided Dialogue State Tracking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 782–796, Online. Association for Computational Linguistics.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-Based Conversations Dataset: A Corpus for Building More In-Depth Task-Oriented Dialogue Systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020. Schema-Guided Multi-Domain Dialogue State Tracking with Graph Attention Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7521–7528.

Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, Jason D. Williams, Hong Yu, Diarmuid Ó Séaghdha, and Anders Johannsen. 2020. Conversational Semantic Parsing

for Dialog State Tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8107–8117, Online. Association for Computational Linguistics.

Hyundong Cho, Chinnadhurai Sankar, Christopher Lin, Kaushik Ram Sadagopan, Shahin Shayandeh, Asli Celikyilmaz, Jonathan May, and Ahmad Beirami. 2021. CheckDST: Measuring Real-World Generalization of Dialogue State Tracking Performance. *arXiv:2112.08321 [cs]*.

Yinpei Dai, Hangyu Li, Yongbin Li, Jian Sun, Fei Huang, Luo Si, and Xiaodan Zhu. 2021. Preview, Attend and Review: Schema-Aware Curriculum Learning for Multi-Domain Dialogue State Tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 879–885, Online. Association for Computational Linguistics.

Suvodip Dey, Ramamohan Kummara, and Maunendra Desarkar. 2022. Towards Fair Evaluation of Dialogue State Tracking by Flexible Incorporation of Turn-level Performances. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 318–324, Dublin, Ireland. Association for Computational Linguistics.

Bosheng Ding, Junjie Hu, Lidong Bing, Mahani Aljunied, Shafiq Joty, Luo Si, and Chunyan Miao. 2022. GlobalWoZ: Globalizing MultiWoZ to Develop Multilingual Task-Oriented Dialogue Systems. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1639–1657, Dublin, Ireland. Association for Computational Linguistics.

Saket Dingliwal, Shuyang Gao, Sanchit Agarwal, Chien-Wei Lin, Tagyoung Chung, and Dilek Hakkani-Tur. 2021. Few Shot Dialogue State Tracking using Meta-learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1730–1739, Online. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines. *arXiv:1907.01669 [cs]*.

Manaal Faruqui and Dilek Hakkani-Tür. 2021. Revisiting the Boundary between ASR and NLU in the Age of Conversational Dialog Systems. *arXiv:2112.05842 [cs, eess]*.

Yue Feng, Aldo Lipani, Fanghua Ye, Qiang Zhang, and Emine Yilmaz. 2022. Dynamic Schema Graph Fusion Network for Multi-Domain Dialogue State Tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 115–126, Dublin, Ireland. Association for Computational Linguistics.

Yue Feng, Yang Wang, and Hang Li. 2021. A Sequence-to-Sequence Approach to Dialogue State Tracking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1714–1725, Online. Association for Computational Linguistics.

Shuyang Gao, Sanchit Agarwal, Di Jin, Tagyoung Chung, and Dilek Hakkani-Tur. 2020. From Machine Reading Comprehension to Dialogue State Tracking: Bridging the Gap. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 79–89, Online. Association for Computational Linguistics.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog State Tracking: A Neural Reading Comprehension Approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Rahul Goel, Shachi Paul, and Dilek Hakkani-Tür. 2019. HyST: A Hybrid Approach for Flexible and Accurate Dialogue State Tracking. In *Interspeech 2019*, pages 1458–1462. ISCA.

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, Dilek Hakkani-Tür, Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Minlie Huang, Jianfeng Gao, Shikib Mehri, Yulan Feng, Carla Gordon, Seyed Hossein Alavi, David Traum, Maxine Eskenazi, Ahmad Beirami, Eunjoon, Cho, Paul A. Crook, Ankita De, Alborz Geramifard, Satwik Kottur, Seungwhan Moon, Shivani Poddar, and Rajen Subba. 2020. Overview of the Ninth Dialog System Technology Challenge: DSTC9. *arXiv:2011.06486 [cs]*.

Jinyu Guo, Kai Shuang, Jijie Li, Zihan Wang, and Yixuan Liu. 2022. Beyond the Granularity: Multi-Perspective Dialogue Collaborative Selection for Dialogue State Tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2320–2332, Dublin, Ireland. Association for Computational Linguistics.

Raghav Gupta, Harrison Lee, Jeffrey Zhao, Abhinav Rastogi, Yuan Cao, and Yonghui Wu. 2022. Show, Don't Tell: Demonstrations Outperform Descriptions for Schema-Guided Task-Oriented Dialogue. *arXiv:2204.04327 [cs]*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey,

and Noah A. Smith. 2020. Don't Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2021. MultiWOZ 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. *arXiv:2010.05594 [cs]*.

Michael Heck, Nurul Lubis, Carel van Niekerk, Shutong Feng, Christian Geishauser, Hsien-Chin Lin, and Milica Gašić. 2022. Robust Dialogue State Tracking with Weak Supervision and Sparse Data. *arXiv:2202.03354 [cs]*.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Matthew Henderson. 2015. Machine Learning for Dialog State Tracking: A Review. In *Proceedings of The First International Workshop on Machine Learning in Spoken Language Processing*.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014a. The Second Dialog State Tracking Challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Steve Young. 2014b. Word-Based Dialog State Tracking with Recurrent Neural Networks. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 292–299, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A Simple Language Model for Task-Oriented Dialogue. In *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc.

Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-Context Learning for Few-Shot Dialogue State Tracking. *arXiv:2203.08568 [cs]*.

Minlie Huang, Xiaoyan Zhu, and Jianfeng Gao. 2020. Challenges in Building Intelligent Open-domain Dialog Systems. *ACM Transactions on Information Systems*, 38(3):21:1–21:32.

Yi Huang, Junlan Feng, Xiaoting Wu, and Xiaoyu Du. 2021. Counterfactual Matters: Intrinsic Probing For Dialogue State Tracking. In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 1–6, Online. Association for Computational Linguistics.

Vojtěch Hudeček, Ondřej Dušek, and Zhou Yu. 2021. Discovering Dialogue Slots with Weak Supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2430–2442, Online. Association for Computational Linguistics.

Chia-Chien Hung, Anne Lauscher, Simone Ponzetto, and Goran Glavaš. 2022. DS-TOD: Efficient Domain Specialization for Task-Oriented Dialog. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 891–904, Dublin, Ireland. Association for Computational Linguistics.

Alice Shoshana Jakobovits, Francesco Piccinno, and Yasemin Altun. 2022. What Did You Say? Task-Oriented Dialog Datasets Are Not Conversational!? *arXiv:2203.03431 [cs]*.

Seokhwan Kim, Yang Liu, Di Jin, A. Papangelis, Karthik Gopalakrishnan, Behnam Hedayatnia, and Dilek Z. Hakkani-Tür. 2021. "How Robust R U?": Evaluating Task-Oriented Dialogue Systems on Spoken Conversations. *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient Dialogue State Tracking by Selectively Overwriting Memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Takyoung Kim, Hoonsang Yoon, Yukyung Lee, Pilsung Kang, and Misuk Kim. 2022. Mismatch between Multi-turn Dialogue and its Evaluation Metric in Dialogue State Tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 297–309, Dublin, Ireland. Association for Computational Linguistics.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021a. SIMMC 2.0: A Task-oriented Dialog Dataset for Immersive Multimodal Conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Satwik Kottur, Chinnadhurai Sankar, Zhou Yu, and Alborz Geramifard. 2021b. DialogStitch: Synthetic Deeper and Multi-Context Task-Oriented Dialogs. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*,

pages 21–26, Singapore and Online. Association for Computational Linguistics.

Monica S. Lam, Giovanni Campagna, Mehrad Moradshahi, Sina J. Semnani, and Silei Xu. 2022. ThingTalk: An Extensible, Executable Representation Language for Task-Oriented Dialogues.

Hung Le, Nancy Chen, and Steven Hoi. 2022. Multimodal dialogue state tracking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3394–3415, Seattle, United States. Association for Computational Linguistics.

Hung Le, Richard Socher, and Steven C. H. Hoi. 2020. Non-Autoregressive Dialog State Tracking. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021a. Dialogue State Tracking with a Language Model using Schema-Driven Prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2021b. SGD-X: A Benchmark for Robust Generalization in Schema-Guided Dialogue Systems. *arXiv:2110.06800 [cs]*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021a. CoCo: Controllable Counterfactuals for Evaluating Dialogue State Trackers. *arXiv:2010.12850 [cs]*.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian McAuley. 2021b. Zero-shot Generalization in Dialog State Tracking through Generative Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1063–1074, Online. Association for Computational Linguistics.

Xinmeng Li, Qian Li, Wansen Wu, and Quanjun Yin. 2021c. Generation and Extraction Combined Dialogue State Tracking with Hierarchical Ontology Integration. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2249, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Weizhe Lin, Bo-Hsiang Tseng, and Bill Byrne. 2021a. Knowledge-Aware Graph-Enhanced GPT-2 for Dialogue State Tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7871–7881, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaojiang Lin, Bing Liu, Andrea Madotto, Seungwhan Moon, Zhenpeng Zhou, Paul Crook, Zhiguang Wang, Zhou Yu, Eunjoon Cho, Rajen Subba, and Pascale Fung. 2021b. Zero-Shot Dialogue State Tracking via Cross-Task Transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7890–7900, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021c. Leveraging Slot Descriptions for Zero-Shot Cross-Domain Dialogue StateTracking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5640–5648, Online. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, and Pascale Fung. 2020. MinTL: Minimalist Transfer Learning for Task-Oriented Dialogue Systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3391–3405, Online. Association for Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Genta Indra Winata, Peng Xu, Feijun Jiang, Yuxiang Hu, Chen Shi, and Pascale Fung. 2021d. BiToD: A Bilingual Multi-Domain Dataset For Task-Oriented Dialogue Modeling. In *Thirty-Fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Qingbin Liu, Pengfei Cao, Cao Liu, Jiansong Chen, Xunliang Cai, Fan Yang, Shizhu He, Kang Liu, and Jun Zhao. 2021. Domain-Lifelong Learning for Dialogue State Tracking via Knowledge Preservation Networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2301–2311, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, Pascale Fung, and Zhiguang Wang. 2021. Continual Learning in Task-Oriented Dialogue Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7452–7467, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Olga Majewska, Evgeniia Razumovskaia, Edoardo Maria Ponti, Ivan Vulić, and Anna Korhonen. 2022. Cross-Lingual Dialogue Dataset Creation via Outline-Based Generation. *arXiv:2201.13405 [cs]*.

Jarana Manotumruksa, Jeff Dalton, Edgar Meij, and Emine Yilmaz. 2021. Improving Dialogue State Tracking with Turn-based Loss Function and Sequential Data Augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1674–1683, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fei Mi, Wanhao Zhou, Lingjing Kong, Fengyu Cai, Minlie Huang, and Boi Faltings. 2021. Self-training Improves Pre-training for Few-shot Learning in Task-oriented Dialog Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1887–1898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nikita Moghe, Mark Steedman, and Alexandra Birch. 2021. Cross-lingual Intermediate Fine-tuning improves Dialogue State Tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1137–1150, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural Belief Tracker: Data-Driven Dialogue State Tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Vancouver, Canada. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021a. SOLOIST: Building Task Bots at Scale with Transfer Learning and Machine Teaching. *arXiv:2005.05298 [cs]*.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2021b. RADDLE: An Evaluation Benchmark and Analysis Platform for Robust Task-oriented Dialog Systems. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4418–4429, Online. Association for Computational Linguistics.

Kun Qian, Ahmad Beirami, Zhouhan Lin, Ankita De, Alborz Geramifard, Zhou Yu, and Chinnadhurai Sankar. 2021. Annotation Inconsistency and Entity Bias in MultiWOZ. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 326–337, Singapore and Online. Association for Computational Linguistics.

Jun Quan, Shian Zhang, Qian Cao, Zizhong Li, and Deyi Xiong. 2020. RiSAWOZ: A Large-Scale Multi-Domain Wizard-of-Oz Dataset with Rich Semantic Annotations for Task-Oriented Dialogue Modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 930–940, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020a. Schema-Guided Dialogue State Tracking Task at DSTC8. *arXiv:2002.01359 [cs]*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020b. Towards Scalable Multi-Domain Conversational Agents: The Schema-Guided Dialogue Dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.

Jamin Shin, Hangyeol Yu, Hyeongdon Moon, Andrea Madotto, and Juneyoung Park. 2022. Dialogue Summaries as Dialogue States (DS2), Template-Guided Summarization for Few-shot Dialogue State Tracking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3824–3846, Dublin, Ireland. Association for Computational Linguistics.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-Task Pre-Training for Plug-and-Play Task-Oriented Dialogue System. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676, Dublin, Ireland. Association for Computational Linguistics.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2021. Adding Chit-Chat to Enhance Task-Oriented Dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1570–1583, Online. Association for Computational Linguistics.

Anirudh Sundar and Larry Heck. 2022. Multimodal conversational AI: A survey of datasets and approaches. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 131–147, Dublin, Ireland. Association for Computational Linguistics.

Ryuichi Takanobu, Qi Zhu, Jinchao Li, Baolin Peng, Jianfeng Gao, and Minlie Huang. 2020. Is Your Goal-Oriented Dialog Model Performing Really Well? Empirical Analysis of System-wise Evaluation. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 297–310, 1st virtual meeting. Association for Computational Linguistics.

Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He, Yunyi Yang, Hua Wu, Fan Wang, and Shuqi Sun. 2021a. Amendable Generation for Dialogue State Tracking. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 80–92, Online. Association for Computational Linguistics.

Xin Tian, Xinxian Huang, Dongfeng He, Yingzhan Lin, Siqi Bao, Huang He, Liankai Huang, Qiang Ju, Xiyuan Zhang, Jian Xie, Shuqi Sun, Fan Wang, Hua Wu, and Haifeng Wang. 2021b. TOD-DA: Towards Boosting the Robustness of Task-oriented Dialogue Modeling on Spoken Conversations. *arXiv:2112.12441 [cs]*.

Carel van Niekerk, Andrey Malinin, Christian Geishauser, Michael Heck, Hsien-chin Lin, Nurul Lubis, Shutong Feng, and Milica Gasic. 2021. Uncertainty Measures in Neural Belief Tracking and the Effects on Dialogue Policy Performance. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7901–7914, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Longshaokan Wang, Maryam Fazel-Zarandi, Aditya Tiwari, Spyros Matsoukas, and Lazaros Polymenakos. 2020. Data Augmentation for Training Dialog Models Robust to Speech Recognition Errors. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 63–70, Online. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The Dialog State Tracking Challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 404–413, Metz, France. Association for Computational Linguistics.

Jason D. Williams, Antoine Raux, and Matthew Henderson. 2016. The Dialog State Tracking Challenge Series: A Review. *Dialogue & Discourse*, 7(3):4–33.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained Natural Language Understanding for Task-Oriented Dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable Multi-Domain State Generator for Task-Oriented Dialogue Systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Yuting Yang, Wenqiang Lei, Juan Cao, Jintao Li, and Tat-Seng Chua. 2022. Prompt Learning for Few-Shot Dialogue State Tracking. *arXiv:2201.05780 [cs]*.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. 2021a. MultiWOZ 2.4: A Multi-Domain Task-Oriented Dialogue Dataset with Essential Annotation Corrections to Improve State Tracking Evaluation. *arXiv:2104.00773 [cs]*.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021b. Slot Self-Attentive Dialogue State Tracking. In *Proceedings of the Web Conference 2021*, WWW '21, pages 1598–1608, New York, NY, USA. Association for Computing Machinery.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D. Williams. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proceedings of the IEEE*, 101(5):1160–1179.

Tao Yu, Rui Zhang, Alex Polozov, Chris Meek, and Ahmed H. Awadallah. 2021. SCoRe: Pre-Training for Context Representation in Conversational Semantic Parsing. In *International Conference on Learning Representations*.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A Dialogue Dataset with Additional Annotation Corrections and State Tracking Baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Find or Classify? Dual Strategy for Slot-Value Predictions on Multi-Domain Dialog State Tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

Ye Zhang, Yuan Cao, Mahdis Mahdieh, Jeffrey Zhao, and Yonghui Wu. 2021. Improving Longer-range Dialogue State Tracking. *arXiv:2103.00109 [cs]*.

Jeffrey Zhao, Raghav Gupta, Yuan Cao, Dian Yu, Mingqiu Wang, Harrison Lee, Abhinav Rastogi, Izhak Shafran, and Yonghui Wu. 2022. Description-Driven Task-Oriented Dialog Modeling. *arXiv:2201.08904 [cs]*.

Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. Effective Sequence-to-Sequence Dialogue State Tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qi Zhu, Yuxian Gu, Lingxiao Luo, Bing Li, Cheng Li, Wei Peng, Minlie Huang, and Xiaoyan Zhu. 2021. When does Further Pre-training MLM Help? An Empirical Study on Task-Oriented Dialog Pre-training. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 54–61, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A Large-Scale Chinese Cross-Domain Task-Oriented Dialogue Dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

Qi Zhu, Bing Li, Fei Mi, Xiaoyan Zhu, and Minlie Huang. 2022. Continual Prompt Tuning for Dialog State Tracking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1124–1137, Dublin, Ireland. Association for Computational Linguistics.

Lei Zuo, Kun Qian, Bowen Yang, and Zhou Yu. 2021. AllWOZ: Towards Multilingual Task-Oriented Dialog Systems for All. *arXiv:2112.08333 [cs]*.

MultiWOZ 2.4: A Multi-Domain Task-Oriented Dialogue Dataset with Essential Annotation Corrections to Improve State Tracking Evaluation

Fanghua Ye
University College London
London, UK

Jarana Manotumruksa
University College London
London, UK

Emine Yilmaz
University College London
London, UK

{fanghua.ye.19, j.manotumruksa, emine.yilmaz}@ucl.ac.uk

Abstract

The MultiWOZ 2.0 dataset has greatly stimulated the research of task-oriented dialogue systems. However, its state annotations contain substantial noise, which hinders a proper evaluation of model performance. To address this issue, massive efforts were devoted to correcting the annotations. Three improved versions (i.e., MultiWOZ 2.1-2.3) have then been released. Nonetheless, there are still plenty of incorrect and inconsistent annotations. This work introduces MultiWOZ 2.4, which refines the annotations in the validation set and test set of MultiWOZ 2.1. The annotations in the training set remain unchanged (same as MultiWOZ 2.1) to elicit robust and noise-resilient model training. We benchmark eight state-of-the-art dialogue state tracking models on MultiWOZ 2.4. All of them demonstrate much higher performance than on MultiWOZ 2.1[1].

1 Introduction

In recent years, tremendous advances have been made in the research of task-oriented dialogue systems, attributed to a number of publicly available dialogue datasets like DSTC2 (Henderson et al., 2014), FRAMES (El Asri et al., 2017), WOZ (Wen et al., 2017), M2M (Shah et al., 2018), MultiWOZ 2.0 (Budzianowski et al., 2018), SGD (Rastogi et al., 2020), CrossWOZ (Zhu et al., 2020), RiSAWOZ (Quan et al., 2020), and TreeDST (Cheng et al., 2020). Among them, MultiWOZ 2.0 is the first large-scale dataset spanning multiple domains and thus has attracted the most attention.

However, substantial noise has been found in the dialogue state annotations of MultiWOZ 2.0 (Eric et al., 2020). To remedy this issue, Eric et al. (2020) fixed 32% of dialogue state annotations across 40% of the dialogue turns, resulting in an improved version MultiWOZ 2.1. Despite the significant improvement in annotation quality, MultiWOZ 2.1 still severely suffers from incorrect and inconsistent annotations (Zhang et al., 2020; Hosseini-Asl et al., 2020). The state-of-the-art joint goal accuracy (Zhong et al., 2018) for dialogue state tracking on MultiWOZ 2.1 is merely around 60% (Li et al., 2021). Even worse, the noise in the validation set and test set makes it relatively challenging to assess model performance properly and adequately. To reduce the impact of noise, different preprocessing strategies have been utilized by existing models. For example, TRADE (Wu et al., 2019) fixes some general annotation errors. SimpleTOD (Hosseini-Asl et al., 2020) cleans partial noisy annotations in the test set. TripPy (Heck et al., 2020) constructs a label map to handle value variants. These preprocessing strategies, albeit helpful, lead to an unfair performance comparison.

Massive efforts have been made to further improve the annotation quality of MultiWOZ 2.1, resulting in MultiWOZ 2.2 (Zang et al., 2020) and MultiWOZ 2.3 (Han et al., 2021). However, they both have some limitations. More concretely, MultiWOZ 2.2 allows the presence of multiple values in the dialogue state. But it does not cover all the value variants. This incompleteness brings about serious inconsistencies. MultiWOZ 2.3 focuses on dialogue act annotations. The noise on dialogue state annotations has not been fully resolved.

In this work, we introduce MultiWOZ 2.4, an updated version on top of MultiWOZ 2.1, to improve dialogue state tracking evaluation. Specifically, we identify incorrect and inconsistent annotations in the validation set and test set, and fix them meticulously. This refinement results in changes to the state annotations of more than 41% of turns over 65% of dialogues. Since our main purpose is to improve the correctness and fairness of model evaluation, the annotations in the training set remain unchanged. Even so, our empirical study shows that much better performance can be achieved on MultiWOZ 2.4 than on all the previous versions.

[1]MultiWOZ 2.4 is released to the public at https://github.com/smartyfh/MultiWOZ2.4.

Proceedings of the SIGdial 2022 Conference, pages 351–360
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Error Type	Conversation Example	MultiWOZ 2.1	MultiWOZ 2.4
(I) Context Mismatch	**Usr:** Hello, I would like to book a taxi from restaurant 2 two to the museum of classical archaeology.	taxi-destination=museum of archaelogy and anthropology	taxi-destination=museum of classical archaeology
	Usr: I am looking for a restaurant that serves Portuguese food.	rest.-food=Portugese	rest.-food=Portuguese
(II) Missing Annotation	**Usr:** I need a place to dine in the centre of town.	rest.-area=none	rest.-area=centre
	Usr: Please recommend one and book it for 6 people. **Sys:** I would recommend express by holiday inn Cambridge. From what day should I book? **Usr:** Starting Saturday. I need 5 nights for 6 people.	hotel-book people=none hotel-book people=6	hotel-book people=6 hotel-book people=6
(III) Not Mentioned	**Usr:** I am planning a trip in Cambridge.	hotel-internet=dontcare	hotel-internet=none
(IV) Incomplete Value	**Sys:** I recommend Charlie Chan. Would you like a table? **Usr:** Yes. Monday, 8 people, 10:30.	rest.-name=Charlie	rest.-name=Charlie Chan
	Usr: Something classy nearby for dinner, preferably Italian or Indian cuisine?	rest.-food=Indian	rest.-food=Indian|Italian
(V) Implicit Time Processing	**Usr:** I need a train leaving after 10:00.	train-leaveat=10:15	train-leaveat=10:00
(VI) Unnecessary Annotation	**Usr:** I am looking for a museum. **Sys:** The Broughton house gallery is a museum in the centre. **Usr:** That sounds good. Could I get their phone number?	attraction-area=centre	attraction-area=none

Figure 1: Examples of each error type. Only the problematic slots are presented. "rest." is short for restaurant.

Furthermore, a noisy training set motivates us to design robust and noise-resilient training mechanisms, e.g., data augmentation (Summerville et al., 2020) and noisy label learning (Han et al., 2020). Considering that collecting noise-free large multi-domain dialogue datasets is costly and labor-intensive, we believe that training robust dialogue state tracking models from noisy training data will be of great interest to both industry and academia.

2 Annotation Refinement

In MultiWOZ 2.0 & 2.1, the dialogue state is represented as a series of *slot-value* pairs. For example, *attraction-area=centre* means that the slot is *attraction-area* and its value is *centre*. Considering that MultiWOZ 2.1 has significantly improved the annotation quality of MultiWOZ 2.0, we choose to continue the refinement on the basis of MultiWOZ 2.1. Another choice is to perform the refinement on top of MultiWOZ 2.2. However, as mentioned earlier, MultiWOZ 2.2 allows each slot to have multiple value variants. This relaxation increases the difficulty of annotating. It is challenging to include all the value variants. New value variants may also emerge as time goes by. Even worse, some value variants are ambiguous and invalid. For instance, "Peking" can be a shared variant of "Peking University" and "Peking restaurant". Hence, it is an ambiguous value variant. Besides, the benchmark evaluation on MultiWOZ 2.2 shows no evident performance improvements over MultiWOZ 2.1 (Zang et al., 2020). In light of these, MultiWOZ 2.1 is a better basis for our refinement.

2.1 Annotation Error Types

The main goal of dialogue state tracking is to track what has been uttered by a user. Thus, it is generally assumed that the dialogue state should mainly rely on user utterances[2]. Based on this assumption, we identify and fix six types of annotation errors in the validation set and test set of MultiWOZ 2.1. Figure 1 shows examples for each error type.

Context Mismatch: The slot value is inconsistent with the one mentioned in the dialogue context. We also include values with typos in this error type.

Missing Annotation: The slot is unlabelled, even though its value has been mentioned. In some cases, the annotations are delayed to later turns.

Not Mentioned: The slot has been annotated, however, its value has not been mentioned at all.

Incomplete Value: The slot value is a substring or an abbreviation of its full shape (e.g., "Thurs" vs. "Thursday"). In some cases, the slot should have multiple values, but not all values are included.

Implicit Time Processing: This relates to the slots that take time as the value. Instead of copying the time specified in the dialogue context, the value has been implicitly processed (e.g., adding 15 min)[3].

[2] If the user requirements cannot be satisfied (e.g., a restaurant asked by the user does not exist), the system should still track the "wrong" requirements as the dialogue state and then ask a clarification question (Doğan et al., 2022) to the user.

[3] The value is implicitly processed when the time is after or before a certain point. Albeit reasonable, it is hard to decide the exact time offset. Thus, we copy the specified time directly.

Refinement Type	Count	Ratio(%)
no change	432,972	97.90
none→value	3,230	0.73
valueA/dontcare→valueB	1,598	0.36
value/dontcare→none	2,846	0.64
none/value→dontcare	1,614	0.36

Table 1: The count and ratio of slot values changed in MultiWOZ 2.4 compared with MultiWOZ 2.1.

Unnecessary Annotation: These unnecessary annotations exacerbate inconsistencies as different annotators have different opinions on whether to annotate these slots or not. In general, the values of these slots are mentioned by the system to respond to previous user requests or provide supplementary information. We found that in most dialogues, these slots are not annotated. Hence, we remove these annotations. However, the name-related slots are an exception. If the user requests more information (e.g., *address* and *postcode*) about the recommended "name", the slots will be annotated.

2.2 Annotation Refinement Procedure

The validation set and test set of MutliWOZ 2.1 contain 2,000 dialogues with more than 14,000 dialogue turns. These dialogues span over 5 domains with a total of 30 slots. To guarantee that the refined annotations are as correct and consistent as possible, we decided to rectify the annotations by ourselves rather than crowd-workers. However, if we check the annotations of all 30 slots at each turn, the workload is too heavy. To ease the burden, we instead only checked the annotations of turn-active slots. A slot being turn-active means that its value is determined by the dialogue context of current turn and is not inherited from previous turns. The average number of turn-active slots in the original annotations and in the refined annotations is 1.16 and 1.18, respectively. The full dialogue state is then obtained by accumulating all turn-active states from the first turn to current turn.

We also observed that some slot values are mentioned in different forms, such as "concert hall" vs. "concerthall" and "guest house" vs. "guest houses". The name-related slot values may have a word the at the beginning, e.g., "Peking restaurant" vs. "the Peking restaurant". We normalized these variants by selecting the one with the highest frequency. In addition, all time-related slot values have been updated to the 24:00 format. We performed the above refining process twice to reduce mistakes and it took us one month to finish this task.

Dataset	Slot(%)	Turn(%)	Dialogue(%)
val	5.04	42.61	67.40
test	5.17	39.74	64.16
total	5.10	41.17	65.78

Table 2: The ratio of refined slots, turns and dialogues.

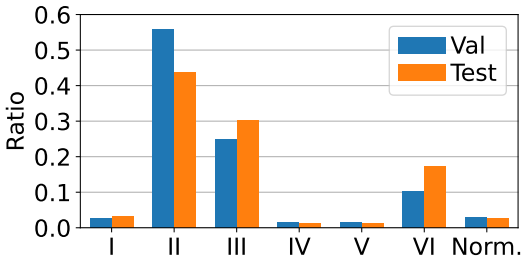


Figure 2: The ratio of different error types. "Norm." refers to values normalized based on their frequency.

2.3 Statistics on Refined Annotations

Table 1 shows the count and percentage of slot values changed in MultiWOZ 2.4 compared with MultiWOZ 2.1. Note that none and dontcare are regarded as two special values. As can be seen, most slot values remain unchanged. This is because a dialogue only has a few active slots and the other slots always take the value none. Table 2 further reports the ratio of refined slots, turns and dialogues. Here, the ratio of refined slots is computed on the basis of refined turns. It is shown that the corrected states relate to more than 41% of turns over 65% of dialogues. On average, the annotations of 1.53 ($30 \times 5.10\%$) slots at each refined turn have been rectified.

Figure 2 illustrates the distribution of different error types. We also treat unnormalized values (cf. §2.2) as a special type of errors. Figure 2 shows that "*Missing Annotation*" and "*Not Mentioned*" are the two most frequent error types. It also shows that more than 10% of errors are related to "*Unnecessary Annotation*", while the other types of errors only account for a relatively small proportion.

3 Benchmark Evaluation

3.1 Benchmark Models

Existing neural dialogue state tracking models can be roughly divided into two categories: predefined ontology-based methods and open vocabulary-based methods. The ontology-based methods perform classification by scoring all possible slot-value pairs in the ontology and selecting the value with the highest score as the prediction. By contrast, the open vocabulary-based methods directly generate or extract slot values from the dialogue

	Model	Joint Goal Accuracy (%)			Slot Accuracy (%)	
		MWZ 2.1 Test	MWZ 2.4 Test	MWZ 2.4 Val	MWZ 2.1 Test	MWZ 2.4 Test
predefined ontology	SUMBT	49.01	61.86 (+12.85)	62.31	96.76	97.90
	STAR	**56.36**	**73.62** (+17.26)	74.59	**97.59**	**98.85**
open vocabulary	TRADE	45.60	55.05 (+9.45)	57.01	96.55	97.62
	PIN	48.40	58.92 (+10.52)	60.37	97.02	98.02
	SOM-DST	51.24	**66.78** (+15.54)	68.77	97.15	**98.38**
	SimpleTOD	51.75	57.18 (+5.43)	55.02	96.78	96.97
	SAVN	54.86	60.55 (+5.69)	61.91	**97.55**	98.05
	TripPy	**55.18**	64.75 (+9.57)	64.27	97.48	98.33

Table 3: Joint goal accuracy and slot accuracy of different models on MultiWOZ 2.1 and MultiWOZ 2.4.

Dataset	SUMBT (%)	TRADE (%)
MultiWOZ 2.0	48.81	48.62
MultiWOZ 2.1	49.01	45.60
MultiWOZ 2.2	49.70	46.60
MultiWOZ 2.3	52.90	49.20
MultiWOZ 2.3-cof	54.60	49.90
MultiWOZ 2.4	**61.86**	**55.05**

Table 4: Comparison of test set joint goal accuracy on different versions of the MultiWOZ dataset.

Domain	SOM-DST (%)		STAR (%)	
	2.1	2.4	2.1	2.4
attraction	69.83	83.22	70.95	84.45
hotel	49.53	64.52	52.99	69.10
restaurant	65.72	77.67	69.17	84.20
taxi	59.96	54.76	66.67	73.63
train	70.36	82.73	75.10	90.36

Table 5: Comparison of domain-specific test set joint goal accuracy.

context. We benchmark the performance of our refined dataset on both types of methods, including SUMBT (Lee et al., 2019), STAR (Ye et al., 2021), TRADE (Wu et al., 2019), PIN (Chen et al., 2020), SOM-DST (Kim et al., 2020), SimpleTOD (Hosseini-Asl et al., 2020), SAVN (Wang et al., 2020), and TripPy (Heck et al., 2020).

3.2 Benchmark Results

We adopt joint goal accuracy (Zhong et al., 2018) and slot accuracy as evaluation metrics. The joint goal accuracy is defined as the ratio of dialogue turns in which all slot values are correctly predicted. The slot accuracy is defined as the average accuracy of all slots. As shown in Table 3, all models achieve much higher performance on MultiWOZ 2.4. SimpleTOD shows the least performance improvement. The reason may be that SimpleTOD generates state values directly while other methods such as TRADE leverage the copy mechanism (See et al., 2017) to assist in the generation process. SAVN also shows a low performance increase, as it has already utilized value normalization to tackle label variants in MultiWOZ 2.1. We then report the joint goal accuracy of SUMBT and TRADE on different versions of the dataset in Table 4, in which MultiWOZ 2.3-cof means MultiWOZ 2.3 with coreference applied. As can be seen, both methods perform better on MultiWOZ 2.4 than on all previous versions. We include the domain-specific accuracy of SOM-DST and STAR in Table 5, which shows that except SOM-DST in the *taxi* domain, both methods demonstrate higher performance in each domain of MultiWOZ 2.4.

4 Human Evaluation

We also perform a human evaluation on the quality of the refined annotations. We randomly sampled 50 dialogues from the test set and recruited 5 computer science students to compare our refinement against the annotations in MultiWOZ 2.1. Specifically, the raters were asked to assign a score to each turn of the sampled dialogues based on the following criteria: 1) **-2**: A score of -2 means that both the refined annotation and original annotation are not completely correct; 2) **-1**: A score of -1 means that the original annotation is correct while the refined annotation is problematic; 3) **0**: A score of 0 means that both the refined annotation and original annotation are correct, that is, no changes have been made to the original annotation; 4) **1**: A score of 1 means that the refined annotation is correct while the original annotation is invalid.

We obtain an average score of 0.1653, meaning that our refined annotations are more accurate. We further employ Fleiss' kappa (Fleiss, 1971) to measure the level of agreement among different raters. We obtain $\kappa = 0.9226$, which indicates an almost perfect agreement across the five raters.

We illustrate the score distributions of different raters in Figure 3. From this figure, we can intuitively observe that there is a high level of agree-

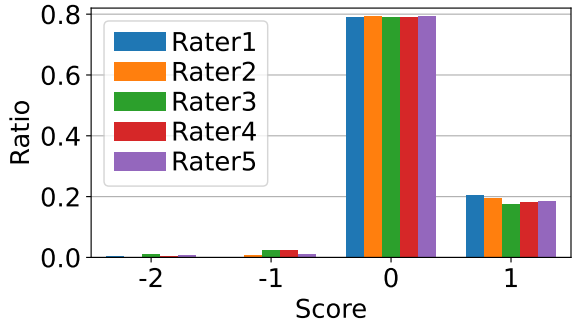


Figure 3: The score distribution of different raters.

ment among the five raters. Figure 3 also shows that in most cases, the refined annotation and the original annotation are both correct, meaning that there is no need to make any changes to the original annotation. This is desirable, as our refinement is based on MultiWOZ 2.1 which has already fixed lots of annotation errors. Around 20% of annotations in MultiWOZ 2.4 are deemed to be more accurate than MultiWOZ 2.1, while only about 1% of annotations in MultiWOZ 2.1 are evaluated as better. This verifies again that our refinement has higher quality.

We further inspected the annotations in MultiWOZ 2.1 that are assessed to be more appropriate. We found that these annotations are mainly related to the slot *hotel-type*. This slot has four candidate values {"hotel", "guest house", "none", "dontcare"}, which are relatively confusing because the term "hotel" is also one candidate value. In practice, when a user says "I am looking for a hotel with 4 stars", the user may actually mean that "I am looking for a place to stay with 4 stars". However, by convention, the term "hotel" is used more often, even though the user does not mean that the hotel type must be "hotel". In our refinement procedure, we chose to annotate this slot based on the whole dialogue session to understand the true user intention (i.e., *hotel type=hotel*?) while the raters tended to take into account only the dialogue history. This ambiguous slot tells us that it is crucial to develop appropriate slots and candidate values that will not cause any confusions to the annotators.

5 Caveats and Lessons Learned

Although we have tried our best to correct as many annotations in the validation set and test set as possible, it is unlikely that we have fixed all the annotation errors. In fact, there are several challenges we faced during the refinement process that are particularly difficult to overcome. Firstly, as discussed earlier, the candidate values of some slots are confusing, which makes it really challenging to choose the most appropriate value. Secondly, in some scenarios, the user intention can have different interpretations. For example, the user utterance "the hotel does not need to have internet though" can mean that the user does not need internet at all (*hotel-internet=no*) or the user does not care about if the internet is provided (*hotel-internet=dontcare*). Thirdly, some slots may have multiple values. Sometimes these values should even be ordered according to users' preferences. When there are too many values (more than two), it is also questionable if the corresponding slot should be annotated. Suppose that the system recommended 10 museums to the user and the user asked "Does any of them have zero entrance fee?", should the slot *attraction-name* be annotated?

Further, the dialogue state can be regarded as a structured representation of the complex user intentions. Due to the complexity of the language itself, some information will be inevitably lost when transforming unstructured user utterances into structured state representations. In this regard, dialogue state annotating is in essence a challenging task.

Given these challenges, it is necessary to define unambiguous slots and unconfusable candidate values to facilitate state annotating. It is also important to provide annotators with full instructions for each slot so that they can make consistent annotations.

6 Conclusion

We introduce MultiWOZ 2.4, an updated version of MultiWOZ 2.1, by rectifying (almost) all the annotation errors in the validation set and test set. We keep the annotations in the training set as is to encourage robust and noise-resilient model training. We further benchmark eight state-of-the-art dialogue state tracking models on MultiWOZ 2.4 to facilitate future research. All the benchmark models have demonstrated much better performance on MultiWOZ 2.4 than on MultiWOZ 2.1.

MultiWOZ 2.4 can also be applied to train better overall dialogue systems, e.g., by utilizing data augmentation techniques to generate high-quality training data based on the clean validation set.

Acknowledgments

This project was funded by the EPSRC Fellowship titled "Task Based Information Retrieval" and grant reference number EP/P024289/1.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Junfan Chen, Richong Zhang, Yongyi Mao, and Jie Xu. 2020. Parallel interactive networks for multi-domain dialogue state generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1921–1931, Online. Association for Computational Linguistics.

Jianpeng Cheng, Devang Agrawal, Héctor Martínez Alonso, Shruti Bhargava, Joris Driesen, Federico Flego, Dain Kaplan, Dimitri Kartsaklis, Lin Li, Dhivya Piraviperumal, Jason D. Williams, Hong Yu, Diarmuid Ó Séaghdha, and Anders Johannsen. 2020. Conversational semantic parsing for dialog state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8107–8117, Online. Association for Computational Linguistics.

Fethiye Irmak Doğan, Ilaria Torre, and Iolanda Leite. 2022. Asking follow-up clarifications to resolve ambiguities in human-robot conversation. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, pages 461–469.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: a corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 207–219, Saarbrücken, Germany. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. 2020. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*.

Ting Han, Ximing Liu, Ryuichi Takanabu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2021. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 206–218. Springer.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gasic. 2020. TripPy: A triple copy strategy for value independent neural dialog state tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44, 1st virtual meeting. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason D. Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 263–272, Philadelphia, PA, U.S.A. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582, Online. Association for Computational Linguistics.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483, Florence, Italy. Association for Computational Linguistics.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2021. Coco: Controllable counterfactuals for evaluating dialogue state trackers. *arXiv preprint arXiv:2010.12850*.

Jun Quan, Shian Zhang, Qian Cao, Zizhong Li, and Deyi Xiong. 2020. RiSAWOZ: A large-scale multi-domain Wizard-of-Oz dataset with rich semantic annotations for task-oriented dialogue modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 930–940, Online. Association for Computational Linguistics.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51, New Orleans - Louisiana. Association for Computational Linguistics.

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Adam Summerville, Jordan Hashemi, James Ryan, and William Ferguson. 2020. How to tame your data: Data augmentation for dialog state tracking. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 32–37, Online. Association for Computational Linguistics.

Yexiang Wang, Yi Guo, and Siqi Zhu. 2020. Slot attention with value normalization for multi-domain dialogue state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3019–3028, Online. Association for Computational Linguistics.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 808–819, Florence, Italy. Association for Computational Linguistics.

Fanghua Ye, Jarana Manotumruksa, Qiang Zhang, Shenghui Li, and Emine Yilmaz. 2021. Slot self-attentive dialogue state tracking. In *Proceedings of the Web Conference 2021*, pages 1598–1608.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. MultiWOZ 2.2 : A dialogue dataset with additional annotation corrections and state tracking baselines. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 109–117, Online. Association for Computational Linguistics.

Jianguo Zhang, Kazuma Hashimoto, Chien-Sheng Wu, Yao Wang, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 154–167, Barcelona, Spain (Online). Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2018. Global-locally self-attentive encoder for dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1458–1467, Melbourne, Australia. Association for Computational Linguistics.

Qi Zhu, Kaili Huang, Zheng Zhang, Xiaoyan Zhu, and Minlie Huang. 2020. CrossWOZ: A large-scale Chinese cross-domain task-oriented dialogue dataset. *Transactions of the Association for Computational Linguistics*, 8:281–295.

A Additional Statistics on the Refined Annotations

In Table 6, we report the value vocabulary size (i.e., the number of candidate values) of each slot in MultiWOZ 2.1 & 2.4, respectively. We also report their value change ratios. As can be observed, for some slots, the value vocabulary size decreases due to value normalization and error correction. For some slots, the value vocabulary size increases mainly because a few labels that contain multiple values have been additionally introduced. Table 6 also indicates that the `name`-related slots have the highest value change ratio. Since these slots usually have "longer" values, the annotators are more likely to make incomplete and inconsistent annotations.

Slot	2.1	2.4	Val(%)	Test(%)
attraction-area	7	8	1.97	1.93
attraction-name	106	92	5.34	5.16
attraction-type	17	23	4.62	3.77
hotel-area	7	8	3.92	3.99
hotel-book day	8	8	0.33	0.52
hotel-book people	9	9	0.68	0.53
hotel-book stay	6	7	0.42	0.42
hotel-internet	5	4	2.32	2.24
hotel-name	48	46	6.28	3.95
hotel-parking	5	4	2.54	2.35
hotel-pricerange	6	6	1.76	2.06
hotel-stars	8	10	1.52	1.44
hotel-type	5	4	5.06	4.78
rest.-area	7	8	2.18	2.38
rest.-book day	8	11	0.35	0.27
rest.-book people	9	9	0.37	0.45
rest.-book time	59	62	0.56	0.46
rest.-food	89	93	2.58	2.28
rest.-name	135	121	7.81	5.90
rest.-pricerange	5	7	1.51	2.05
taxi-arriveby	62	61	0.41	0.56
taxi-departure	177	172	0.92	0.86
taxi-destination	185	181	1.14	0.75
taxi-leaveat	92	89	0.84	0.45
train-arriveby	109	73	1.40	2.86
train-book people	11	12	1.22	1.76
train-day	8	9	0.31	0.24
train-departure	19	15	0.71	1.10
train-destination	20	17	0.71	1.00
train-leaveat	128	96	4.64	5.12

Table 6: The slot value vocabulary size counted on the validation set and test set of MultiWOZ 2.1 and MultiWOZ 2.4, respectively, and the slot-specific value change ratio. "rest." is the abbreviation of restaurant.

B Per-Slot (Slot-Specific) Accuracy

In Section 3, we have presented the joint goal accuracy and average slot accuracy of eight state-of-the-art dialogue state tracking models. The results have demonstrated that much better performance can be achieved on our refined annotations in terms of the two metrics. Here, we further report the per-slot (slot-specific) accuracy of SUMBT on different versions of the MultiWOZ dataset. The slot-specific accuracy is defined as the ratio of dialogue turns in which the value of a particular slot has been correctly predicted. The results are shown in Table 7, from which we can observe that the majority of slots (21 out of 30) demonstrate higher accuracies on MultiWOZ 2.4. Even though MultiWOZ 2.3-cof additionally introduces the co-reference annotations as a kind of auxiliary information, it still only shows the best performance in 7 slots. Compared with MultiWOZ 2.1, SUMBT has achieved higher slot-specific accuracies in 26 slots on MultiWOZ 2.4. These results confirm again the utility and validity of our refined version MultiWOZ 2.4.

C Case Study

Except for the quantitative analyses provided in the benchmark evaluation and human evaluation, we also conduct a qualitative analysis to understand more intuitively why and how the refined annotations boost the performance of evaluation. To this end, we showcase several dialogues from the test set in Table 8, where we include the annotations of MultiWOZ 2.1 and MultiWOZ 2.4 and also the predictions of SOM-DST and STAR. It is easy to check that the annotations of MultiWOZ 2.1 are incorrect, while the annotations of MultiWOZ 2.4 are consistent with the dialogue context and therefore are valid. From Table 8, we also observe that the predictions of both SOM-DST and STAR are the same as the annotations of MultiWOZ 2.4 in the first four dialogues. In the last dialogue, the prediction of STAR is consistent with the annotation of MultiWOZ 2.4, whereas the predicted slot value of SOM-DST is different from the annotations of both MultiWOZ 2.1 and MultiWOZ 2.4. These examples show that the performance of existing dialogue state tracking models is underestimated because of the invalid annotations in MultiWOZ 2.1. While MultiWOZ 2.4 can better manifest the true model performance owing to the refined annotations that align well with the dialogue context.

Slot	MultiWOZ 2.1	MultiWOZ 2.2	MultiWOZ 2.3	MultiWOZ 2.3-cof	MultiWOZ 2.4
attraction-area	95.94	95.97	96.28	**96.80**	96.38
attraction-name	93.64	93.92	95.28	94.59	**96.38**
attraction-type	96.76	97.12	96.53	96.91	**98.24**
hotel-area	94.33	94.44	94.65	95.02	**96.16**
hotel-book day	98.87	99.06	99.04	99.32	**99.52**
hotel-book people	98.66	98.72	98.93	99.17	**99.19**
hotel-book stay	99.23	99.50	99.70	99.70	**99.88**
hotel-internet	97.02	97.02	97.45	97.56	**97.96**
hotel-name	94.67	93.76	94.71	94.71	**96.92**
hotel-parking	97.04	97.19	97.90	98.34	**98.68**
hotel-pricerange	96.00	96.23	95.90	96.40	**96.59**
hotel-stars	97.88	97.95	97.99	98.09	**99.16**
hotel-type	94.67	94.22	**95.92**	95.65	94.75
restaurant-area	96.30	95.47	95.52	96.05	**97.52**
restaurant-book day	98.90	98.91	98.83	**99.66**	98.59
restaurant-book people	98.91	98.98	99.17	99.21	**99.31**
restaurant-book time	99.43	99.24	99.31	**99.46**	99.28
restaurant-food	97.69	97.61	97.49	97.64	**98.71**
restaurant-name	92.71	93.18	95.10	94.91	**96.01**
restaurant-pricerange	95.36	95.65	95.75	96.26	**96.59**
taxi-arriveby	98.36	98.03	98.18	**98.45**	98.17
taxi-departure	96.13	96.35	96.15	**97.49**	96.55
taxi-destination	95.70	95.50	95.56	**97.59**	95.68
taxi-leaveat	98.91	98.96	**99.04**	99.02	98.72
train-arriveby	96.40	96.40	96.54	96.76	**98.85**
train-book people	97.26	97.04	97.29	97.67	**98.62**
train-day	98.63	98.60	99.04	**99.38**	98.94
train-departure	98.43	98.40	97.56	97.50	**99.32**
train-destination	98.55	98.30	97.96	97.86	**99.43**
train-leaveat	93.64	94.14	93.98	93.96	**96.96**

Table 7: Per-slot (slot-specific) accuracy (%) of SUMBT on different versions of the MultiWOZ dataset. The results on MultiWOZ 2.1-2.3 and MultiWOZ 2.3-cof are from (Han et al., 2021). It is shown that most slots demonstrate stronger performance on MultiWOZ 2.4 than on all the other versions.

D Discussion

Recall that in MultiWOZ 2.4, we only refined the annotations of the validation set and test set. The annotations in the training set remain unchanged (the same as MultiWOZ 2.1). As a result, all the benchmark models are retrained on the original noisy training set. The only difference is that we use the cleaned validation set to choose the best model and then report the results on the cleaned test set. Even so, we have shown in our empirical study that the benchmark models can obtain better performance on MultiWOZ 2.4 than on all the previous versions. Considering that all the previous refined versions also corrected the (partial) annotation errors in the training set, the superiority of MultiWOZ 2.4 indicates that existing versions have not fully resolved the incorrect and inconsistent annotations. Therefore, although there have been three refined versions, our refinement is still necessary and meaningful. In addition, the refined validation set and test set can be combined with the training set of MultiWOZ 2.3. Since MultiWOZ 2.3 has the cleanest training set by far, this combination has the potential to result in even higher performance of existing methods.

On the other hand, it is well-understood that deep (neural) models are data-hungry. However, it is costly and labor-intensive to collect high-quality large-scale datasets, especially dialogue datasets

Dialogue ID	Dialogue Context, Groundtruth Annotations, and Predictions of SOM-DST and STAR			
PMUL1931	**Sys:** We have 6 different guest houses that fit your criteria. Do you have a specific price range in mind? **Usr:** No, it does not matter.			
	MultiWOZ 2.1	MultiWOZ 2.4	SOM-DST	STAR
	hotel-pricerange none	*hotel-pricerange dontcare*	*hotel-pricerange dontcare*	*hotel-pricerange dontcare*
PMUL3158	**Usr:** I want to find a place in town to visit called jesus green outdoor pool.			
	MultiWOZ 2.1	MultiWOZ 2.4	SOM-DST	STAR
	attraction-type swimming pool	*attraction-type none*	*attraction-type none*	*attraction-type none*
MUL1489	**Sys:** Ok, you are all set for cote on Friday, table for 8 at 17:30. Can I help with anything else? **Usr:** Can I have the reference number for the reservation please? **Sys:** Booking was unsuccessful. Can you try another time slot? **Usr:** What about 16:30?			
	MultiWOZ 2.1	MultiWOZ 2.4	SOM-DST	STAR
	restaurant-book time 17:30	*restaurant-book time 16:30*	*restaurant-book time 16:30*	*restaurant-book time 16:30*
PMUL0550	**Sys:** I recommend Charlie Chan. Would you like to reserve a table? **Usr:** Yes. Monday, 8 people, 10:30.			
	MultiWOZ 2.1	MultiWOZ 2.4	SOM-DST	STAR
	restaurant-name Charlie	*restaurant-name Charlie Chan*	*restaurant-name Charlie Chan*	*restaurant-name Charlie Chan*
MUL1697	**Sys:** I am sorry none of them have booking available for that time, another time maybe? **Usr:** Is 09:45 an available time?			
	MultiWOZ 2.1	MultiWOZ 2.4	SOM-DST	STAR
	restaurant-book time 21:45	*restaurant-book time 09:45*	*restaurant-book time 10:45*	*restaurant-book time 09:45*

Table 8: Examples of test set dialogues in which the annotations of MultiWOZ 2.1 are incorrect but the predictions of SOM-DST and STAR are correct (except the prediction of SOM-DST in the last example), as the predicted slot values are consistent with the dialogue context. Given that the annotations of MultiWOZ 2.4 are consistent with the dialogue context as well, there is no doubt that higher performance can be achieved when performing evaluation on MultiWOZ 2.4. Note that only the problematic slots are presented.

that involve multiple domains and multiple turns. The dataset composed of a large noisy training set and a small clean validation set and test set is more common in practice. In view of this, our refined dataset is a better reflection of the realistic situation we encounter in our daily life. Moreover, a noisy training set may motivate us to design more robust and noise-resilient training paradigms. As a matter of fact, noisy label learning (Han et al., 2020; Song et al., 2022) has been widely studied in the machine learning community to train robust models from noisy training data. Numerous advanced techniques have been investigated as well. We hope to see that these techniques can also be applied to the study of dialogue systems and thus accelerate the development of conversational AI.

E Potential Impacts

We believe that our refined dataset MultiWOZ 2.4 would have substantial impacts in academia. First of all, the cleaned validation set and test set can help us evaluate the performance of dialogue state tracking models more properly and fairly, which is undoubtedly beneficial to the research of task-oriented dialogue systems. In addition, MultiWOZ 2.4 may also serve as a potential dataset to assist the research of noisy label learning in the machine learning community. The advantage of MultiWOZ 2.4 is that it is a multi-label dataset with real noise in the training set. In the machine learning community, it has been recognized as a future research direction to study noisy label learning for multi-label classification (Song et al., 2022).

The Duration of a Turn Cannot be Used to Predict When It Ends

Charles Threlkeld and **JP de Ruiter**
Tufts University
{charles.threlkeld, jp.deruiter}@tufts.edu

Abstract

Turn taking in conversation is a complex process. We still do not know how listeners are able to anticipate the end of a speaker's turn. Previous work focuses on prosodic, semantic, and non-verbal cues that a turn is coming to an end. In this paper, we look at simple measures of duration—time, word count, and syllable count—to see if we can exploit the duration of turns as a cue. We find strong evidence that these metrics are useless.

1 Introduction

Turn-taking is a fundamental aspect of dialogue. Timing of turn initiation is critical. Sometimes long pauses are socially relevant (Bogels et al., 2015). Sometimes people overlap in conversation without the reason being clear (Heldner and Edlund, 2010). When trouble occurs, people can pause to signal misunderstandings (Mertens and De Ruiter, 2021). But turn taking as a whole is not well understood.

What is known is that the time between successive turns is generally very short—much shorter than can be attributed to simple reactions to a turn ending (De Ruiter, 2019). What this means is that people must anticipate the end of a turn (Ruiter et al., 2006). If we are anticipating the end of a turn, then there must be some features of utterances that we use to predict their ending, enabling fluid turn transition.

In artificial agents that engage in spoken dialogue, turn-taking often falls by the wayside, leading to stilted conversations with long delays between turns or interruptions at inappropriate times (Skantze, 2021). Typical human interactions with current conversational agents work uniformly sequentially, as the agent processes and responds to the human once the end of an utterance has been completed, and it does not expect interruptions or overlaps (Gervits et al., 2020).

In general, computers can process information much faster than humans, but we have not yet developed fluid turn-taking algorithms. Humans prepare a one-word utterance in around 600ms (Indefrey and Levelt, 2004). Computers can perform much faster than this and their speed is still increasing. But if an agent doesn't know when a turn ends, fluidity can be compromised. For smooth turn taking, agents need to know how to time their contributions appropriately.

Previous research looks at lexical (Magyari and de Ruiter, 2012), semantic (Gervits et al., 2020; Riest et al., 2015), prosodic (Bögels and Torreira, 2015), or non-verbal (Roddy et al., 2018) attributes of utterances in order to anticipate turn ends. Each of these has its own merits and drawbacks. Lexical boundaries are relatively easy to compute and reason about. Semantic completion of an utterance makes logical sense for an end-point to a thought. Prosodic cues can be computed quickly from the speech signal, and non-verbal cues are ripe for deep learning techniques (Lala et al., 2019). Turn *duration*, however, has not been studied yet for its use as a cue in anticipating its end, despite its ready availability to any spoken dialogue system.

Intuitively, one would expect that the duration of a turn is a strong cue about its ending. It would be plausible to assume that the longer someone has been talking already, the higher the probability is that the speaker will end their turn. Compare it to waiting for a bus – we tend to assume that the longer we have waited for the bus, the higher the probability that it will finally arrive. But this is only so when the duration of a turn is normatively constrained. However, looking at distribution of a large number of conversational turns in Dutch, De Ruiter (2019) found that the distribution of turn-duration looks suspiciously much like an exponential distribution. And a unique and counter-intuitive property of this distribution is that it has a constant *hazard*

Proceedings of the SIGdial 2022 Conference, pages 361–367
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

rate: no matter how long we have waited for the process to complete, the probability of it terminating in the next instant remains constant. If turn durations are in fact exponentially distributed, it would mean that the duration of a turn so far does not contain any information about its projected duration.

However, the observation in De Ruiter (2019) were only for one small corpus in Dutch, and measured in milliseconds. It could be that measuring duration in other units, like words, syllables, or other turn-related units would show a different distribution. In this study we set out to study if this suspected property of turn durations is generalizable to a larger corpus in English, and to other units of duration.

Turns in dialogue are composed of turn construction units (TCUs). TCUs are bounded by transition relevant places (Sacks et al., 1978). At each transition relevant place, another person could take the floor or the current speaker could continue. In this study, we will investigate the duration of both TCUs and entire turns. As there may be social preferences regarding the number of TCUs within a turn, we will also examine the usefulness of the number of TCUs per turn in predicting floor transitions.

In the following sections, we outline the data collection and our statistical analyses. Then we will show the distributions of the data and the statistical models describing the data. We will then discuss the implications of our results, and present ideas on how these results can be used to improve spoken dialogue agents.

2 Methods

2.1 Dataset

For this study, we are using the Switchboard corpus (Godfrey and Holliman, 1993). The Switchboard corpus is a large, well-studied corpus of dyadic, open-ended telephone conversations. Its use limits our ability to draw conclusions about face-to-face speech patterns, but extends the work of De Ruiter (2019, p.542–543) — a study of Dutch telephone conversations — to English. Since the corpus is well-studied, we can draw on previous work for transcription, timing, and segmentation.

We used two transcriptions of the Switchboard corpus. First, the Mississippi State University transcriptions[1] were used for word-by-word timing. Second, the Discourse Language Modeling Project transcriptions[2] break the conversation into turn construction units. We are interested in TCUs as the basic building blocks of turns, and to compare that to the analysis of turn duration in De Ruiter (2019, p.542–543) which only looked at duration in seconds.

After merging these two sources, we analyzed only conversations where the word-level exact matches were at least 90% of words in a conversation, and the total error rate of the conversation (that is, words matching none of our word-matching heuristics) was below 2%. Heuristics included accounted for simple, systematic alternative transcriptions, like repeated or omitted words, alternative spellings ("uh-uh" / "uh-huh"), or abandoned words ("ho-" / "how"). The analyses use the resulting 75 conversations with 5,857 turns and 11,796 turn construction units.

2.2 Probabilistic Modeling

For each aspect of the data, we will build two models. The first model will be a best-fit exponential distribution; the second will be a best-fit gamma distribution (except for TCUs per turn; see below). We will show the curves of the data along with curves for each model so that we can quantify and visualize the differences in prediction between the models and in reference to the data. Full descriptive statistics can be found in the appendix.

We chose the exponential distribution as a null model. It is the maximum entropy distribution for positive-domain data with a known mean. It also has the property of being memoryless, or having a constant hazard rate. This means that no matter how long an exponential process has been ongoing, the chance that it will end in the next time step is constant. This makes it a good null model, as there is very little information that can be gained about a distribution via the exponential distribution. Both the mean and the hazard rate are related to the single distribution parameter λ, which is the hazard rate or probability of the process ending in the next time-step. More concretely, λ is the chance of stopping at the next millisecond, word, or syllable, given that the process has not stopped so far.

The gamma distribution is a generalization of the exponential distribution. It is parameterized by

[1] http://www.openslr.org/5/

[2] Available at https://web.stanford.edu/~jurafsky/ws97/

a shape and rate parameter. If the shape parameter is one, the gamma is equivalent to the exponential distribution. Importantly, other shape parameters allow a gamma model to fit many different positive-domain datasets with different modes and varying hazard rates.

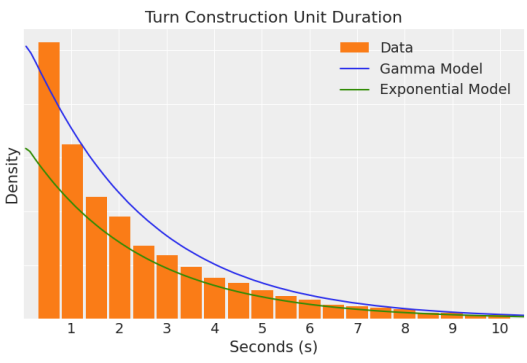
TCUs per Turn data is fit to Geometric and Negative Binomial distributions, which are the discrete forms of the Exponential and Gamma distributions, respectively. The p and n parameters of the discrete distributions mirror the $rate$ and $shape$ parameters of their continuous analogues. This decision was made because the small number of TCUs per turn does not lend itself to an assumption of continuity.

We will compare the exponential and gamma models for each dataset using the widely-applicable information criterion (WAIC). The WAIC estimates the effective number of parameters to adjust for overfitting, and gives results similar to a leave-one-out cross-validation for model-fitting. A lower WAIC is a better fit.

3 Results

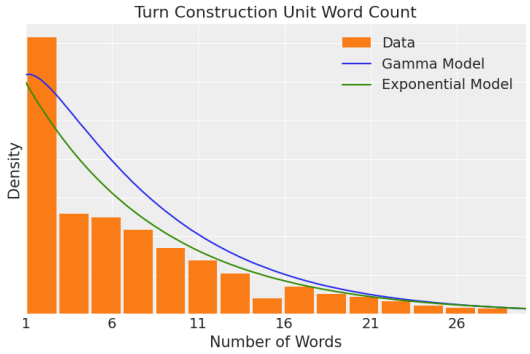
Here we will report the basic findings of our analyses. The interpretation of the findings will be delayed to discussion.

3.1 TCU Duration



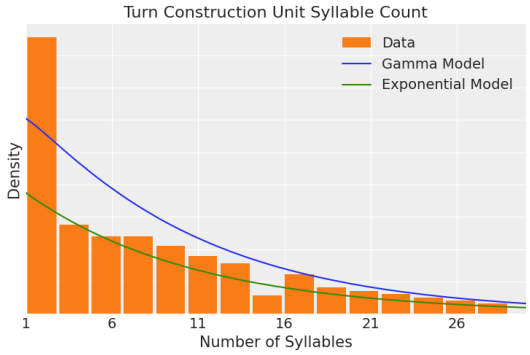
TCU duration exponential and gamma models were very similar, since the best-fit gamma model has a shape of 1.01, which is effectively the same as an exponential distribution. We can see this close fit in the WAIC scores, too, which were identical to the third decimal place.

3.2 TCU Word Count



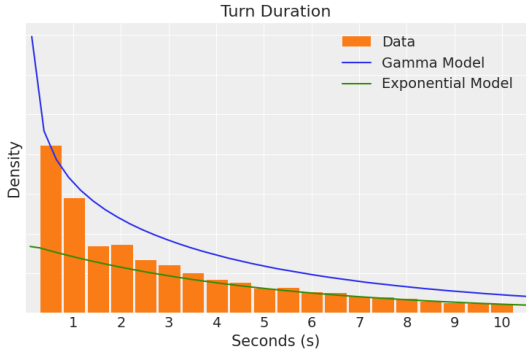
TCU word count was interesting in that it was the only model with a gamma shape parameter substantially above one, at 1.18. We can see this reflected in the concavity of the gamma model curve near zero. WAIC scores were very similar, with the exponential distribution only 0.3% higher than the gamma distribution.

3.3 TCU Syllable Count



The TCU syllable count gamma model had a shape parameter of 1.05, very nearly identical to an exponential distribution. We can see the similarity in the chart. The WAIC is again only 0.3% higher for the exponential model than the gamma model.

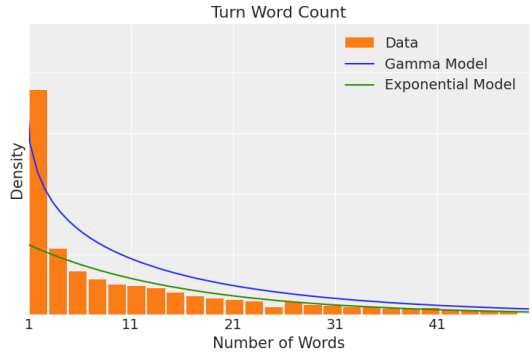
3.4 Turn Duration



In the turn duration statistics, we see a sizable difference between where the gamma model is positioned and the exponential model, but also the data. The best fit gamma pulls the curve toward the tail

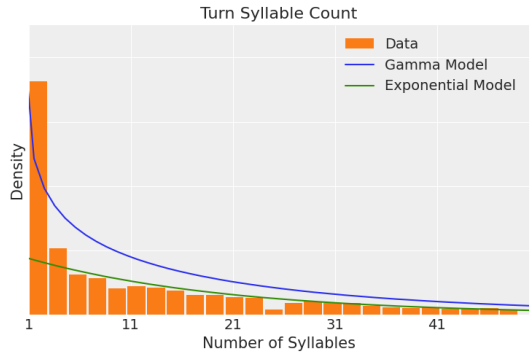
in order to accommodate the high number of fast turns in the data. Despite the different orientations to the data, the trade-off between good fit on low values or good fit at high values cancel out and the WAIC is again only 0.3% higher for the exponential model than the gamma model.

3.5 Turn Word Count



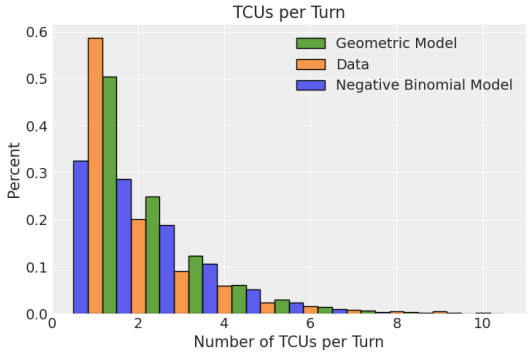
The turn word count models show similar tendencies to the turn duration models—the gamma model better accounts for low values, but the exponential distribution fits better at higher values. The low shape parameter of the gamma distribution (0.718) allows this distortion. The WAIC of the exponential distribution is 1.1% higher than the gamma.

3.6 Turn Syllable Count



Similar to the turn words model, a low gamma shape value—only 0.667—lets the gamma model account for many utterances with very few syllables compared to the expectations in the exponential distribution. Here we have our largest WAIC disparity with at 1.65% higher for the exponential model compared to the gamma model.

3.7 TCUs per Turn



TCUs per turn was fit to geometric and negative binomial models rather than exponential and gamma models. The negative binomial model has underestimated the low numbers, trading off probability mass at low counts for larger predictions at high counts. The WAIC shows that the simpler exponential model is a better fit with a 20% lower score.

4 Discussion

The analyses above, when taken together, suggest that there is little to be learned from examining the length of utterances as a sole heuristic for predicting their end. As was suspected on theoretical grounds (De Ruiter, 2019) there is very little information in simple duration. The work here extended this previous work from timing of TCus to examine semantic content as shown by word counts, phonetic information as shown by syllable counts, or social action as shown by TCUs per turn. None of these linguistics frames showed any substantial departure from the constant hazard-rate distribution.

There may be contexts where utterance length is a useful heuristic for TCU or turn end or situations in which the statistics describe here do not fit well. For example, one would suspect that different dialogue acts may lend themselves to different TCU lengths — short backchannels, for example. Or, particular social situations may lend themselves to fewer TCUs per turn to ensure participants maintain the same mental models. A follow-up study on (e.g.,) the Map Task Corpus might show these deviation, if they exist.

The largest deviation from the exponential model occurred in the turn word and syllable count analyses. These two results reflect the combination of high rates of single TCU turns and large shares of low TCU word and syllable counts. Our TCU per turn analysis shows that single-TCU turns are more common than the exponetial model is able to fit, but the negative binomial distribution moves prob-

ability mass to the tail, making it an even worse fit, both visually and statistically. Short turns skew the data in turn-level word and syllable count toward very low counts as compared to the exponential model.

We expected that the TCUs per turn negative binary model would account for the high number of single TCU turns, much like the gamma model does for the other data views. However, the geometric model outperforms the negative binomial by the largest WAIC difference of any models discussed. Therefore, we must conclude that the geometric model is the superior fit, and the best-fit hazard rate for each TCU—the chance that the speaker's turn is over at the end of any TCU—is 50%, or a coin flip. So, not only is the maximum entropy geometric model a better fit, but there is no reliable bias for whether a turn is over at the end of a TCU.

5 Conclusion

In this paper, we first confirmed the suspicions raised in De Ruiter (2019)—the duration of TCUs follows an exponential distribution. We then extended these findings in several ways. First, TCUs also follow this distribution by syllable or word count. Conversation does not orient to the amount of phonological or semantic information. It follows that if these factors are useful for turn taking, they are useful based on their meaning and structure, not their quantity or base informational load.

Next, we expanded our findings to the turn level, rather than just TCUs. Turn duration, syllable, and word count findings were akin to those at the TCU level, and so we must draw the conclusions that these turn length measurements are not useful either to exploit as information source in turn taking.

Finally, we looked at TCUs per turn for evidence that the number of dialogue acts of which a turn has numerical norms. The TCU per turn analysis showed that the end of a TCU is essentially a coin flip for whether there will be a floor transfer. Not only did the maximum entropy distribution have the best fit, but the hazard rate was very close to 0.5. So, we must conclude that there is no more pragmatic pressure to end one's turn when it is already very long.

Our general conclusion therefore is that, surprisingly, the duration of turns are not useful cues for turn segmentation or turn taking decisions. This is independent of whether we use temporal, phonological, lexical, or TCU-based measures of information. Agents that do turn taking will need to use linguistic or prosodic cues other than duration to achieve accurate timing in their turn taking behavior.

Acknowledgments

This paper was funded in part by a grant from the Data Intensive Studies Center at Tufts University.

References

Sara Bogels, Kobin H. Kendrick, and Stephen C. Levinson. 2015. Never say no ... how the brain interprets the pregnant pause in conversation. *PLOS ONE*, 10(12):e0145474.

Sara Bögels and Francisco Torreira. 2015. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52:46–57.

Jan P. De Ruiter. 2019. Turn-taking. *The Oxford Handbook of Experimental Semantics and Pragmatics*, page 536–548.

Felix Gervits, Ravenna Thielstrom, Antonio Roque, and Matthias Scheutz. 2020. It's about time: Turn-entry timing for situated human-robot dialogue. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 86–96, 1st virtual meeting. Association for Computational Linguistics.

John Godfrey and Edward Holliman. 1993. Switchboard-1 release 2 ldc97s62. *Linguistic Data Consortium*.

Mattias Heldner and Jens Edlund. 2010. Pauses, gaps and overlaps in conversations. *Journal of Phonetics*, 38(4):555–568.

P Indefrey and W.J.M Levelt. 2004. The spatial and temporal signatures of word production components. *Cognition*, 92(1–2):101–144.

Divesh Lala, Koji Inoue, and Tatsuya Kawahara. 2019. Smooth turn-taking by a robot using an online continuous model to generate turn-taking cues. In *2019 International Conference on Multimodal Interaction*, pages 226–234.

Lilla Magyari and J. P. de Ruiter. 2012. Prediction of turn-ends based on anticipation of upcoming words. *Frontiers in Psychology*, 3.

Julia Beret Mertens and J. P. De Ruiter. 2021. Cognitive and social delays in the initiation of conversational repair. *Dialogue & Discourse*, 12(1):21–44.

Carina Riest, Annett B Jorschick, and Jan P de Ruiter. 2015. Anticipation in turn-taking: mechanisms and information sources. *Frontiers in Psychology*, 6:89.

Matthew Roddy, Gabriel Skantze, and Naomi Harte. 2018. Multimodal continuous turn-taking prediction using multiscale rnns. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 186–190.

Jan-Peter de Ruiter, Holger. Mitterer, and N. J. Enfield. 2006. Projecting the end of a speaker's turn: A cognitive cornerstone of conversation. *Language*, 82(3):515–535.

Harvey Sacks, Emanuel A Schegloff, and Gail Jefferson. 1978. A simplest systematics for the organization of turn taking for conversation. In *Studies in the organization of conversational interaction*, pages 7–55. Elsevier.

Gabriel Skantze. 2021. Turn-taking in conversational systems and human-robot interaction: A review. *Computer Speech & Language*, 67:101178.

A Appendix

A.1 TCU Duration

Data Mean	2.40E+03
Exponential Model Mean	2.40E+03
Gamma Model Mean	2.40E+03
Data Std Dev	3.36E+03
Exponential Model Std Dev	2.40E+03
Gamma Model Std Dev	2.39E+03
Exponential Rate Mean	4.16E-04
Gamma Rate Mean	4.22E-04
Exponential Rate Std Dev	3.79E-06
Gamma Rate Std Dev	6.27E-06
Gamma Shape Mean	1.01E+00
Gamma Shape Std Dev	1.17E-02
Exponential WAIC	2.101E+05
Gamma WAIC	2.101E+05

A.2 TCU Word Count

Data Mean	7.70E+00
Exponential Model Mean	7.70E+00
Gamma Model Mean	7.70E+00
Data Std Dev	7.62E+00
Model Std Dev	7.70E+00
Gamma Model Std Dev	7.10E+00
Exponential Rate Mean	1.30E-01
Gamma Rate Mean	1.53E-01
Exponential Rate Std Dev	1.20E-03
Gamma Rate Std Dev	2.23E-03
Gamma Shape Mean	1.18E+00
Gamma Shape Std Dev	1.38E-02
Exponential WAIC	7.275E+04
Gamma WAIC	7.255E+04

A.3 TCU Syllable Count

Data Mean	9.81E+00
ExponentialModel Mean	9.80E+00
Gamma Model Mean	9.80E+00
Data Std Dev	1.01E+01
Exponential Model Std Dev	9.80E+00
Gamma Model Std Dev	9.56E+00
Exponential Rate Mean	1.02E-01
Gamma Rate Mean	1.07E-01
Exponential Rate Std Dev	9.44E-04
Gamma Rate Std Dev	1.57E-03
Gamma Shape Mean	1.05E+00
Gamma Shape Std Dev	1.21E-02
Exponential WAIC	7.852E+04
Gamma WAIC	7.850E+04

A.4 Turn Duration

Data Mean	4.83E+03
Exponential Model Mean	4.82E+03
Gamma Model Mean	4.83E+03
Data Std Dev	7.93E+03
Exponential Model Std Dev	4.83E+03
Gamma Model Std Dev	5.56E+03
Exponential Rate Mean	2.07E-04
Gamma Rate Mean	1.56E-04
Exponential Rate Std Dev	2.69E-06
Gamma Rate Std Dev	3.49E-06
Gamma Shape Mean	7.55E-01
Gamma Shape Std Dev	1.23E-02
Exponential WAIC	1.116E+05
Gamma WAIC	1.113E+05

A.5 Turn Word Count

Data Mean	1.53E+01
Exponential Model Mean	1.53E+01
Gamma Model Mean	1.53E+01
Data Std Dev	2.34E+01
Exponential Model Std Dev	1.53E+01
Gamma Model Std Dev	1.81E+01
Exponential Rate Mean	6.53E-02
Gamma Rate Mean	4.69E-02
Exponential Rate Std Dev	8.55E-04
Gamma Rate Std Dev	1.04E-03
Gamma Shape Mean	7.18E-01
Gamma Shape Std Dev	1.14E-02
Exponential WAIC	4.390E+04
Gamma WAIC	4.341E+04

A.6 Turn Syllable Count

Data Mean	1.95E+01
Exponential Model Mean	1.95E+01
Gamma Model Mean	1.95E+01
Exponential Model Std Dev	1.95E+01
Data Std Dev	3.02E+01
Gamma Model Std Dev	2.39E+01
Exponential Rate Mean	5.13E-02
Gamma Rate Mean	3.42E-02
Exponential Rate Std Dev	6.72E-04
Gamma Rate Std Dev	7.73E-04
Gamma Shape Mean	6.67E-01
Gamma Shape Std Dev	1.05E-02
Exponential WAIC	4.676E+04
Gamma WAIC	4.600E+04

A.7 TCUs per Turn

Data Mean	1.99E+00
Geometric Model Mean	1.99E+00
Neg Binomial Model Mean	1.99E+00
Data Std Dev	1.90E+00
Geometric Model Std Dev	1.40E+00
Neg Binomial Model Std Dev	1.62E+00
Geometric p Mean	5.03E-01
Neg Binomial p Mean	7.59E-01
Geometric p Std Dev	4.61E-03
Neg Binomial p Std Dev	1.12E-02
Neg Binomial n Std Dev	6.27E+00
Neg Binomial n Std Dev	3.82E-01
Geometric	16209.397714
NegativeBinomial	20315.258775

B Supplemental Material

Getting Better Dialogue Context for Knowledge Identification by Leveraging Document-level Topic Shift

Nhat Tran and **Diane Litman**
University of Pittsburgh
nlt26@pitt.edu, dlitman@pitt.edu

Abstract

To build a goal-oriented dialogue system that can generate responses given a knowledge base, identifying the relevant pieces of information to be grounded in is vital. When the number of documents in the knowledge base is large, retrieval approaches are typically used to identify the top relevant documents. However, most prior work simply uses an entire dialogue history to guide retrieval, rather than exploiting a dialogue's topical structure. In this work, we examine the importance of building the proper contextualized dialogue history when document-level topic shifts are present. Our results suggest that excluding irrelevant turns from the dialogue history (e.g., excluding turns not grounded in the same document as the current turn) leads to better retrieval results. We also propose a cascading approach utilizing the topical nature of a knowledge-grounded conversation to further manipulate the dialogue history used as input to the retrieval models.

1 Introduction

Knowledge identification (KI) is the task of identifying relevant information from a database of documents that should be used when generating responses in a *knowledge-grounded dialogue system* (Feng et al., 2020; Wu et al., 2021). When the number of documents is large, information retrieval is typically used to find relevant documents (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Yu et al., 2021). Most approaches encode both the knowledge sources and the dialogue context (i.e., all prior turns), which is later used as an input query, into the same vector space. Since the quality of the input query significantly impacts the retrieval results (Yu et al., 2020, 2021), using an optimal dialogue context is crucial.

In knowledge-grounded dialogues, each turn can be grounded in a different document. Blindly including all previous turns into the dialogue context can introduce unnecessary noise because a turn grounded in a different document can provide redundancy or irrelevant information for the grounding process of the current turn. Our **hypothesis** is that including only turns in the dialogue context that are grounded in the same document as the current turn when creating a retrieval query will improve KI task performance. To test this hypothesis, we tried several approaches to select relevant turns to be included in the dialogue context. Specifically, we vary the input to a previously used predictive model (Lewis et al., 2020b) to see whether querying using only turns grounded in the same document as the current turn improves retrieval performance. After verifying our hypothesis using oracle results, we utilize automatically computed *document-level topic shifts* to improve the dialogue context used for KI. Even with imperfect automatic predictive models, our initial results show that improving dialogue context increases the retrieval results on dialogues grounded on at least 2 documents. Further analysis on errors from dialogues grounded only in 1 document leads us to a simple heuristic that raises the retrieval accuracy for the entire dataset.

Our contribution is twofold. First, we verify the importance of a proper contextualized query in the KI task, as excluding utterances from the dialogue context that are not grounded in the same document as the current turn leads to better knowledge retrieval results in an oracle condition. Second, based on that verification, we develop a simple automatic approach that improves KI in document-grounded dialogue by leveraging a proposed topic segmentation algorithm that uses both dialogue content and grounding documents.

2 Related Work

Our work is related to recent work in **knowledge identification (KI) in knowledge-grounded dialogues** (Choi et al., 2018; Dinan et al., 2019; Qu et al., 2020; Feng et al., 2020; Campos et al., 2020; Wu et al., 2021). However, prior work has largely

Proceedings of the SIGdial 2022 Conference, pages 368–375
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

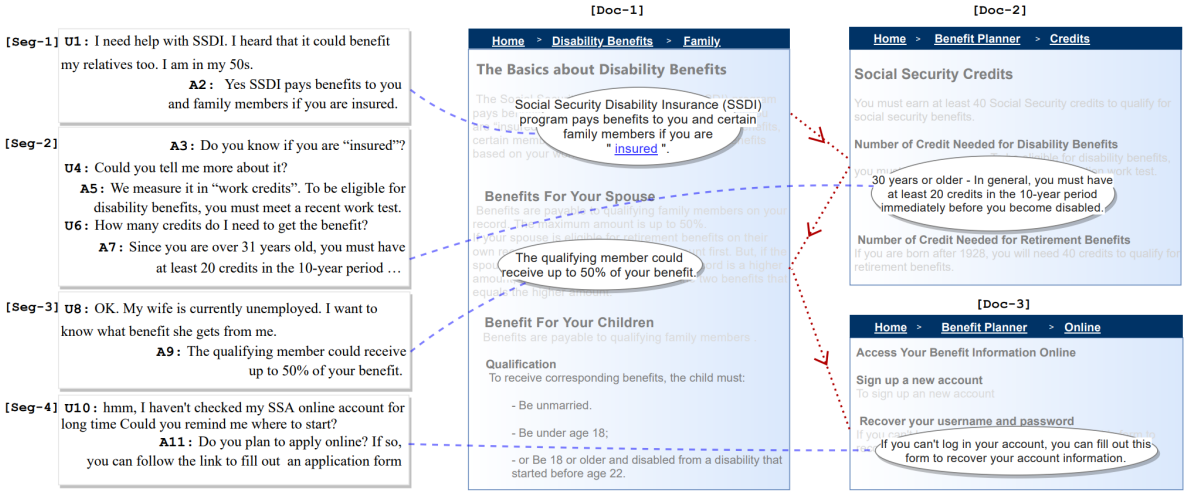


Figure 1: An example dialogue from MultiDoc2Dial (Feng et al., 2021) that is grounded in three different documents.

treated KI as reading comprehension since all turns in a conversation were typically grounded in one document. In a dataset such as MultiDoc2Dial (Feng et al., 2021), a reading comprehension approach is less computationally feasible due to the surge in the number of grounding documents. We thus approach KI as information retrieval following Dalton et al. (2020) and Yu et al. (2021). However, in those studies, turns were again closely related to the same topic, so a full dialogue context was typically used for the query. We instead use predicted document-level topic shifts as the basis of a simple discourse-informed query approach, yielding improved results for KI in MultiDoc2Dial.

Our focus on document-level topic shifts in dialogue is related to the task of **discourse segmentation**. Prior work in identifying topic changes has used topic tracking with predefined topics (Soleimani and Miller, 2016; Takanobu et al., 2018) and used coherence scores between consecutive utterances to split the conversation into smaller topics (Xu et al., 2021; Xing and Carenini, 2021). However, such segmentation approaches have typically been based solely on the content of conversations. In contrast, we propose a topic segmentation approach based not only on the dialogue content, but also on the grounding document.

3 Task and Dataset

3.1 Knowledge Identification Task

We follow the definition of *knowledge identification (KI)* from Feng et al. (2021): given the current user turn, dialogue context, and the entire set of documents from the same domain, find the grounding text span from one document that the next agent response needs to refer to.

3.2 Dataset

We use **MultiDoc2Dial** (Feng et al., 2021) as our dataset. It consists of 4796 information-seeking conversations grounded in 488 documents from 4 domains (only one domain per dialogue). 948 of them are grounded in only 1 document.

This dataset suits our study as the full dialogue context of a turn may span multiple topics. Figure 1 shows a dialogue in the corpus that contains four segments and is grounded in three different documents. A *segment* signals that all turns within it are grounded in the same document and the boundary between two segments indicates a *topic* shift. The presence of such document-level topic shifts can make a turn more contextually distant from the previous turn (Arguello and Rosé, 2006). In Figure 1, Seg-3 requires knowledge about "spouse" from Doc-1, but that information is unimportant for the query about "SSA online account" in Seg-4. Including U8 and A9 in the dialogue context when asking about "SSA online account" is not useful and can even add noise to the retrieval query.

4 Background

Passages as Retrieval Units. Since a grounding document can be very long, we split each one into passages and use them as the units for retrieval. We follow Feng et al. (2021) to split a document based on its original paragraphs indicated by markup tags and then attach the hierarchical titles from their html source to each paragraph as a *passage*.

Dense Passage Retrieval (DPR). DPR (Karpukhin

Algorithm 1 Cascading algorithm to get the top 10 passages for a N-turn dialogue Dial = $\{t_1, t_2, ..., t_N\}$

procedure FINDTOPK($Dial$)
  $DOCS$ = { } ▷ List of documents have been used for grounding so far, empty in the beginning
  for $i = 1$ to N **do**
    for $j = 1$ to len($DOCS$) **do**
      h_j = concatenation of turns t_k where $k < i$ and $ground[k] = j$
      for each passage p_x in $DOCS[j]$ **do** ▷ $DOCS[j] = \{p_1, p_2, ..., p_m\}$
        $Score[p_x] = PC(t_i, h_j, p_x)$
        if $Score[p_x] > Best\_score[j]$ **then**
          $Best\_passage[j] = p_x$
          $Best\_score[j] = Score[p_x]$
    $Best\_doc = \arg\max_d Best\_score[d]$
    if $Best\_score[Best\_doc] < 0.5$ **then** ▷ No old documents can be used for grounding
      Use DPR with **only** the current turn t_i as the query to retrieve the top 10 passages $TOP\_10[i]$
      $ground[i]$ = The document containing the highest-score passage from $TOP\_10[i]$
    else
      $ground[i] = Best\_doc$ ▷ Choose the highest-score passage for grounding
      $TOP\_10[i]$ includes:
      - The passage with the highest score: $Best\_passage[Best\_doc]$
      - Top 3 other passages from $Best\_doc$
      - Up to top 3 other passages with the highest $Score$ this turn
      - Remaining non-duplicate passages from the entire database retrieved by DPR, using only the previous turns of t_i grounded on *Best_doc* for the dialogue history

      Add documents contains passages from $TOP\_10[i]$ to $DOCS$

  return $TOP\_10[N]$

et al., 2020) is an approach to quickly find the top k passages relevant to a given input query from a big database. DPR uses two BERT encoders (Devlin et al., 2019), one to index all passages to d-dimensional vectors, and one to map the input query to the same d-dimensional vector space. Because the similarity between a query and a passage is defined as the dot product of their vectors, retrieving the top k passages at inference time can be done efficiently when the encoded passages are indexed offline by FAISS (Johnson et al., 2021). The *input query* in our task is the concatenation of the current user turn and the dialogue context.

Retrieval-Augmented Generation (RAG). RAG (Lewis et al., 2020b) is our base response generation model. It consists of a *retriever* module (DPR) and a *generator* module (BART, Lewis et al., 2020a). The retriever gets the most relevant passages given the input query, and the generator takes the query and top-k passages as input to generate the response as output. In our task, the target response is the grounding span, that is, the specific piece of information used to ground the response for the current user turn (see ovals in Figure 1).

5 Method: Document-level Topic Shift

Since KI methods typically use all previous turns as the dialogue context, instead of focusing on improving model architectures for knowledge-grounded response generation, we examine whether varying the ***input*** (e.g., dialogue context) to such models improves the retrieval and generation results. Specifically, we hypothesise that for the current turn t_i, including only previous turns grounded in the same document as t_i in the dialogue context to DPR will improve the overall passage retrieval results. To verify this hypothesis, we first create an oracle model called **RAG-oracle**. It assumes that the correct grounding passages of previous turns are known, so it only uses the turns grounded in the same document as t_i in the input query to DPR.

However, since the gold-standard grounding information of the dialogue is not available in real use cases, we build a simple classification model

to estimate it. This model, which we call the **Passage Checking Model (PC)**, is a BERT model fine-tuned on Multidoc2Dial. The input includes the current user turn t_i, the dialogue context h, and one passage p. The output is 1 if t_i should be grounded in p given h and 0 otherwise. During training, the dialogue context only contains turns grounded in the same document as t_i. For each training instance, we sample 128 negative passages[1], at most half of them are from the same document which p belongs to and the rest are from different documents. Our PC model achieved 69.4 F_1 score on validation set. We also use the probability scores from the last layer (softmax) as a confidence measure below.

Next, we use PC in a cascading algorithm to retrieve the top 10 passages for the current user turn (details in Algorithm 1). For each conversation, we process the turns increasingly while keeping track of a list of documents (*DOCS*) that have been used for grounding so far. At each turn t_i, we try to ground it to each document in *DOCS* and use only turns grounded in the same targeting document as the dialogue context. We add the documents containing one of the top-10 passages to the set *DOCS* before going to the next turn. The model based on this algorithm is called **RAG-cascade**.

Finally, since the BART generator relies on the top-5 passages to provide the grounding span, having a better top-5 can yield improved generation results. We explore this idea by reusing the probability scores from the PC model as a **ranking** metric instead of building another ranking model.

6 Experiments and Results

Following Feng et al. (2021), all numbers reported in this section are the mean of three runs with different random seeds. For retrieval, we use recall at k (R@k), which measures the frequency of the correct passage found in the top-k retrieved passages. Token-level F_1 score and Exact Match (EM) (Rajpurkar et al., 2016) are used to evaluate the grounding span generation results. Implementation details can be found in Appendix A.

6.1 Experiment Setup

RAG was the only model used to identify the grounding passage (retriever) and generate the grounding span (generator) in our experiments. We only vary the **input** to the RAG model to demonstrate different approaches to choose the dialogue context (details in Table 1).

6.2 Passage Retrieval Results

We report the passage retrievel results on the entire evaluation data of MultiDoc2Dial (*D*) as well as on a subset of data containing at least two segments (D_2) in Table 2. On *D*, `RAG-oracle` consistently outperforms the RAG baselines. The gap is most noticeable at R@10 (6.4 points). The discrepancy is even bigger on D_2 with more than 7.5 points increases in both R@1 and R@10. These numbers support our hypothesis that only using turns grounded in the same document as the current turn in the dialogue context creates a better contextualized input query for the retriever module (DPR).

While `RAG-cascade` has higher recall on D_2 compared to `RAG-baseline`, they perform similarly (less than 1.3-point differences) on D. This implies that the improvement on data with multiple segments was offset by the degradation in data with only one segment (about 19.7% of *D*). We believe these errors come from the loss of context from previous turns when our model incorrectly decides to split a one-segment dialogue into multiple segments at some point and this error starts propagating (see Appendix B for an example).

The distribution of incorrect segmentation in one-segment dialogues from validation set shows that about 70% of them occur when more than 6 turns appear in the dialogue context (Appendix C). A naive heuristic of limiting the number of turns in the context to 6, while it does not affect the retrieval performance on D_2, reduces errors on one-segment data, and as a result, increases the overall performance in *D*. This is demonstrated by the fact that `RAG-limit` is superior to `RAG-baseline` and `RAG-cascade` in the full evaluation data.

`RAG-topic` also uses topic segmentation as additional information to create the relevant dialogue context, but it has the worst performances in terms of passage retrieval. This implies that in contrast to our proposed `RAG-cascade` model where the "topic" is identified based on the grounding document, using a document-agnostic approach to do dialogue topic segmentation is ineffective.

Re-ranking does not always improve R@1. The rises in R@5 are clearer, where the largest boosts in *D* and D_2 come from `RAG-oracle` (3.3) and `RAG-cascade` (4.4), respectively. We observe several decreases in recall with re-ranking, but all of them are within 0.8 points. `RAG-oracle` with

[1]The same negative sample size used by Feng et al. (2021).

Model	Dialogue Context used in the Input to RAG
RAG-baseline	All previous turns
RAG-oracle	Turns grounded in the same document as the current turn
RAG-cascade	Turns grounded in the same document as the current turn, predicted by algorithm 1
RAG-limit	Same as RAG-cascade but the maximum number of turns is limited to 6
RAG-topic	Like RAG-oracle but uses a dialogue topic segmentation method (Xing and Carenini, 2021) to decide the thresholds from calculated coherence scores between 2 consecutive utterances while ignoring all grounding documents (in contrast to RAG-cascade)

Table 1: Dialogue context used in the input for the experimented RAG models.

Model	Passage Retrieval						Span Generation	
	All Data (D)			> 2 Segments Data (D_2)				
	@1	@5	@10	@1	@5	@10	F_1	EM
RAG (Feng et al., 2021)	49.0	72.3	80.0	n/a	n/a	n/a	41.9	*24.9*
RAG-baseline	48.6	72.5	79.2	40.2	63.5	72.3	41.1	23.8
+ re-ranking	49.0	74.7	79.2	40.1	65.4	72.3	*43.7*	23.4
RAG-oracle	55.1	74.5	**85.6**	**47.9**	69.2	**79.8**	43.1	**25.9**
+ re-ranking	**55.3**	**77.8**	**85.6**	47.5	**73.2**	**79.8**	**43.8**	25.7
RAG-topic	42.1	65.7	71.3	40.2	60.9	70.3	36.2	20.9
+ re-ranking	42.0	67.6	71.3	39.4	62.6	70.3	36.5	20.6
RAG-cascade	48.9	72.8	80.4	44.4	67.2	76.1	41.0	23.7
+ re-ranking	49.7	75.3	80.4	*44.6*	*71.6*	76.1	41.2	23.8
RAG-limit	*52.8*	74.1	*82.3*	44.3	67.0	*76.3*	41.5	23.8
+ re-ranking	52.5	*75.4*	*82.3*	44.3	71.0	*76.3*	41.4	24.0

Table 2: Passage retrieval and span generation results. Best results from MultiDoc2Dial (Feng et al., 2021) are reported in the first row. **Bold** numbers are the best overall results, *underlined* numbers demonstrate the best results besides RAG-oracle. All numbers are statistically significant ($p < 0.05$) compared to RAG-baseline.

re-ranking achieves the best results in all categories, except for R@1 in D_2 where the version without re-ranking shows a 0.4-point lead.

6.3 Span Generation Results

We also report the grounding span generation results. With automation, we see no improvements in F_1 and EM. Even with increases in R@5 from re-ranking, we do not witness much gain in span metrics. Feng et al. (2021) reported a similar pattern where some models perform better in passage retrieval but are inferior in grounding span generation. Our assumption is that passages in top-5 that are not the correct grounding for the current user turn may contain irrelevant or contextually incorrect information for the BART generator.

7 Conclusion

In this work, we showed that exploiting document-level topic shifts in document-grounded dialogues relying on multiple documents as the knowledge base can raise passage retrieval results. We first proposed a simple cascading approach based on a simple BERT model for passage checking and re-ranking that yielded improved retrieval results for multiple-segment dialogues. An error analysis suggested that limiting the number of turns in the dialogue context to 6 reduced the false segmentation errors for the one-segment dialogues and thus improved the scores for the full corpus. Furthermore, no improvement from span generation with the increased retrieval results implied that a general-purpose generative model like RAG might not be a good fit for knowledge identification task in information-seeking dialogues. Future plans include using better generative models to generate better system responses from the identified knowledge and conducting further analysis on the segmentation yielded from the proposed algorithm.

Acknowledgments

We would like to thank Cuong Nguyen and Kien Do for their comments on the initial draft of the paper, and the reviewers for their helpful feedback.

References

Jaime Arguello and Carolyn Rosé. 2006. Topic-segmentation of dialogue. In *Proceedings of the Analyzing Conversations in Text and Speech*, pages 42–49, New York City, New York. Association for Computational Linguistics.

Jon Ander Campos, Arantxa Otegi, Aitor Soroa, Jan Deriu, Mark Cieliebak, and Eneko Agirre. 2020. DoQA - accessing domain-specific FAQs via conversational QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7302–7314, Online. Association for Computational Linguistics.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Jeffrey Dalton, Chenyan Xiong, Vaibhav Kumar, and Jamie Callan. 2020. *CAsT-19: A Dataset for Conversational Information Seeking*, page 1985–1988. Association for Computing Machinery, New York, NY, USA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Song Feng, Siva Sankalp Patel, Hui Wan, and Sachindra Joshi. 2021. MultiDoc2Dial: Modeling dialogues grounded in multiple documents. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6162–6176, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128, Online. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. *ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT*, page 39–48. Association for Computing Machinery, New York, NY, USA.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. *Open-Retrieval Conversational Question Answering*, page 539–548. Association for Computing Machinery, New York, NY, USA.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Hossein Soleimani and David J. Miller. 2016. Semi-supervised multi-label topic models for document classification and sentence labeling. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, page 105–114, New York, NY, USA. Association for Computing Machinery.

Ryuichi Takanobu, Minlie Huang, Zhongzhou Zhao, Fenglin Li, Haiqing Chen, Xiaoyan Zhu, and Liqiang Nie. 2018. A weakly supervised method for topic segmentation and labeling in goal-oriented dialogues via reinforcement learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4403–4410. International Joint Conferences on Artificial Intelligence Organization.

Zeqiu Wu, Bo-Ru Lu, Hannaneh Hajishirzi, and Mari Ostendorf. 2021. DIALKI: Knowledge identification in conversational systems through dialogue-document contextualization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1852–1863, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Linzi Xing and Giuseppe Carenini. 2021. Improving unsupervised dialogue topic segmentation with utterance-pair coherence scoring. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 167–177, Singapore and Online. Association for Computational Linguistics.

Yi Xu, Hai Zhao, and Zhuosheng Zhang. 2021. Topic-aware multi-turn dialogue modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14176–14184.

Shi Yu, Jiahua Liu, Jingqin Yang, Chenyan Xiong, Paul Bennett, Jianfeng Gao, and Zhiyuan Liu. 2020. *Few-Shot Generative Conversational Query Rewriting*, page 1933–1936. Association for Computing Machinery, New York, NY, USA.

Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. 2021. *Few-Shot Conversational Dense Retrieval*, page 829–838. Association for Computing Machinery, New York, NY, USA.

A Implementation Details

Since we do not modify the architecture of the RAG models, we adopt the implementation from Feng et al. (2021) [2] and keep all of the hyperparameters unchanged. We also use the same 5:1:1 train/validation/test split. For the implementation of the Passage Checking (PC) model, we use the uncased BERT version with default parameters [3].

B An Example Error in One-segment Document

Table 3 illustrates a case when the prediction errors were propagated in a one-segment document grounded entirely in document **ssa#1**. Here, **ssa#1** refers to the document "How Financial Aid Works | Federal Student Aid#1_0" and **ssa#3** is "Teacher Loan Forgiveness | Federal Student Aid#1_0". At the turn 3, `RAG_cascade` incorrectly predicted the grounding document to **ssa#3**, which is still relevant to "loan", but for teachers instead. Starting from this, the algorithm favors **ssa#3** and omits the presence of "financial aid" from **ssa#1** in the dialogue context.

[2] https://github.com/IBM/multidoc2dial

[3] https://huggingface.co/bert-base-uncased

C Error Distribution in One-Segment Dialogues

Figure 2 illustrates the proportions of errors in relation to the number of turns included in the dialogue history when the entire conversation is grounded in one document.

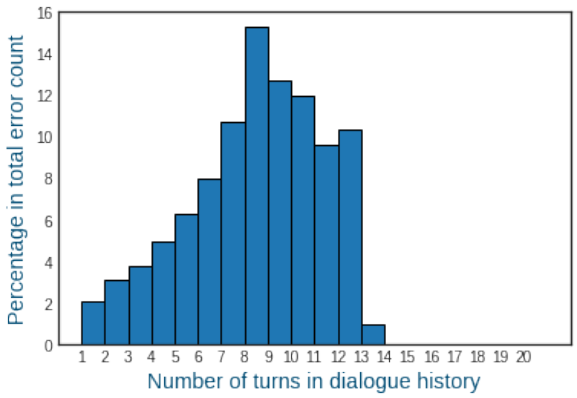


Figure 2: Error distribution in one-segment dialogues.

Turn	Utterance	Predicted Doc
1	Hello, I would like to know who can receive financial aid	ssa#1
1	of course we are here to give you More information	
2	How can I estimate the aid I can access	ssa#1
2	Use FAFSA4caster to get an early estimate of your eligibility for federal student aid.	
3	I also want to about the repayment. And would you recommend that I pay the student loans?	ssa#3
3	As you prepare to graduate, prepare to pay off your student loans. Good news! Federal student loan borrowers have a six-month grace period before payments begin.	
4	and how to determine if I am eligible for help?	ssa#3
4	Your college uses your FAFSA data to determine your eligibility for federal aid.	

Table 3: An example one-segment dialogue where the prediction errors are propagated. User's utterances are in grey.

Neural Generation Meets Real People: Building a Social, Informative Open-Domain Dialogue Agent

Ethan A. Chi*, **Ashwin Paranjape***, **Abigail See***, **Caleb Chiam***, **Trenton Chang**, **Kathleen Kenealy**, **Swee Kiat Lim**, **Amelia Hardy**, **Chetanya Rastogi**, **Haojun Li**, **Alexander Iyabor**, **Yutong He**, **Hari Sowrirajan**, **Peng Qi**, **Kaushik Ram Sadagopan**, **Nguyet Minh Phu**, **Dilara Soylu**, **Jillian Tang**, **Avanika Narayan**, **Giovanni Campagna**, and **Christopher D. Manning**

Stanford NLP
{ethanchi, ashwinp, abisee, calebc96, manning}@cs.stanford.edu

Abstract

We present Chirpy Cardinal, an open-domain social chatbot. Aiming to be both informative and conversational, our bot chats with users in an authentic, emotionally intelligent way. By integrating controlled neural generation with scaffolded, hand-written dialogue, we let both the user and bot take turns driving the conversation, producing an engaging and socially fluent experience. Deployed in the fourth iteration of the Alexa Prize Socialbot Grand Challenge, Chirpy Cardinal handled thousands of conversations per day, placing second out of nine bots with an average user rating of 3.58/5.

1 Introduction

Despite recent major advances (Adiwardana et al., 2020), open-domain *chit-chat*—friendly, social, casual conversation—remains a challenging task. In addition to difficulties with the sheer length and open-endedness required, social chatbots, or "socialbots," often struggle with *fluency*—whether due to the canned responses of manually constructed dialogue trees (Walker et al., 2001) or the anomalies of neural generators (Nie et al., 2021). But just being error-free isn't enough: to have a rewarding conversation, socialbots must be *personable*—displaying emotional intelligence, a rich personality, and an understanding of social dynamics. Although methods exist to address many of these issues individually, combining all of these features into a full-bodied conversation remains difficult.

In this paper, we describe Chirpy Cardinal, an open-domain conversational socialbot, which aims to bridge the gap between traditional dialogue tree-based approaches (Walker et al., 2001; Chen et al., 2018) and large pretrained neural dialogue agents (Adiwardana et al., 2020; Roller et al., 2020). Capable of discussing thousands of topics, Chirpy centers emotional and social intelligence with the goal of authentic, engaging interaction. Specifically, we make the following contributions:

- Conversations with open-domain socialbots often lack a stable structure. To ameliorate this, we present an **extensible design** for open-domain dialogue which prioritizes conversational stability and flexibility through mixed initiative (Horvitz, 1999).
- Although pretrained neural generators can be extremely fluent (Collins and Ghahramani, 2021), real-life deployment can suffer from a lack of both controllability and consistency (Nie et al., 2021). To address this, we describe several approaches to **integrate neural generation** into a symbolic setup, achieving local fluency without sacrificing global coherence.
- Towards the goal of a rewarding conversation, we suggest a set of approaches—ranging from small routines to complete submodules—which aim to make our socialbot a **good conversational partner**. We focus on being both *flexible*—handling a wide variety of topics in an interesting and informative way (Section 4)—and *personable*—empathizing with the other interlocutor even in difficult topics or situations (Section 5).

Deployed in the Alexa Prize Socialbot Grand Challenge 4, Chirpy Cardinal reached thousands of users per day; with conversations lasting up to 45 minutes at a time, it placed second out of nine agents in the finals. We open-source our system as an extensible framework for open-domain social dialogue,[1] providing an example of real-world deployment of conversational NLP systems.[2]

*Equal contribution.

[1] github.com/stanfordnlp/chirpycardinal
[2] stanfordnlp.github.io/chirpycardinal

Proceedings of the SIGdial 2022 Conference, pages 376–395
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

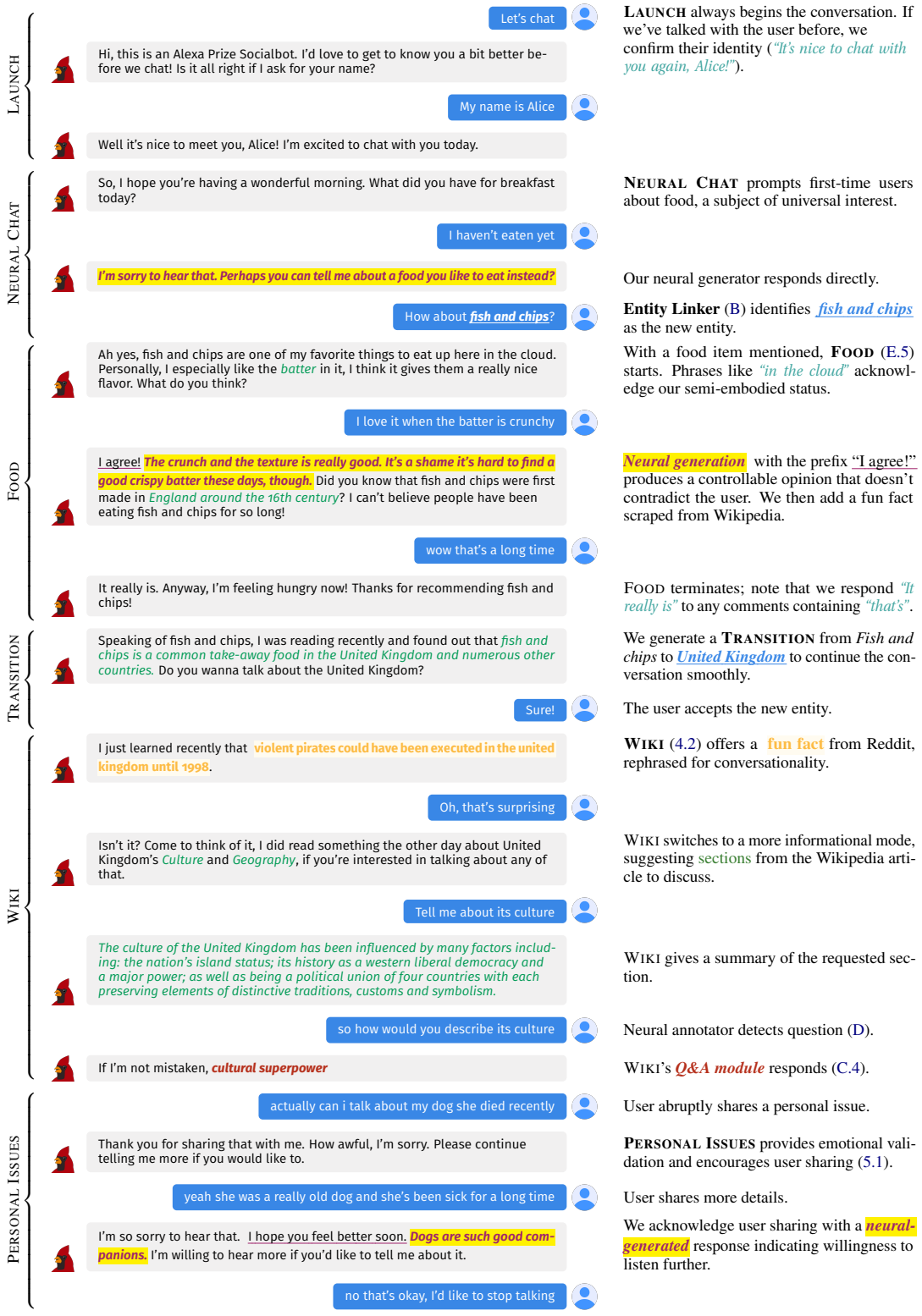


Figure 1: An example dialogue. To respect users' privacy, this is a conversation with an author, not a real user. We model dialogue as a series of subconversations (left side), whose *topics* are parsed and tracked by a neural entity linker. Each subconversation is managed by a RG, whose "scaffolded" architecture comprises hand-written treelets (plain text) incorporating numerous dynamic elements, ranging from ***neural generation*** to *retrieval from Wikipedia* to **neurally rephrased fun facts**. Prefix-based generation provides controllability, especially for sensitive topics like personal issues.

2 Design

2.1 System Design

We model a user dialogue as a series of subconversations (Figure 1), each handled by a *response generator* (RG). Varying greatly in scope and domain, each RG handles a specific topic (e.g. MOVIES, SPORTS) grounded in the outside world. RGs comprise dialog trees (Weizenbaum et al., 1966), whose tree nodes, which we term *treelets*, implement custom logic (e.g. intent classification or retrieval) to generate a response.

At the start of each turn, the user utterance is annotated for linguistic features (Appendix C), then processed in parallel by all RGs. By default, the previous turn's RG is selected; should the RG that last responded crash or a different RG request to take over, we seamlessly switch RGs and move to a new subconversation.

2.2 Navigation

To enable mixed initiative—shared user-bot responsibility in driving the conversation (Horvitz, 1999)—we provide a suggested navigational path, while letting users deviate drastically from it. Specifically, each RG continues through its dialogue tree until exhausting its subconversation; we then transition to another RG by bringing up a previously user-mentioned topic (*"You mentioned **cats** earlier; would you say you're a big fan?"*), mentioning a tangentially related topic that we can discuss well, or simply sampling a new RG and corresponding topic at random. Users may explicitly change the topic (*"can we talk about roblox"*); implicitly suggest a desire to redirect the conversation (*"yeah"* or *"uh-huh"*); or otherwise behave in ways that require the bot to act dynamically (*"i don't know, how about you?"*). We handle these deviations from the conversational flow through neural handlers that allow periods of flexibility before returning to the overall conversational structure (Appendix F).

2.3 Entity Handling

To allow users to discuss a vast array of interesting topics relevant to their lives, we support any Wikipedia entity as a topic of discussion.[3] To do so, we entity-link (Kolitsas et al., 2018) the user utterance to relevant entities using a fine-tuned BERT model (Broscheit, 2019; also B.3), mitigating ASR errors through a phonetic similarity search (B.2).

[3]Specifically, those with sufficiently high cross-references and meeting certain criteria for definiteness (Appendix B.1).

Since incorporating Wikipedia article titles directly into bot utterances can be awkward (e.g. *"can we talk about **cat**"*), we refer to entities by more natural *talkable names* (e.g. *"**cats**"*), generated using GPT-3 (Brown et al., 2020).

RG	Prefix	Sample Completion
FOOD	A hoagie is a great choice! I especially love...	*"...mine with a little cheese and bacon!"*
PERSONAL ISSUES	That sounds frustrating. I hope that...	*"...she feels better soon."*

Table 1: Sample uses of conditional neural generation.

3 Neural Generation

Although neural generative models (Roller et al., 2021) have achieved success in open-domain dialogue, significant obstacles impede deployment in real-life situations: neural text degeneration (Holtzman et al., 2020; Welleck et al., 2019), hallucination (Dziri et al., 2021), and inconsistency (Zhang et al., 2018), In addition, large latency can make models challenging to deploy in practice (Worswick, 2020). In this section, we investigate ways to utilize the power of such models in the context of structured dialogue. We propose integrating neural generation in the context of hand-written scaffolding, aiming to benefit from its variety and fluency while maintaining coherency over time.

3.1 DistillBlender: A Fast, General-Purpose Neural Generator

For general use, we distill a single model from BlenderBot-3B (Roller et al., 2021) with 9 decoder layers,[4] reducing latency significantly over the original model. We use it as follows:

- The **NEURAL CHAT** RG, which directly exposes lightly edited neural model outputs as a subconversation. Due to BlenderBot's end-to-end training, this is initially a rich, fluent conversational experience, but due to rapid degradation we terminate after 5 turns.
- *Conditional prompting* (Keskar et al., 2019), which enables controllability in a structured context. We apply hand-written prefixes to guide the model towards fluent, contextually appropriate completions (Table 1).

[4]Reduced from an original 24.

Template	I love how [actor] acted in [film], especially their <mask>.
Infilled	I love how [Keanu Reeves] acted in [The Matrix], especially their ability to freeze time.

Table 2: An example of template-based infilling using Keanu Reeves as the knowledge source.

3.2 Template-Based Infilling

Towards the goal of rich, coherent conversation for a wide class of topics, we propose *template-based infilling*, a more flexible version of standard slot-filling methods (Haihong et al., 2019) that does not require a structured knowledge base. Using both freeform information and an end-user-defined template, we use a fine-tuned BART model (Lewis et al., 2020) to generate a grounded statement. Defining a diverse set of templates for each entity category allows us to provide expressive yet controllable conversation on many different types of entities (Table 2).

4 Response Generators

4.1 NEWS

The **NEWS** RG aims to discuss current events, which often feature heavily in typical human-to-human chit-chat (De Boer and Velthuijsen, 2001). When an entity or topic mentioned in *The Washington Post* or *The Guardian* appears in conversation, we offer a headline, conversationally paraphrased using GPT-3 (Brown et al., 2020), as a subject of conversation.[5] If the user is interested, we provide a summarized (Zhang et al., 2020a) snippet of the storyand allow the user to ask follow-up questions answered via neural QA (Clark et al., 2020; Rajpurkar et al., 2018; also B.3). Answers are then rephrased (Paranjape et al., 2020) and reranked using PCMI (Paranjape and Manning, 2021), allowing our socialbot to dynamically integrate current events into conversations when relevant.

4.2 WIKI

In contrast to humans, open-domain chatbots are commonly expected to be able to "engage in conversation on any topic" (Adiwardana et al., 2020). Towards this end, the **WIKI** RG discusses any entity. We aim to be informative, not overwhelming; in addition to encouraging users to share their own knowledge and experience about the entity, we bring up interesting factoids from /r/todayilearned (conversationally rephrased; E.3.4), as well as infilled remarks. We then discuss the entity in more depth based on its article, flexibly acknowledging user questions and comments with the Q&A handler (C.4) or neural generation.

4.3 OPINION

A core part of social chit-chat (Walker, 2009), exchanging and commenting on opinions allows a socialbot to project a stronger sense of personality. The **OPINION** RG solicits users' opinions on topics and reciprocates with its 'own' opinions (sourced from Twitter), including occasional *disagreement* to help engage user interest (E.4).

4.4 Rules-based RGs

In order to broaden the scope of our bot, we manually build several domain-specific response generators. **FOOD**, which always opens the conversation, discusses common foods scraped from Wikipedia.[6] **MOVIES** uses the Alexa Linked Data API to discuss movies and actors. **MUSIC** uses the MusicBrainz[7] database to discuss songs, artists, and music genres. **SPORTS** uses the ESPN API to discuss NFL football and NBA basketball. We describe these RGs in more detail in Appendix E.

5 Being Personable

To achieve truly social conversation, a socialbot must be a *good conversational partner*: empathetic, supportive, and interested in what its human interlocutor has to say (Salovey and Mayer, 1990; Li et al., 2017). In this section, we describe several approaches that aim to achieve this, ranging from full RGs to smaller subroutines.

5.1 Handling Personal Issues

Many users—especially those who chat with our socialbot looking for companionship—share personal struggles with our bot, requiring emotional sensitivity and tact. Handling such conversations purely neurally would result in rapid degeneration due to neural toxicity (Dinan et al., 2021). To address this, the PERSONAL ISSUES RG responds to personal disclosures using active listening techniques (Bodie et al., 2015), asking exploratory questions about

[5] We use `davinci` with the following prompt: "Paraphrase news headlines into a complete, grammatical sentence in plain English. The sentence should be in the past tense."

[6] In practice, we found that always starting with FOOD proved to be most successful for ratings (E.1), perhaps since food is such a universal human need and discussion point.

[7] https://musicbrainz.org/

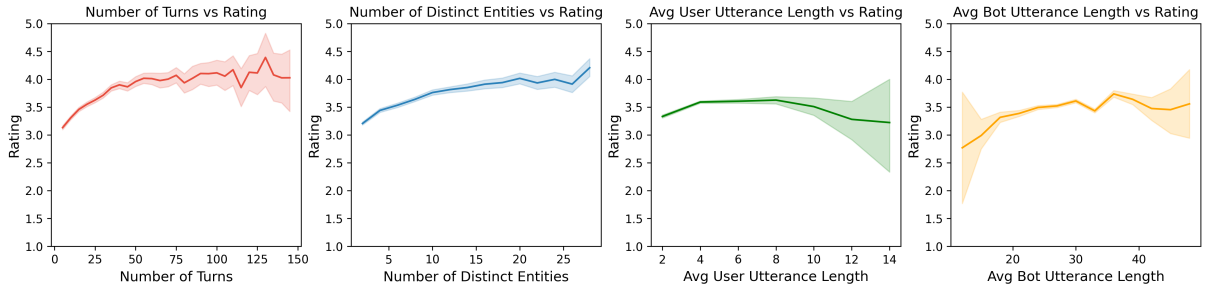


Figure 2: Engagement metrics vs. rating. We bucket (with size $5, 2, 2, 3$ respectively) conversations based on four engagement metrics—number of turns, number of distinct entities, average user utterance length, and average both utterance length—and plot each bucket against user rating (Likert 1-5 scale, measured per-conversation). 95% confidence intervals computed via bootstrapping ($n = 1000$).

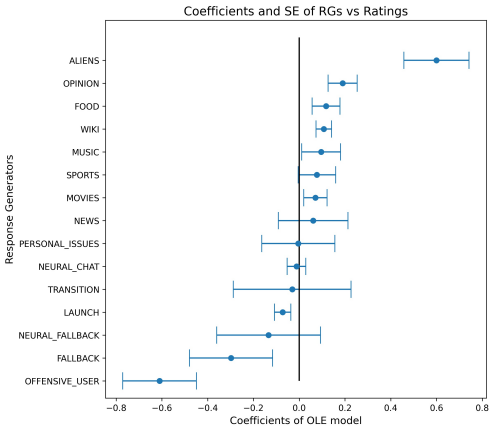


Figure 3: Linear regression coefficients for response generator vs. rating; each RG is weighted by the number of turns it contributes. 95% confidence intervals determined via bootstrapping with $n = 1000$.

the nature of the user's issue (*"When did you start feeling this way?"*), and validating their concerns (*"I see, that sounds difficult."*)

On the other hand, a significant subset of users become verbally abusive during the conversation (Curry and Rieser, 2018, 2019). We follow the strategy of Li et al. (2021): a de-escalating statement to avoid confrontation, addressing the user by name (*"John"*); then changing the topic.

5.2 Self-disclosure

The ALIENS RG allows the socialbot to muse about its pet topic—the possible existence of extraterrestrial life—as well as its own identity and sense of purpose. Contrasting with purely informational modes, this RG fleshes out a personality for our agent and enables *self-disclosure*—disclosing goals, attitudes, and personal interests to support interpersonal intimacy (Altman and Taylor, 1973; Ignatius and Kokkonen, 2007).[8]

[8]This RG comes up only after sufficient rapport has been built—i.e. after 30 turns in the conversation.

5.3 Personalization

Users often expect chatbots to remember personal preferences and user details (Chaves and Gerosa, 2021; Svikhnushina et al., 2021) and to tailor their responses accordingly (Neururer et al., 2018; Shum et al., 2018). We personalize bot responses with the user's preferences: for example, in regards to the Olympics, *"Ah, that makes sense since you did say it's your favorite sport!"*. Referencing this user state across conversations makes repeated conversations with Chirpy feel fresh and dynamic, rather than retreading past questions and topics.

6 Results

In this work, we have outlined a set of design priorities and corresponding approaches to design a fluent, flexible, and sociable chatbot. We validate these through the Alexa Socialbot Grand Challenge 4: engaging in approximately 1,000 conversations per day, our socialbot achieved an average user rating of **3.55**, ending the development period tied for first place in rating.[9] Validating our design goals, we observe high ratings for a hybrid neural-scaffolded approach (FOOD, etc.), personable RGs (ALIENS), and open-domain techniques (WIKI) (Figure 3). Our socialbot engages in long, varied conversations without repeating itself (Figure 2).

That said, both overall rating and sample conversations testify that Chirpy remains far from the goal of truly compelling and enjoyable human-bot interaction. We do not argue that our approaches are sufficient—or even necessary—to create such an ideal system; rather, we hope that the *priorities* outlined here can serve as a starting point to help inform further socialbot development, whether purely neural or hybrid in nature.

[9]Likert scale between 1 and 5; overall average across teams was **3.47**. For more information, please consult the proceedings of the Alexa Prize Socialbot Grand Challenge 4.

Ethics Statement

In this work, we have presented a conversational agent that conducts an open-domain dialogue. We believe that many people would enjoy having a chat partner who is empathetic and knowledgeable, and our ratings seem to suggest that a reasonable number of people appreciate their conversations enough to want to talk to the bot again. Prior to engaging with the chatbot, all user participants are required to consent to their conversations, feedback, and ratings being recorded, as per the Alexa Terms of Use. Additionally, the chatbot clearly identifies itself as a bot at the start of each conversation. No actual user conversations or identifying information are used in this paper.

However, as our system incorporates computational methods for generating conversational utterances automatically, there exists a risk that users may be exposed to unsafe utterances or discussion topics. Conversational models of all kinds can produce sexist, racist, or otherwise unsafe statements; neural conversational agents can be particularly vulnerable due to pre-training on Internet chat forums, which can be particularly toxic (Xu et al., 2020). Towards this end, our system incorporates a safety module that prevents our model from producing utterances with certain hard-coded words or categories. Yet the use of a blacklist in itself raises additional ethical issues, as poorly designed blacklists can marginalize communities by blocking topics that ideally, one should be able to discuss equitably.

Finally, the human-like nature of open-domain dialogue systems can be particularly damaging when used in an adversarial context, e.g. by state actors (Boshmaf et al., 2012). Ultimately, like all text generation methods, the benefit of releasing an open-domain dialogue model must be weighed against its possible downsides.

Acknowledgements

We thank Amazon.com, Inc. for a grant partially supporting the work of the team and *The Guardian* for allowing us to use their news API for our system. Additionally, we thank Anna Goldie and Monica Lam for helpful discussions.

The user icon in Figure 1 is from *kdg design* and used under a free license.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Irwin Altman and Dalmas A Taylor. 1973. *Social penetration: The development of interpersonal relationships.* Holt, Rinehart & Winston.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings. In *Proceedings of the 5th International Conference on Learning Representations*.

Graham D Bodie, Andrea J Vickery, Kaitlin Cannava, and Susanne M Jones. 2015. The role of "active listening" in informal helping conversations: Impact on perceptions of listener helpfulness, sensitivity, and supportiveness and discloser emotional improvement. *Western Journal of Communication*, 79(2):151–173.

Yazan Boshmaf, Ildar Muslukhov, Konstantin Beznosov, and Matei Ripeanu. 2012. Key challenges in defending against malicious socialbots. In *5th {USENIX} Workshop on Large-Scale Exploits and Emergent Threats ({LEET} 12)*.

Samuel Broscheit. 2019. Investigating entity knowledge in BERT with simple neural end-to-end entity linking. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 677–685, Hong Kong, China. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ana Paula Chaves and Marco Aurelio Gerosa. 2021. How should my chatbot interact? a survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction*, 37(8):729–758.

Chun-Yen Chen, Dian Yu, Weiming Wen, Yi Mang Yang, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, et al. 2018. Gunrock: Building a human-like social bot by leveraging large scale real user data. *Alexa Prize Proceedings*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Eli Collins and Zoubin Ghahramani. 2021. LaMDA: our breakthrough conversation technology. *Google AI Blog*.

Amanda Cercas Curry and Verena Rieser. 2018. #MeToo Alexa: How conversational systems respond to sexual harassment. In *Proceedings of the Second ACL Workshop on Ethics in Natural Language Processing*, pages 7–14.

Amanda Cercas Curry and Verena Rieser. 2019. A crowd-based evaluation of abuse response strategies in conversational agents. In *20th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 361.

Connie De Boer and Aart S Velthuijsen. 2001. Participation in conversations about the news. *International Journal of Public Opinion Research*, 13(2):140–158.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Gavin Abercrombie, A Stevie Bergman, Shannon Spruit, Dirk Hovy, Y-Lan Boureau, and Verena Rieser. 2021. Anticipating safety issues in e2e conversational ai: Framework and tooling. *arXiv preprint arXiv:2107.03451*.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, Shrimai Prabhumoye, Alan W Black, Alexander Rudnicky, Jason Williams, Joelle Pineau, Mikhail Burtsev, and Jason Weston. 2019a. The second conversational intelligence challenge (convai2). ArXiv preprint arXiv:1902.00098.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019b. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Nouha Dziri, Andrea Madotto, Osmar Zaiane, and Avishek Joey Bose. 2021. Neural path hunter: Reducing hallucination in dialogue systems via path grounding. *arXiv preprint arXiv:2104.08455*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Xiang Gao, Yizhe Zhang, Michel Galley, Chris Brockett, and Bill Dolan. 2020. Dialogue response ranking training with large-scale human feedback data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 386–395, Online. Association for Computational Linguistics.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinglang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, Dilek Hakkani-Tür, and Amazon Alexa AI. 2019. Topical-chat: Towards knowledge-grounded open-domain conversations. In *INTERSPEECH*, pages 1891–1895.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Eric J. Horvitz. 1999. Principles of mixed-initiative user interfaces. In *CHI '99: Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 159–166.

Emmi Ignatius and Marja Kokkonen. 2007. Factors contributing to verbal self-disclosure. *Nordic Psychology*, 59(4):362–391.

Dan Jurafsky, Liz Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse function annotation coders manual. In *Technical Report Draft 13, University of Colorado, Institute of Cognitive Science*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *CoRR*, abs/2006.10369.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, et al. 2018. Advancing the state of the art in open domain dialog systems through the Alexa Prize. *arXiv preprint arXiv:1812.10757*.

Nikolaos Kolitsas, Octavian-Eugen Ganea, and Thomas Hofmann. 2018. End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Haojun Li, Dilara Soylu, and Christopher D. Manning. 2021. Large-scale quantitative evaluation of dialogue agents' response strategies against offensive users. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Luca Massarelli, Fabio Petroni, Aleksandra Piktus, Myle Ott, Tim Rocktäschel, Vassilis Plachouras, Fabrizio Silvestri, and Sebastian Riedel. 2020. How decoding strategies affect the verifiability of generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 223–235, Online. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL*, pages 152–159.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.

Mario Neururer, Stephan Schlögl, Luisa Brinkschulte, and Aleksander Groth. 2018. Perceptions on authenticity in chat bots. *Multimodal Technologies and Interaction*, 2(3):60.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2021. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1699–1713.

Ashwin Paranjape and Christopher Manning. 2021. Human-like informative conversations: Better acknowledgements using conditional mutual information. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 768–781, Online. Association for Computational Linguistics.

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI tech report*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Stephen Roller, Y-Lan Boureau, Jason Weston, Antoine Bordes, Emily Dinan, Angela Fan, David Gunning, Da Ju, Margaret Li, Spencer Poff, et al. 2020. Open-domain conversational agents: Current progress, open problems, and future directions. *arXiv preprint arXiv:2006.12442*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Peter Salovey and John D Mayer. 1990. Emotional intelligence. *Imagination, cognition and personality*, 9(3):185–211.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235.

Sam Shleifer and Alexander M. Rush. 2020. Pre-trained summarization distillation. ArXiv preprint arXiv:2010.13002.

Heung-Yeung Shum, Xiao-dong He, and Di Li. 2018. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers of Information Technology & Electronic Engineering*, 19(1):10–26.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Ekaterina Svikhnushina, Alexandru Placinta, and Pearl Pu. 2021. User expectations of conversational chatbots based on online reviews. In *Designing Interactive Systems Conference 2021*, pages 1481–1491.

Marilyn Walker, Rebecca J Passonneau, and Julie E Boland. 2001. Quantitative and qualitative evaluation of DARPA communicator spoken dialogue systems. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 515–522.

Marilyn A Walker. 2009. Endowing virtual characters with expressive conversational skills. In *International workshop on intelligent virtual agents*, pages 1–2. Springer.

Joseph Weizenbaum et al. 1966. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1):36–45.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Steve Worswick. 2020. Bot battle update — we won?

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.

Dian Yu, Michelle Cohn, Yi Mang Yang, Chun-Yen Chen, Weiming Wen, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, Giritheja Sreenivasulu, Sam Davidson, Ashwin Bhandare, and Zhou Yu. 2019. Gunrock: A social bot for complex and engaging long conversations. ArXiv preprint arXiv:1910.03042.

Dian Yu and Zhou Yu. 2021. MIDAS: A dialog act annotation scheme for open domain HumanMachine spoken conversations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1103–1120, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020a. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and William B Dolan. 2020b. DialoGPT: Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278.

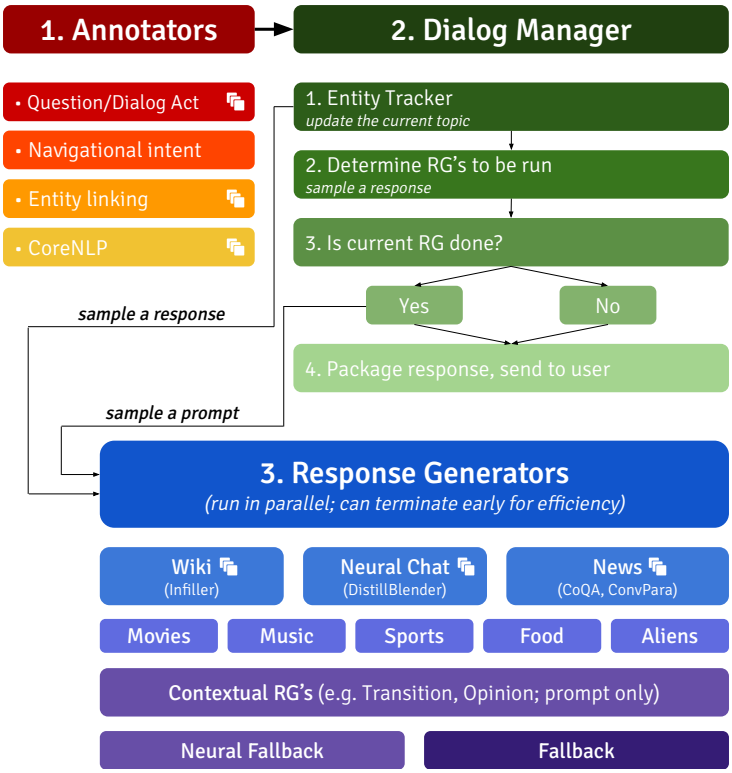


Figure 4: Overall system design.

A Additional Architectural Details

A.1 Overall Architecture

Our system (Figure 4) is based on CoBot (Khatri et al., 2018). During the Alexa Prize, Chirpy Cardinal ran on AWS Lambda, a serverless computing platform; our open-source demo runs on Kubernetes. For reliability, our function is stateless; therefore, to preserve information between turns, we store our bot's overall state in an external PostgreSQL state table (see Figure 4). We execute the following steps on each turn:

1. Fetch the previous turn's state from the state table.
2. Generate a response from our neural generator (for latency reasons; D.1).
3. Execute all annotators (C), which run on remote CPU-only instances.
4. Analyze the user utterance for **navigational intent** (A.3) to determine whether we should change topic.
5. Analyze the user utterance for entities (B.4). If warranted by the user's navigational intent or the last bot response, the **current entity** (B.4) is updated.
6. Run all RG's (Section 4) in parallel; RG's that require a neural response await the neural generator. Out of all received responses, select a response (A.2), and update the current entity if necessary.
7. If the chosen response generator has finished its conversation, we run our collection of RG's a second time to produce prompts (A.2) Select a prompt, update the current entity again if needed, and form the bot's utterance by appending the prompt to the response.

At the end of the turn, the bot's overall state contains the user's utterance, the conversational history, the NLP Pipeline annotations for the user's utterance, and a state for each individual RG. Each individual RG state contains information required by that RG – for example, it might contain the current treelet in the RG's dialogue graph, or a list of the utterances and/or entities that have been discussed, to avoid repetition.

A.2 Response Design

Responses and prompts both carry a *priority*, with the highest-priority response/prompt chosen at the corresponding stage. In general, the RG which responded last has the highest priority; however, RG's can optionally specify a lower priority so that other RG's take over, or a higher priority to take over from another RG. In practice, these priority

levels are rarely used due to their tendency to produce a choppy conversation.

A.3 Navigational Intent Classifier

A user has *positive* navigational intent if they want to discuss a topic; conversely, *negative* navigational intent means that the user would like to avoid discussing a topic. Users may express navigational intent while specifying a topic (*"can we talk about minecraft"*, referring to the current topic (*"let's discuss this more"*), or referring to no topic (*"I don't want to chat anymore"*). Positive and negative navigational intents can even be combined (*"I don't want to talk about movies any more, let's chat about you"*). We classify use manually-constructed regexes, which achieve extremely high precision.

B Entity-Linking Details

Detecting and understanding references to real-world entities is essential to any open-domain conversational system; we find that users appreciate being able to discuss a wide variety of topics that interest them or are relevant to their lives. For our socialbot, we train and deploy a neural entity linker that links spans to Wikipedia entities.

B.1 Entity Pool

To obtain our pool of potential entities, we process the May 20th, 2020 dump of English-language Wikipedia[10] using MWParserFromHell[11] and Spark[12]. We store our data in a large ElasticSearch index, keeping only entities with at least 200 cross-references in Wikipedia. In total, we have 171,961 entities.

Notably, certain entities are inappropriate to discuss even if correctly entity-linked by our model; for example, our system is unable to handle abstract nouns well (e.g., *philosophy*, *film*). To ameliorate this, we manually created a set of *low-precision* entities composed of both WikiData categories (e.g., *conspiracy theory*, *financial risk*, *research method*) and specific common entity names (e.g., *bank*, *catalog*, *coast*). The bot will not start a conversation itself about such entities; however, it is able to handle explicit user navigational requests (e.g., *can we talk about the bank*). Separately, we also ban certain racial, religious, and other identity-based terms that are unlikely to result in a good conversation on either the bot's or user's part, as well as certain short acronyms (e.g. *cet*, *ep*, *fm*) that are almost always triggered by ASR errors.

B.2 Candidate generation

For a given user utterance, we want to compute the set of entities that the user could possibly be referring to; for example, if the user mentions *"swift"*, this could refer to the bird, musical artist, or programming language. To do so, for each possible span, we pre-compute the set of entities for which the span serves as a Wikipedia anchortext, creating a mapping from spans to sets of candidate entities. At execution time, for all n-grams in the user utterance with 5 or fewer tokens[13], we retrieve the set of candidate entities from our database.

Since we do not have access to original user audio, ASR errors can impede candidate generation (Chen et al., 2018). For example, if an user's reference to the film *Ford v Ferrari* is erroneously transcribed as *"four v ferrari"*, a naïve entity linker will fail to identify the correct entity. To address this, we pre-compute phoneme and metaphone representations for all of our entities (e.g. converting *Harry Potter* to `'HH EH R IY P AA T ER'`[14] and `'HRPTR'`[15]). At execution time, each n-gram's candidate set is augmented with the sets for spans with similar phoneme/metaphone representations.

B.3 Entity disambiguation

Given a set of candidate entities, we want to select those candidates that the user is interested in. Towards this end, we fine-tune a BERT-medium (Devlin et al., 2019) to disambiguate entities, following Broscheit (2019) with minor modifications. Specifically, we learn an embedding for each entity in our dataset. Then given a span within an user utterance, we model the probability that the span refers to a given candidate entity as the dot product between the contextual span representation and the entity's embedding. At deployment, we only take entities with a predicted likelihood of at least 0.5; additionally, we use only the highest-likelihood entity for each span.

We depart from Broscheit by mean-pooling over the contextualized span representation, rather than doing per-token entity-level disambiguation. Fine-tuning takes about 20 days using 4 Titan X GPUs; during deployment, we execute using CPU only.

[10] https://dumps.wikimedia.org/

[11] `https://mwparserfromhell.readthedocs.io/en/latest`

[12] `https://spark.apache.org`

[13] specifically, those not solely composed of stopwords

[14] `https://pypi.org/project/g2p-en/`

[15] `https://pypi.org/project/metaphone/`

B.4 Entity Tracking

At any given point, we track the *current entity* (the current subject of conversation), a set of *untalked* entities (entities which the user has mentioned but we have not yet addressed), and a set of *rejected* entities (which the user does not want to discuss; these are no longer brought up by our bot.). These are updated every turn as follows:

- Entities receiving negative navigational intent (*"can we not talk about paraguay"*) are **rejected**. Non-specific negative navigational intent (*"let's not discuss this"*) causes the current entity to be rejected instead.
- Entities receiving positive navigational intent (*"can we talk about mexico"*) are **set as the current entity**. The previous conversation ends, with all RGs are prompted to handle this new current entity instead.
- If the currently active RG asked a question on the last turn, the current highest-priority entity is identified as the presumable user answer and **set as the current entity**. Additionally, if the previous question expects a particular category of entities (e.g. *"What's your favorite movie?"*), we pick the highest-priority entity matching the expected category (e.g., `film`).
- All remaining entities are marked as *untalked* (to be possibly discussed later).

C Annotators

All annotators—modules which provide linguistic annotations for the user utterance—are executed in parallel at the beginning of each turn.

C.1 CoreNLP

We use the following annotators from Stanford CoreNLP (Manning et al., 2014): tokenization, sentence splitting, part-of-speech tagging, lemmatization, named entity recognition, constituency parsing, dependency parsing, coreference resolution, and sentiment analysis. Due to the format of the user utterances (lowercase with no punctuation), we use caseless models[16] for part-of-speech tagging, constituency parsing and named entity recognition. We use these annotations for certain hand-written NLU operations.

Training Regime	Silver	Gold	Test F1
Baseline	0	0	0.53
Self-training ($\tau = .95$)	41,152	0	0.54
Self-training ($\tau = .75$)	62,150	0	0.54
Hand-labeled	0	2,407	**0.81**

Table 3: Performance of our Dialogue Act model under different training regimes. All models have access to $10,090$ examples in the MIDAS training set, but training a baseline model solely on these examples suffers from domain shift. *Self-training*, which first uses this baseline model to silver-label a large number of unlabeled Chirpy Cardinal examples with confidence above some cutoff τ, then retrains on the union of the two, does not improve performance. *Hand-labelling* a small amount of additional data significantly improves performance.

C.2 Dialogue Act Classifier

Dialogue acts, an ontology over user intents (Stolcke et al., 2000; Jurafsky et al., 1997), have been successfully employed in open-domain dialogue agents (Yu et al., 2019). We modify MIDAS (Yu and Yu, 2021)—an annotation schema designed specifically for human-chatbot dialogue— to better fit the needs of our bot, removing 4 labels[17] due to low frequency in our conversations and creating 5 new labels: *correction*, *clarification*, *uncertain*, *non-compliant*, and *personal question*. In total, our modified schema has 24 labels.

Evaluated on the MIDAS test set, a fine-tuned BERT baseline achieves .78 micro-F1; however, evaluated on an OOD test set composed of our own conversations, it achieves only .53 (Table 3). Although self-training (McClosky et al., 2006) proved ineffective, hand-labeling additional OOD conversations achieved a micro-F1 of 0.81. The predictions of this final model inform navigation, as well as RG-specific NLU.

C.3 Question Classifier

Users often spontaneously ask factual questions, personal questions, follow-up questions, and even questions unrelated to the current topic. Recognizing and answering these questions is important, particularly for user initiative, but is also non-trivial, as ASR-transcribed user utterances do not contain punctuation. To recognize questions, we fine-tuned a RoBERTa model (Liu et al., 2019; Wolf et al., 2019) on an simplified version of the Dialogue Act training data, framing the task as binary classifica-

[16] https://stanfordnlp.github.io/CoreNLP/caseless.html

[17] *apology*, *apology-response*, *other*, and *thanks*

tion, conditioned only on the user utterance. This model achieved an F1-score of 0.92 and improved the reliability of question detection.

C.4 QA Annotator

The **QA annotator**, an ELECTRA-Large model (Clark et al., 2020) pretrained on SQuAD2.0 (Rajpurkar et al., 2018), performs question answering for the NEWS (Section 4.1) and WIKI (Section 4.2) RGs. Unlike other annotators, this annotator does not run unless called by these RGs.

D Neural Generation

Our neural agent is a distilled (Hinton et al., 2015) version of BlenderBot-3B (Roller et al., 2021), an autoregressive Seq2Seq model trained on Blended Skill Talk (Smith et al., 2020), Wizard of Wikipedia (Dinan et al., 2019b), ConvAI2 (Dinan et al., 2019a), and Empathetic Dialogues (Rashkin et al., 2019). We distill using Sanh et al. (2019)'s method (as implemented in ParlAI; Miller et al., 2017), using Adafactor (Shazeer and Stern, 2018) with learning rate 6.25×10^{-5}, validation loss-based LR reduction, warmup, and FP16 (Gupta et al., 2015). We used a batch size of 1 for training on a single V100 GPU.

For decoding, we use top-k sampling ($k = 5$) with temperature $T = 0.7$. To encourage response diversity across the conversation, we sample sequences of minimum length randomly chosen from 5, 10, 15, 20, 25; in practice, the length of the generations is 0-2 tokens above the minimum selected length. Additionally, we use delayed beam search (Massarelli et al., 2020), with the conversational history up to 128 tokens in the past serving as context. After decoding, we first filter out offensive, null, and repetitive responses, as well as questions after the first turn. We then select a final response based on the posterior likelihood, among other metrics.

D.1 Analysis

We find that our model qualitatively outperforms a GPT-2 (Radford et al., 2019) baseline fine-tuned on Empathetic Dialogues (Table 4), with similar latency. That said, our model still suffers certain limitations out-of-the-box; we discuss strategies for mitigating these issues.

Diversity-coherence tradeoff For our model, beam search decoding yields coherent but non-diverse responses, while stochastic decoding results in nonsensical generations even under top-p (Holtzman et al., 2020) or top-k (Fan et al., 2018) sampling. Delayed beam search, which samples the first few tokens before defaulting to beam search, yielded more stable behavior than stochastic decoding, and better diversity than beam search.

Degeneration The model outputs conversation-ending phrases (e.g., "I have to go", "It was nice talking") after 7 turns, hurting user experience. Manual examination of the training data revealed this is due to the short lengths of conversations in the training data collected via crowdworkers. We alleviated this with a manual blacklist of conversation-ending phrases and forcing NEURAL CHAT (Section 3.1) to hand off to another RG prior to the 7-turn limit.

Changing the topic Our model tends to generate one or more follow-up questions in every utterance, which are often unrelated to the current topic, creating a disorienting topic change (e.g. *"What's your favorite color?"*). To address this, NEURAL CHAT limits the number of questions in each neural chat *conversation* to one, with subsequent questions truncated. During conditional generation, questions are deleted altogether.

Latency Generating responses can take up to 2s in the worst-case, averaging just under 1s. To mitigate this, we **pre-fetch** responses from DistillBlender at the start of every turn, instead of waiting for the Neural Chat RG to request a neural response. If the responding RG returns a response without utilizing the DistillBlender response during the turn, we ignore its output and return early. Although this results in unused pings to the neural model, we find that the trade-off is worth it.

E Additional RG Details

E.1 Launch

At the beginning of the conversation, the bot initially selected icebreakers at random. However, we eventually found that certain icebreakers tended to fare better than others. Specifically, conversations starting with food-related icebreakers (e.g. "Do you have any recommendations for what I should cook at home?") had an average rating was 3.49 over a sample of 1405 conversations, compared to an average rating of 3.43 for non-food-related icebreakers (e.g. "What did you do over the weekend?") over a sample of 1418 conversations. Digging deeper,

Prev. bot utterance	User utterance	Blender (distilled)	GPT-2 baseline
I love pizza, what's your favorite toppings?	Pepperoni.	That's a good choice. Do you have a favorite kind of cheese? Mine is mozzarella.	Oh yea I love that pizza too. I don't like too much other toppings.
What did you do today?	Alexa, what's up?	Nothing much. I just got back from a run. I'm exhausted.	I'm sorry to hear that. What happened?
Have you been doing anything productive?	Well, I've been trying to be productive but I keep on breaking things.	Breaking things can be very frustrating. I hope you are able to get it all sorted out.	That's no good, it's not productive to be stuck on something and not get to something else.

Table 4: Responses from our neural generator, distilled from BlenderBot-3B (punctuation and capitalization added for clarity), compared to a GPT-2 baseline fine-tuned on Empathetic Dialogues. We find that our neural generator provides stronger performance with similar latency.

we found that if the second turn is handled by the Food RG, we achieved an average rating of 3.64 over 606 conversations, compared to an average rating of 3.49 if the second turn is handled by the Neural Chat RG, over 1684 conversations (second turns are mainly handled by Food and Neural Chat RG's, but sometimes by others).

This prompted us to update our Launch RG so that we open with a food-related question for all conversations, hence increasing the frequency of handing over to the Food RG.

E.2 News

The **NEWS** RG (Section 4.1) curates global news from The Washington Post[18] and The Guardian[19]. Article titles, topic categories, body texts, dates, and content URLs are stored in a constantly updating ElasticSearch index. When a topic or entity available in our index appears in conversation, the News RG brings up related stories from our database. In addition, NEWS also initiaties conversations about currently trending news topics by scraping trending news from Google Trends[20].

Behavior To produce a prompt usable in conversation, we rephrase the headline to conversational form using GPT-3 davinci-instruct-beta.[21] If the user expresses interest in continuing the conversation, the we provides a conversational summary generated by Pegasus-Multinews (Zhang et al., 2020a; Fabbri et al., 2019). Summaries are decoded using 8 beams and a maximum of 50 tokens for conversationality, and are pre-generated for efficiency; if the neural module fails, we instead use an extractive summary (Mihalcea and Tarau, 2004).

Follow-up If the user continues to be engaged, we prompt for questions or comments. If a comment is detected, a neural response is generated using a set of hand-written prefixes; If a question is detected (C.3), they are answered via the QA annotator (C.4). We then conversationally paraphrase the answer using a GPT-2-medium model (Radford et al., 2019) fine-tuned on Topical Chat (Gopalakrishnan et al., 2019) to produce a more human-like response. We use the truncated conversational history as the input history and a merged representation of the answer and the span as the the factual content. It outputs a conversational-sounding paraphrase of the answer. Finally, we rank the generated paraphrases using Fused-PCMI (Paranjape and Manning, 2021).

E.3 Wiki

To support our goal of high-coverage world knowledge (Section 1), the Wiki RG uses Wikipedia articles as grounding to discuss any entity that interests the user and that is not handled by any other RG. Our goal is to allow the user to conversationally discover interesting information about the entity.

E.3.1 Data

We use the Wikipedia dump from May 20th, 2020[22], processed using MWParserFromHell[23] and Spark.[24] We store our data in a large ElasticSearch

[18] https://washingtonpost.com

[19] https://theguardian.com

[20] https://trends.google.com

[21] We use the following prompt: "Paraphrase news headlines into a complete, grammatical sentence in plain English. The sentence should be in the past tense."

[22] https://dumps.wikimedia.org/backup-index.html

[23] https://mwparserfromhell.readthedocs.io/en/latest

[24] https://spark.apache.org

index.

E.3.2 Behavior

Wiki RG facilitates a discussion about an entity based on how it came up in conversation (see Fig. 5). If the user initiates an discussion about an entity, the RG encourages the user to share their own knowledge and experience about the entity. Otherwise, if the entity came up only in passing or as a response to a bot prompt (e.g. "What's a country you would like to visit?"), then the RG responds with an 'infilled' remark (discussed below) or an interesting fact (i.e. 'TILs' scraped from the /r/todayilearned subreddit) about the entity. These conversation starters serve the purpose of drawing the user into a more conversational dialog about the entity before proceeding to a more content-rich discussion of it.

Discussing the entity in depth. If the user responds positively to our initial discussion of the entity, we begin a "Discuss in depth" conversation loop (see Fig. 6). Our bot provides a summary of some section of the entity's Wikipedia article and handles the user's sentiments, opinions, and questions appropriately before checking if the user would like to continue with the discussion. If the user responds affirmatively, we suggest another section for discussion, otherwise we exit the RG. This setup ensures that the user is not overly fatigued by the amount of information generated in these section summaries, while allowing interested users to discuss engrossing topics in great depth.

A short example Wiki interaction is shown in Turns 6 through 10 of Table 1.

E.3.3 Template-Based Infilling

To provide the user with rich, coherent conversation for a wide class of entities, we developed a novel method—*infilling*—which generates interesting remarks from handwritten templates based on relevant context. For example, given the actor Keanu Reaves as the current entity, the template *I love how [actor] acted in [film], especially their <mask>* might be infilled as follows: *I love how [Keanu Reeves] acted in [The Matrix], especially their ability to freeze time*. By defining a diverse set of templates for each entity category, we are able to provide expressive yet controllable conversation on many different types of entities. In effect, this acts as a more flexible version of standard slot-filling methods that does not require a structured knowledge base.

Infilling has the following steps:

- A set of templates and appropriate contexts is **retrieved**. Given some entity, we select a set of handwritten templates based on its Wikidata category (e.g. *actor*, *musical instrument*). For each template, we retrieve an appropriate short context from Wikipedia (approximately 3 sentences) using the mean-pooled GloVe-based method of (Arora et al., 2016).

- Given each (context, template) pair, an **infiller** model fills in the blanks. This is parameterized by a BART-base model trained on a dataset generated by ~ 4300 examples, mostly generated using GPT-3 (Brown et al., 2020) and augmented by hand-written examples.

- The infills are **reranked** by an aggregate DialogRPT (Gao et al., 2020) and likelihood score as measured by a GPT-2-medium model fine-tuned on Empathetic Dialogues.

E.3.4 TIL's: Conversational Paraphrasing

We use this RG as a testbed for our conversational paraphrasing system. The system takes as input the truncated conversational history, and some knowledge context (either a TIL about the current entity, or an excerpt of the Wikipedia article, selected based on TF-IDF similarity to the user's response to an open-ended question). It outputs a conversational-sounding paraphrase of the knowledge context. The model was trained by finetuning a GPT-2-medium language model (Radford et al., 2019) on a processed and filtered version of the TopicalChat dataset (Gopalakrishnan et al., 2019). The paraphrases are generated using top-p decoding with $p = 0.75$ and temperature $\tau = 0.9$, and we pick the one which has the highest unigram overlap with the knowledge context.

E.4 Opinion

Exchanging opinions is a core part of social chit-chat. To form a stronger sense of personality, and to seem more relatable, it is important that our bot can also express its opinions. The Opinion RG's goal is to listen to users' opinions on certain topics, and reciprocate with its 'own' opinions (sourced from Twitter) on those topics.

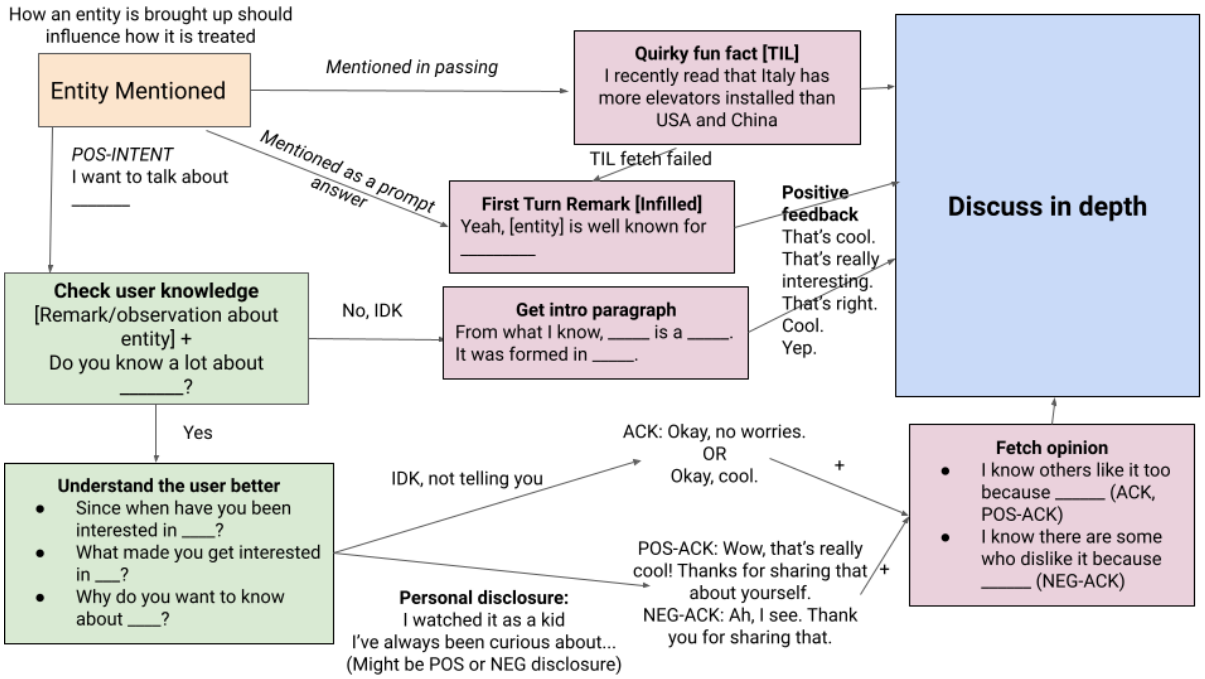


Figure 5: The Wiki RG conversational flow: possible user responses are captured in the edge labels, while bot responses are represented by the vertices.

E.4.1 Data

To collect both positive and negative opinions, we queried a Twitter stream[25] using a regex to collect tweets of the form "`i (love|like|admire|adore|hate|don't like|dislike) TOPIC because REASON`", where `TOPIC` and `REASON` can be any text. We collected 900,000 tweets, which are stored on a Postgres table hosted on AWS Relational Database Service (RDS). Of these, we manually whitelisted 1012 reasons across 109 popular topics. To avoid speaking inappropriately about sensitive topics, we only whitelist uncontroversial entities (such as animals, foods, books/movies/games, everyday experiences such as working from home, being sick, days of the week, etc.), and ensured that all reasons, including negative ones, are inoffensive and good-spirited.

E.4.2 Behavior

Currently, the Opinion RG activates when the user mentions one of the whitelisted entities (e.g. Table 1, Turn 8). We ask whether the user likes the entity and classify their response using the CoreNLP sentiment classifier (Section C.1). We then either agree or disagree with the user. If we disagree, we either ask the user for their reason for their opinion, or supply a reason why we disagree, and ask what they think of our reason. Ultimately, we want the user to have a positive experience with our bot, so regardless of whether we disagree or agree with the user, we will ask the user their opinion on a related entity, and always agree with the user about the new entity. The conversation may end earlier, as we detect on each turn whether the user is still interested via their utterance length. If the utterance contains less than 4 words, and it does not contain any of the 'agreement' words (such as 'same', 'me too', etc.) we will hand off the conversation to another RG. Even when the RG is not active, it keeps track of whether the user has already expressed an opinion on an entity, by applying a regex similar to that applied to the tweets.

E.4.3 Agreement Policies

Disagreement is an unavoidable part of human-human conversations, and we hypothesize that occasional disagreement is necessary in order for our bot to have a convincing and individual personality. To test this, we implemented three policies:

- (i) `ALWAYS_AGREE` – we always agree with the user's sentiment on the entity;
- (ii) `LISTEN_FIRST_DISAGREE` – first we ask the user's reason for liking/disliking the entity, then we offer our reason for disagreeing with their sentiment; and

[25] https://developer.twitter.com/en/docs/tutorials/consuming-streaming-data

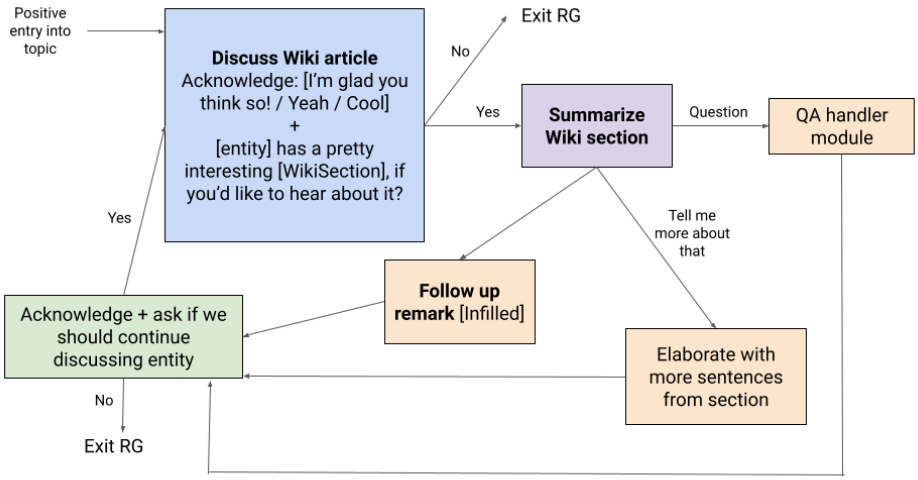


Figure 6: The Wiki RG "Discuss in depth" conversational loop

Policy Name	Continuation Rate (95% CI)
CONVINCED_AGREE	.527 ± .0349
ALWAYS_AGREE	.587 ± .0086
LISTEN_FIRST_DISAGREE	.587 ± .0128

Table 5: Continuation rate for each agreement policy. The Confidence Intervals (CI) differ due to different sample sizes (ALWAYS_AGREE receives 0.5 of traffic, LISTEN_FIRST_DISAGREE receives 0.3, CONVINCED_AGREE receives 0.2).

- (iii) CONVINCED_AGREE – we initially disagree with the user's sentiment on the entity, but after the user gives their reason for liking/disliking the entity, we switch our sentiment to match the user's (i.e. we are convinced by the user).

To evaluate the policies, we ask the user *Would you like to continue sharing opinions?* and interpret the desire to continue is an indication of a successful policy. Table 5 shows that users prefer ALWAYS_AGREE and LISTEN_FIRST_DISAGREE over CONVINCED_AGREE, and all policies have high continuation rates, suggesting that disagreement can be a positive and stimulating part of a conversation, but that the manner and delivery of the disagreement is an important factor.

E.5 Food

The Food RG also focuses on scripted responses to discuss foods and give suggestions. It is often activated at the beginning of the conversation when Neural Chat RG prompts a user for what they have eaten today. The Food RG then goes through a sequence where it asks the user about their favorite variant of that food (e.g. favorite pizza topping), mentions the bot's favorite variant, and possibly provides a fun fact about the food. The Food RG is backed by food data scraped from Wikipedia structured in such a way that subclasses and variants of food are linked to each other. It also uses templated responses with neural infilling to generate descriptions of foods or comments on what the user likes, allowing for variation and flexibility for more interesting responses.

E.6 Movies

The Movies RG is designed to deliver a high-quality scripted conversation about a movie the user specifies, using information drawn from the Alexa Knowledge Graph.[26] Currently, the RG is activated when the user asks to talk about movies, mentions a movie keyword (such as *movies* or *film*) or talks about any movie-related entity (e.g. *Saving Private Ryan*, *Meryl Streep*, *the Coen brothers*, etc.). Once activated, the RG typically asks the user to name a movie, asks the user's opinion on it, gives a fun fact about the movie, asks the user their opinion on an actor in the movie, then asks the user if they've seen a different movie featuring that actor (See Turns 4-7 in Table 1). The RG uses treelets (Section 2) to organize the dialogue graph, hand-written templates to form the bot utterances, and a mixture of regexes and the CoreNLP sentiment classifier (Section C.1) to classify the user's responses.

[26]The Alexa Knowledge Graph is an Amazon-internal resource; our team was given access to parts of it.

E.7 Music

Similar to the Movies RG, the Music RG is designed to deliver scripted conversations about musical entities that the user specify. The RG is activated when a musician/band or a music keyword (such as *music* or *songs*) is mentioned. Once activated, the Music RG engages in a conversation specific to the type of the musical entity that was mentioned. Unlike the Movies RG, the Music RG has a randomized internal prompting system that allows the conversation to be centered around music even when a scripted conversation is exhausted for a specific entity. For example, after the Music RG goes until the end of a scripted conversation for a musician, it can ask for an internal prompt, and start a conversation about musical instruments, songs, or music in general. The randomized nature of the internal prompting system makes the conversation more flexible, and mitigates some of the weaknesses of scripted conversations mentioned in Section E.6.

E.8 Sports

The Sports RG is designed to deliver up-to-date and high-quality conversations on a sport for which the user expresses interest. Currently, we support conversations on NFL football and NBA basketball, the two most-watched sports in the US. When prompted to discuss sports, the user is asked if they are a fan of these two sports. If so, they are asked for their favorite team, but otherwise the conversation moves to a different RG. The RG supports detailed, factual conversation on the user's favorite team, as well as their favorite player on that team. The Sports RG is backed by an ESPN API scraper that pulls information on all NFL and NBA teams (their game schedule, their roster, wins/losses, game analysis, etc.) and facts about all players (their age, position, college, statistics, and expert analysis on their overall play). For example, if the user is a fan of the Denver Broncos, the RG is capable of discussing the Broncos' most recent game (who won/lost, what the score was, what player played well, etc.) and then transitions into discussing a specific Broncos player from the game that the user likes. By utilizing automatic summarization, we are able to intersperse current, specific analysis of their favorite player or team that comes directly from ESPN analysts, giving the conversation a sophisticated and natural tone.

E.9 Fallback

When all other RG's fail to produce a suitable response, we rely upon two fallback RG's that always execute. The Neural Fallback RG selects a fallback responses generated by the DistillBlender model (Section 3.1), with all questions removed. Given that the model is trained on end-to-end dialogue, we find that this is a good conversational baseline. If the Neural Fallback RG fails, we resort to the Fallback RG, which returns a pre-written generic fallback (e.g. "Sorry, I don't know how to answer that").

F High-Initiative User Handling: Example Dialog Flows

Our system uses high-initiative handler functions to detect 18 distinct types of one-turn digressions that interrupt the conversational flow. General examples include complaints and clarifying questions. A comprehensive list of high-initiative situations that our bot can handle and corresponding examples of bot responses are provided in Table 6. These functions can be started in any RG, and allow each RG to pause the flow of the current dialogue tree to handle high-initiative behaviors before either returning to the original conversation or changing the subject.

To avoid repetitiveness, high-initiative handlers are generally non-reentrant, and cannot be triggered consecutively.

G Additional Neural Modeling Experiments

In addition to BlenderBot-3B, we experimented with a variety of autoregressive and non-autoregressive models for text generation. All models are evaluated on an internal dataset of 517 conversation excerpts from early 2021 where the Neural Chat RG was active **AP2021**. We perform qualitative evaluation by passing in each conversational excerpt to the model of interest, and comparing the resultant generation(s) with the original neural generation from GPT2ED. We detail results here.

DialoGPT. DialoGPT (Zhang et al., 2020b) is a GPT-Medium model that has been further fine-tuned on a set of Reddit threads serving as conversational corpora. We evaluated this model offline on a set of excerpts in two settings: 1) **zero-shot** and 2) with fine-tuning on Empathetic Dialogues **(DialoGPT2ED)**. In the **zero-shot** setting, the bot re-

High-Initiative User Behavior	User utterance	Possible response
Was misheard	That's not what I said.	Sorry for the misunderstanding; can you repeat that?
Complains bot was unclear	I don't know what you are talking about.	Oops, let me say that again.
Complains bot was repetitive	Alexa, you already told me that.	You're right, I forgot. Sorry. Let's move on to something else. [NEW_TOPIC]
Complaints about privacy	That's none of your business.	Sorry, maybe that was too personal. I'm happy to talk about something else. [NEW_TOPIC]
Asked about weather	Do you like the weather?	I live in the cloud so I'm not sure what the weather is like on earth!
Said Alexa command	Alexa, play Despacito!	This is an Alexa Prize Socialbot... I can't execute typical Alexa commands...But I'd love to keep talking to you! As I was saying, [PREV]
Wants name correction	Actually, that's not my name.	Oops, it sounds like I got your name wrong. I'm so sorry about that! I won't make that mistake again.
Requested repetition	Alexa, can you say that again?	Oops. Let me repeat that. [PREV]
Requested own name	What's my name?	If I remember correctly, your name is [NAME]
Got cut off	But—	Oops, I think my microphone stopped working for a second. Can you say that again?
Asked for our age	How old are you?	Hmm, I don't know. I'm just a bot, and I don't think we have birthdays.
Asked clarifying question	Wait, did you say "elephants?"	Yes, that's what I said.
Questioned bot's humanity	How can you listen to music if you're a bot?	Well, I'm just a bot, but I live vicariously through others.
Asked personal question	What do you like?	Thanks for asking! [NEURAL]
Interrupted with question	Wait, I have a question.	Sure, what's up?
Expresses desire to chat	Let's chat!	Okay. What would you like to talk about?
Asked for story	Tell me a story.	Here's a story that someone once told me. [STORY]
Gave bot compliment	I like talking to you!	Thanks for saying that. I'm still learning, but it means a lot to hear that from you.

Table 6: Excerpts of possible bot responses to 18 high-initiative user behavior modes. [PREV] refers to the previous bot utterance; [NEW_TOPIC] refers to a sampled prompt from a new RG. [NAME] is the user's name as obtained in the opening turns, and [NEURAL] refers to a DistillBlender-based random response. [STORY] is a handwritten anecdote, omitted here for brevity.

sponds 18% of the time with dirty jokes or memetic content unsafe for open-domain conversation on **AP2021**. After fine-tuning, **(DialoGPT2ED)** responds almost identically to GPT2ED on **AP2021**: qualitatively, the lift from DialoGPT2ED is essentially zero. Hence, this system was not deployed.

DistillBART. DistillBART is our in-house distilled version of BART (Lewis et al., 2020), a model consisting of a non-autoregressive encoder and an autoregressive decoder, each with 12 layers. Notably, this model has decoding complexity $\mathcal{O}(EN + DN^2)$, where N is the sequence length, and E, D are the sizes of the encoder and decoder stacks, respectively. Following results by (Kasai et al., 2020) in the domain of neural machine translation, we hypothesized that we could decrease latency while improving performance by decreasing D; i.e. removing decoder layers and training the decoder via distillation. We performed DistillBERT-style distillation, distilling a BART-Large fine-tuned on Empathetic Dialogues (BARTED) into versions with 6 **(DistillBART-6)** and 3 **(DistillBART-3)** decoder layers. Weight initialization followed a previous setup for BART distillation (Shleifer and Rush, 2020). As baselines, we also trained equivalently-sized models without distillation.

In practice, BART suffered from 1) high latency and 2) mediocre response quality. BART was unable to generate coherent responses stochastically, necessitating the usage of beam search, which hurt decoding speed. On **AP2021**, average decoding speeds for the 12, 6, and 3 layer models were 894ms, 998ms, and 895ms, showing no significant latency gains, which is attributable to the quadratic dependence within the decoding computation on sequence length; i.e. $N^2 \gg D, E$. Furthermore, while distillation certainly resulted in qualitatively better generations on **AP2021** than those of non-distilled models, as shown in Table, there was a sharp dropoff in generation quality on all models except the full-sized BARTED teacher. As BARTED was the only usable model, and yielded generations qualitatively similar to GPT2ED, we did not deploy this system.

DeepCon: An End-to-End Multilingual Toolkit for Automatic Minuting of Multi-Party Dialogues

Aakash Bhatnagar[$], **Nidhir Bhavsar**[&], **Muskaan Singh**[#], **Petr Motlicek**[#]
[$]Boston University, Boston, Massachusetts
[&]Navrachana University Vadodara, India
[#] Speech and Audio Processing Group, IDIAP Research Institute, Martigny, Switzerland
aakash07@bu.edu, nidbhavsar989@gmail.com,(msingh,petr.motlicek)@idiap.ch

Abstract

In this paper, we present our minuting tool *DeepCon*, an end-to-end toolkit for minuting the multiparty dialogues of meetings. It provides technological support for (multilingual) communication and collaboration, with a specific focus on Natural Language Processing (NLP) technologies: Automatic Speech Recognition (ASR), Machine Translation (MT), Automatic Minuting (AM), Topic Modelling (TM) and Named Entity Recognition (NER). To the best of our knowledge, there is no such tool available. Further, this tool follows a microservice architecture, and we release the tool as open-source, deployed on Amazon Web Services (AWS). We release our tool open-source here http://www.deepcon.in.

1 Introduction and Related Work

Due to the COVID-19 pandemic, a substantial part of the working population has seen a significant increase in virtual meetings, especially people working in Information Technology (IT) industry and academia. By all means, meetings are the most vital component to ensure collaborative work and efficient to-and-fro communications. Natural Language Processing (NLP) technologies provide users with a holistic experience in these online interactions. (i) remote conferences or meetings discussions held over an online platform are extremely important in today's globalized world and need interpretation. (ii) Coherent translations of larger documents and dialogues and efficient systems with many sources or target languages. (iii) Summarizing meetings in the form of structured minutes from speech can potentially save up to 80%[1] of time. With all these in mind, we designed an easy-to-use and clean interface that provides (i) Automatic Minuting with dynamic length controlled outputs. (ii) Isometric Translation for five different languages: French, German, Russian, Italian, and Hindi (iii) Topic Extraction

To the best of our knowledge, there is no such tool available. However, some applications, such as Deeptalk[2], provide the fastest way to transform text from chats, emails, surveys, reviews, and social networks into a real business. It is a tool for end users that provides an interactive user interface for topic detection, sentiment analysis, autotagging, analytical interpretation, and summarization. However, this tool lacks in providing users with audio and video support. Wordcab[3] intelligence adds customizable call summaries to applications so that users can revisit conversations in a fraction of the time. It is developer tool that allows developers to integrate their API and easily generate different types of summaries. This tool focuses on generating complex and customized summaries from the given transcript. Happyscribe[4] provides automatic and human transcription services convert audio to text with 85-99% accuracy in 120+ languages and 45+ formats. It provides strong API integration that enables users to handle multi-lingual input. This tool is primarily built for generating transcripts from audio files or automatic subtitles. Happyscribe does not provide any summarization and topic modeling capabilities that limit this platform's scope. Hendrix AI[5] is your intelligent, award-winning AI assistant for meetings that automatically transcribes meeting notes, captures action items and other data points, and uses machine learning to identify productivity insights. Hendrix AI is a highly analytical tool for Zoom meetings. It gives users various outputs such as concise summaries, actionable outcomes, most commonly discussed topics, and meeting effectiveness. Hendrix AI is limited to the Zoom platform's

[1] https://elitr.github.io/automatic-minuting/

[2] https://www.deep-talk.ai

[3] https://wordcab.com

[4] https://www.happyscribe.com

[5] https://hendrix.ai/features

Proceedings of the SIGdial 2022 Conference, pages 396–402
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

(A) Meeting transcript segment:

(PERSON0) Mhm. I will try to get my presentation. All right. Yeah. Mm hmm. Mhm, mm hmm. Hi guys. Sorry. Okay. Excuse me. Mhm, mm hmm. Okay. And then the person. Mm hmm. It is, you know? Yes. Okay. Mhm. But I'm prepared. I'm going in there. Right.
(PERSON2) Mm hmm, mm hmm. Oh, wow,
(PERSON0) amazing.
(PERSON2) It's working. Okay. Thank you. Mhm, mm hmm. Yeah, mm hmm. Okay.
(PERSON1) Yeah, you can flip it. Mhm. Mhm. So, good morning everyone.
(PERSON2) Um
(PERSON2) hmm. Mhm. Mhm. So, the general outline of the, the project will be the first to go through a functional design phase. You all get task in this in this space and ah then we will meet again and discuss this functional design And the same holds for the two faces. After this, the conceptual design and detailed design In with the final project should get 76. Alright. But first we will do from training. Okay. In all, in front of you,
(PERSON4) you are designed we must keep in mind that The selling price of the product will be about €25

(B) Meeting minutes by DeepCon:

- PERSON0 will try to get his presentation
- PERSON3, Person2, Person1 and Person0 are attending the kick off meeting of their latest project
- PERSON1, PERSON2 and PERSON3 discuss the general outline of the project
- The project will go through a functional design phase
- After this, conceptual design and detailed design in with the final project should get 76
- PERSON2, Person3, Person4 and Person0 are designing a project for their company
- The selling price of the product will be about €25

(C) Isometric Translated French Minutes:

- PERSON0 va essayer de faire sa présentation
- PERSON3, Person2, Person1 et Person0 assistent à la réunion de lancement de leur dernier projet
- PERSON1, Person2 et Person3 discutent des grandes lignes du projet
- Le projet passera par une phase de conception fonctionnelle
- Après cela, le design conceptuel et le design détaillé avec le projet final devraient avoir 76
- PERSON2, Person3, Person4 et Person0 conçoivent un projet pour leur entreprise
- Le prix de vente du produit sera d'environ 25 €

(D) Generated Topics:

- tools to communicate
- actual project plan
- functional design phase
- million to make
- thinking television targeting

(E) Translated French Topics:

- outils pour communiquer
- plan de projet réel
- phase de conception fonctionnelle
- millions à faire
- cibler la télévision

Figure 1: An example from our DeepCon tool showing (A) a segment of a meeting transcript (B) along with corresponding generated minutes (C) isometric translated French minutes (d) generated named entity (e) translated French named entity. We have utilized the AMI meeting corpus and anonymized, "PERSON*number*" and "PROJECT*number*" denote persons' and projects' placeholders, respectively.

scope and cannot be used with other platforms and offline recordings.

2 DeepCon

In previous sections, we discuss the various tools that provide similar features as DeepCon. Most of the tools are made as an API service for developers. A tool like Deeptalk, accessible directly to users, does not provide audio and video support. DeepCon, however, provides an easy-to-use interface for end users to utilize advanced transformer models without coding. DeepCon also provides users with multiple features and support of audio and video files on one platform, which are not provided by various other tools.

As seen in Figure 2, DeepCon have various different components. In the following subsections we elaborate on each of the major components in our proposed system.

2.1 Automatic Speech Recognition (ASR)

For generating meeting transcripts, we use Amazon Transcribe [6] which is a Speech-to-text service offered by Amazon AWS. For English, the speech recognition model achieves a WER of 6.2%. The ASR-generated transcripts follow time-sequence order, with both speaker and utterances stated separately. In our DeepCon, we define a post-processing function that aligns speaker roles with corresponding utterances, as shown in figure 1. The user can set a range of $\{2, 10\}$ speakers.

2.2 Automatic Minuting

The meeting summarization module generates minutes, given a transcript. Minuting is primarily concerned with capturing and providing a third-person perspective of essential points raised throughout the meeting. Manual minuting also has drawbacks where the minutes' format and language vary through different annotators. Our tool, DeepCon provides an end-to-end solution to generate consistent and robust meeting minutes.

We use a finetuned BART-large model (Lewis et al., 2019) [7]. We test various summarization models such as T5 (Raffel et al., 2019), Pegasus (Zhang et al., 2019a), RoBERTa2RoBERTa (Liu et al., 2019). However, the BART-based pipeline outperformed the others. This could be because BART utilizes GPT-2 architecture. Further, we fine-tuned

[6] https://aws.amazon.com/transcribe/
[7] https://huggingface.co/lidiya/bart-large-xsum-SAMSum

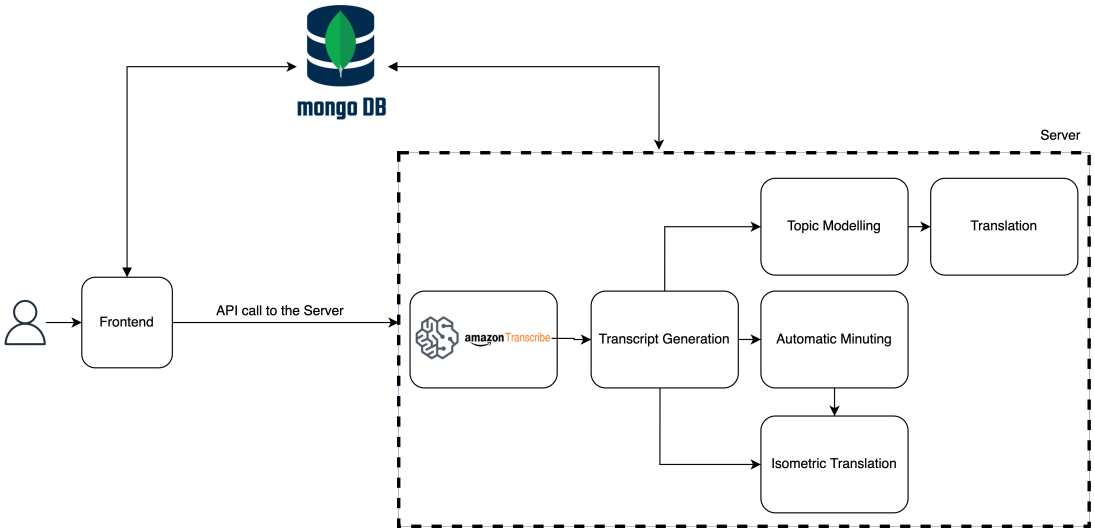


Figure 2: This system architecture diagram represents the pipeline of DeepCon. The first step is for the user to submit the audio file with the desired attributes. This audio file is uploaded on AWS S3, and the link is updated on our MongoDB database. Then, our back-end processes the audio file and generates transcripts using Amazon Transcribe. Further, we use our fine-tuned model for automatic minuting, isometric translation, and topic extraction.

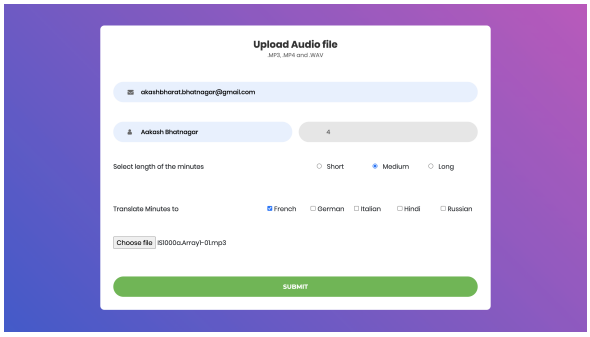


Figure 3: Interface for users to select parameters and upload their audio files. On this page, users enter a name and email ID where they want to receive the process code notification. Users can also select from 3 types of summaries, i.e., short, medium, and long. There is also an option to choose 5 languages per user's requirement.

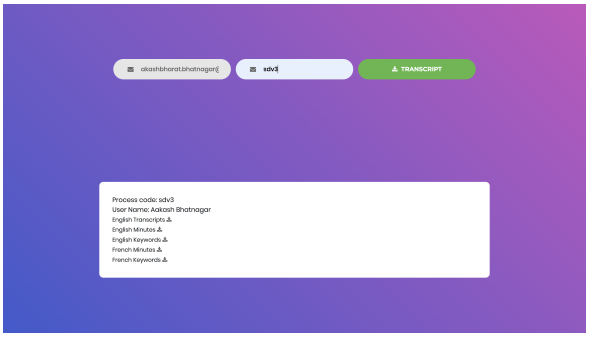


Figure 4: This interface is accessed by the user when the processing of their file is completed. Users can enter the process code here and get the files. All the files here are in text format and can be downloaded easily.

this model on both XSum (Narayan et al., 2018) and SAMSum (Gliwa et al., 2019) datasets. XSum dataset includes short summaries of articles and discussions, whereas SAMSum is a standard dialogue summarization dataset. Training over these datasets provides the BART model the robustness to generate short precise summaries of conversations.

As depicted in Figure 5 Automatic Minuting functionality is divided into 3 major segments. i) We analyze and apply various preprocessing techniques to the generated ASR transcripts, including segmenting the input text into much smaller chunks. (ii) We apply the summarization using a finetuned BART model. (iii) Finally, we use an unsupervised redundancy elimination method to obtain ideal minutes.

The current summarization algorithms are not trained to remove redundancy from a long dialogue discourse and are also restricted to a specific input length for improved text production. Thus to eliminate such redundancies, we specify a few custom rules. We try to eliminate repetitions, pauses, and vocal sounds. We also remove stopwords defined using the publicly available AMI corpus.

We evaluate the generated minutes on the SAMSum Corpus. Table 1 provides scores obtained on the validation & test set measured across various automatic evaluation metrics including R-1, R-2, R-L and R-Lsum (Vasilyev et al., 2020). It is evident that our models achieve higher evaluation scores

	rouge 1	rouge 2	rouge L	rouge Lsum
Validation	54.39	29.81	45.15	49.94
Test	53.31	28.35	44.09	48.92

Table 1: Evaluation results of the summarization model on SAMSum validation and test dataset. We use the ROUGE score for evaluating automatic minutes generated from the text.

Team	rouge 1	rouge 2	rouge L
Auto Minuters	0.25±0.06	0.06±0.03	0.14±0.04
Hitachi	0.26±0.09	0.08±0.03	0.15±0.04
Ours	0.33±0.08	0.08±0.04	0.19±0.06

Table 2: Automated evaluation scores for the best performing system at the AutoMin 2021 shared task.

across all metrics.

Table 2 compares our results on the test set with the two of the best performing system submissions in the AutoMin Shared Task (Ghosal et al., 2021). As depicted our system outperforms the Yamaguchi et al. (2021) & Mahajan et al. (2021).

2.3 Isometric Machine Translation

The Machine translation module allows users to generate transcripts, minutes, and topics in five languages. For all these languages, we provide users with isometric translation output. Isometric MT is the concept of generating translation output that falls within the range of $\pm 10\%$ of the Length Ratio (ratio of the target text and source text). This feature helps to generate synchronous outputs upon text-to-speech conversion. For implementing isometric translation, we develop a multitask learning model similar to Bhatnagar et al. (2022). We use fine-tuned OPUS-MT (Tiedemann and Thottingal, 2020) model for translation and fine-tuned mBART (Liu et al., 2020) for paraphrasing. We use WMT (Bojar et al., 2018) and MuST-C (Di Gangi et al., 2019) dataset for fine-tuning MT models, and PAWS-X (Yang et al., 2019), Opusparcus (Creutz, 2018) and Tapaco (Scherrer, 2020) dataset for paraphrasing. We utilize the IIT-B Hindi-English (Kunchukuttan et al., 2017) dataset for En-Hi translation.

As shown in Figure 6 our system architecture for Isometric Machine Translation utilizes prompt engineering technique during the machine translation & paraphrasing of input text. (i) For translating the input text, we try to maintain a target-to-source length ratio close to 1. This is worked out using the verbosity control feature, where in the finetuned OPUS-MT model tries to localize the source text based on the pre-calculated LR ratio using the predefined short, normal, long prompts. (ii) We also utilize the paraphrasing module to enhance the vocabulary and modify the length of already translated text. For generating output we apply "normal" prompt during the translation phase and reverse-prompts during the paraprhasing phase. Reverse-prompts are applied to alter the length of the translated sentence. We use similar method as done by Bhatnagar et al. (2022).

Language	BLEU Score	BERT Score	Length Ratio	Length Range (%)
de	29.9	0.83	1.05	51.95
it	34	0.84	1.04	57.03
fr	41.2	0.85	1.04	61.81
ru	21.7	0.83	0.97	62.47
hi	11.9	0.84	0.94	42.52

Table 3: Results obtained by the Isometric Machine Translation module on MuST-C test dataset, evaluated using the BLEU metric, BERT score, Length Ratio & Length Range.

We evaluate our MT system for the language de, fr, it & ru on the MuST-C test dataset and use the IIT-B Hindi-English test dataset for hi. As shown in Table 3, we are able to get high BERT score (Zhang et al., 2019b), length ratio, and length range for the language ru, fr & it. As shown in the table 4 we also compare our system using the BLEU metric evaluated on the same MuST-C dataset. We compare our results with that proposed by the Wang et al. (2020). As shown, our proposed system outperforms by a subsequent margin.

	fr	de	it	ru
fairseq S2T T-Sm	32.9	22.7	22.7	15.3
fairseq S2T Multi. T-Md	34.9	24.5	24.6	16.0
Ours	41.2	29.0	34	21.7

Table 4: BLEU Score evaluated on MuST-C test dataset and provides a systematic comparison between the fairseq S2T (Wang et al., 2020) and our proposed system. Here T-Sm and Multi. T-Md are both transformer-based models, later being trained jointly on 8 Languages.

2.4 Topics Modeling

DeepCon also provides a feature for automatic topic extraction based on Named Entity Recognition that extracts the top-k repeating n-grams from the transcripts. We use Yake[8] library for extracting named entities. Our system also supports multilinguality as a feature for generated keywords. Upon

[8]https://github.com/LIAAD/yake

generation, the system can apply translation across all the languages mentioned earlier.

3 Pilot Study

To assess DeepCon, we experiment it with IS1000a recording of the AMI Meeting Corpus (Carletta et al., 2005) that contains 4 speakers. The first step to use DeepCon is for users to login using their credentials. If users are not registered they can use the sign-up link[9] on the landing page. Next as shown in the figure 3 users can upload the *.mp3* or *.mp4* file and select the desired attributes. Users can also add name, email, and number of speakers. We have provided some advanced options that can control the length and translations of the audio file. Users can select the length of meeting-minutes from the three options: *short*, *medium*, and *long*. Users can also select amongst five languages for translation. Once users choose the appropriate options, they can upload the audio file via the upload button at the bottom and click on submit. After clicking the submit button, our front-end micro-service uploads the recording on the AWS S3 bucket and sends a *post request* to our back-end micro-service that contains the user's details along with the unique process code. Once this process is done, users get a confirmation email about the submission of the job. Once the back-end processing is completed, users again receive an email notification from our back-end microservice. This notification informs users that their processing is completed. Users can also download the outputs from our results page as seen in 4. The link of this page is also mailed to the user.

We gave the users a feedback form and question based on transcript quality, adequacy, grammaticality correctness, and fluency for English minutes. We received an average score of 4.57, 4.71, 4.57, and 4.85 out of 5, respectively. For MT quality estimation, we provided the users with the questions for quality in French, German, Russian, Hindi, and Italian Translation and received an average score of 5, 4.83, 4.8, 5, 4.6 out of 5, respectively. The average quality of generated topics is 4.85 out of 5, and the overall user interface received a score of 4.85 with an additional comment of "Improvements can be made in its ability to differentiate between the voice of respective speakers. Also, there was a slight deviation from the actual words in the transcript, that can be improved as well".

4 Design Choices

As mentioned in previous sections, we make use of microservice architecture. We utilize this methodology for three main reasons: As we utilize large pre-train models, it is not an efficient choice to do real-time processing for users. Because every meeting recording can vary in length and users can select many optimization options, real-time processing can take considerable time and slows down the website for other users. With recent advancements in DevOps technology like Kubernetes, microservice architecture has proven to be the most efficient, fault-tolerant, and robust deployment architecture. We use AWS EKS[10] as a container orchestration tool to deploy a highly scalable application. Microservice architecture also enables an easy development process of the application. As we have independent services, changing one will not affect others, and it can lead to a faster and more efficient software development process. We also plan to make this app open source, and we believe microservice architecture will enable developers to work in the module of their interest.

5 Conclusion and Future Work

In this paper, we present a tool for better management of meeting recordings by providing users the ability to generate meeting minutes, topics, and entities in six different languages. Our development architecture is designed in a way that it can be scaled and optimized if traffic increases on the application. We can also extend this application to accommodate more languages like Spanish, Romanian, Telugu, etc. The microservice architecture can further be abstracted by introducing a microservice for each functionality. This abstraction can result in more robustness and efficiency during high workloads.

This application can also be extended as an API service for developers to integrate in their systems. One major application can be of building native apps for online meeting platforms like Zoom, Microsoft Teams, and Google Meet

6 Acknowledgements

This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 833635 (project ROX-

[9] https://forms.gle/Wsbe6ASdggQNxjsHA

[10] https://aws.amazon.com/eks/

ANNE: Real-time network, text, and speaker analytics for combating organized crime, 2019-2022).

References

Aakash Bhatnagar, Nidhir Bhavsar, Muskaan Singh, and Petr Motlicek. 2022. Hierarchical multi-task learning framework for isometric-speech language translation. In *ACL*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, and Mccowan Wilfried Post Dennis Reidsma. 2005. The ami meeting corpus: A pre-announcement. In *In Proc. MLMI*, pages 28–39.

Mathias Creutz. 2018. Open subtitles paraphrase corpus for six languages. *CoRR*, abs/1809.06142.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Tirthankar Ghosal, Ondřej Bojar, Muskaan Singh, and Anja Nedoluzhko. 2021. Overview of the First Shared Task on Automatic Minuting (AutoMin) at Interspeech 2021. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 1–25.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *CoRR*, abs/1911.12237.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The IIT bombay english-hindi parallel corpus. *CoRR*, abs/1710.02855.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Parth Mahajan, Muskaan Singh, and Harpreet Singh. 2021. Team AutoMinuters @ AutoMin 2021: Leveraging state-of-the-art Text Summarization model to Generate Minutes using Transfer Learning. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 34–40.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *CoRR*, abs/1808.08745.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Yves Scherrer. 2020. TaPaCo: A corpus of sentential paraphrases for 73 languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6868–6873, Marseille, France. European Language Resources Association.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Oleg V. Vasilyev, Vedant Dharnidharka, and John Bohannon. 2020. Fill in the BLANC: human-free quality estimation of document summaries. *CoRR*, abs/2002.09836.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Miguel Pino. 2020. fairseq S2T: fast speech-to-text modeling with fairseq. *CoRR*, abs/2010.05171.

Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, Ken ichi Yokote, and Kenji Nagamatsu. 2021. Team Hitachi @ AutoMin 2021: Reference-free Automatic Minuting Pipeline with Argument Structure Construction over Topic-based Summarization. In *Proc. First Shared Task on Automatic Minuting at Interspeech 2021*, pages 41–48.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. *CoRR*, abs/1908.11828.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019a. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. *CoRR*, abs/1912.08777.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675.

A Appendix

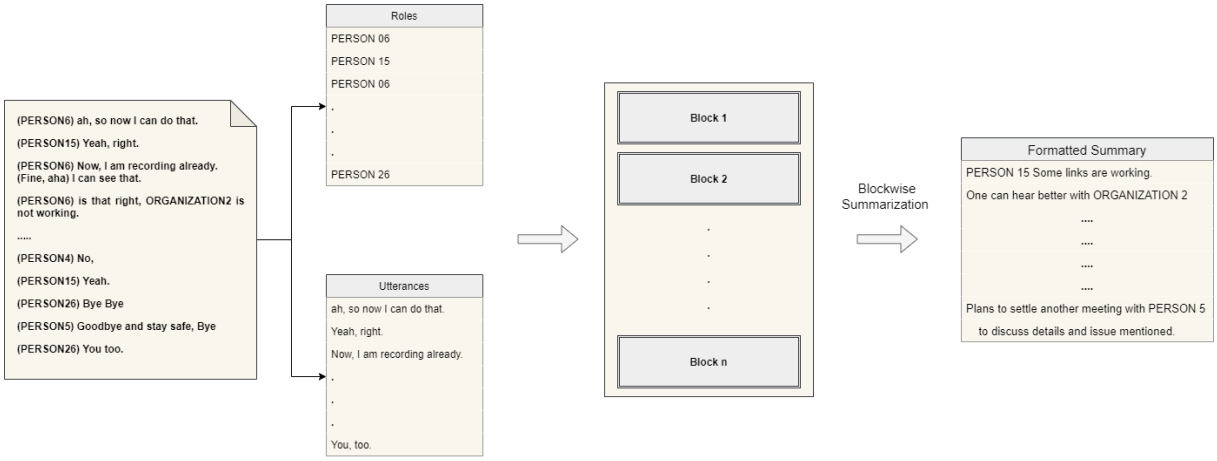


Figure 5: This figure is an architectural representation of the automatic minuting functionality. Here the ASR output is segmented into multiple chunks of text according to the sequence length accepted by the BART model. Next, the finetuned BART model process these chunks. Finally, we stack these chunks and perform redundancy elimination to generate the meeting minutes.

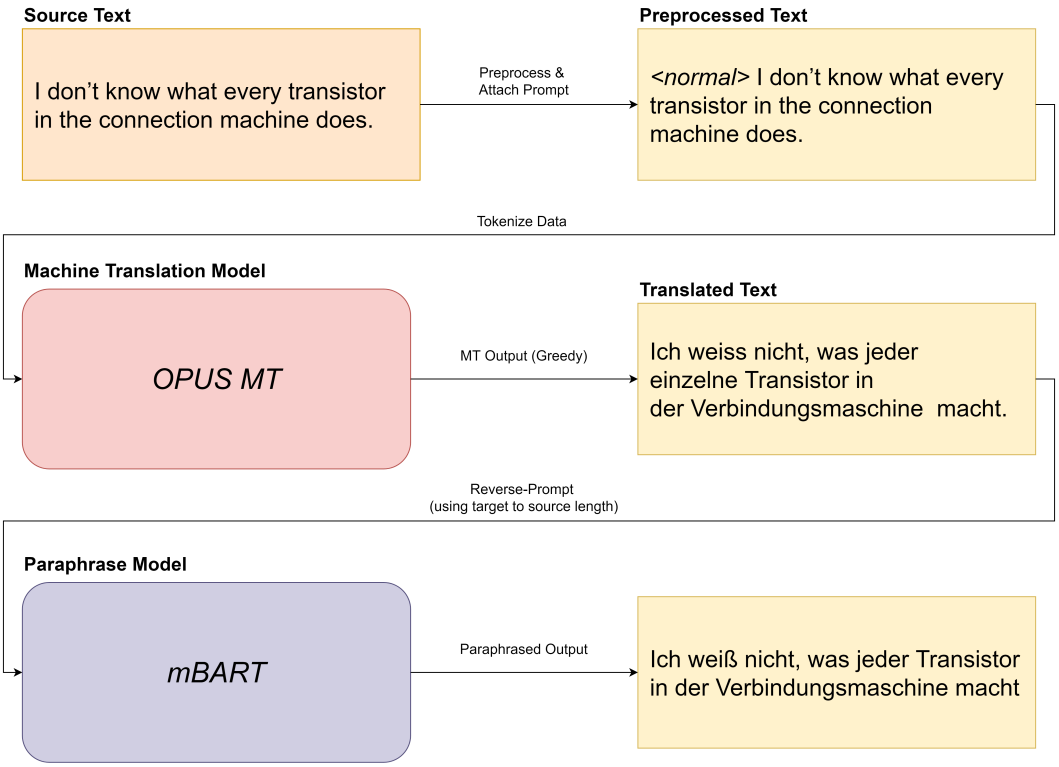


Figure 6: This figure shows the pipeline we use to generate isometric MT outputs from finetuned BART models. The first step, as seen, is attaching a 'normal' prompt to the input source sentence. This will help the translation model generate output with a length ratio close to 1. The next step is to attach a reverse prompt and send input to the paraphrasing module. Based on these reverse prompts, the paraphrasing module tries to shorten the long sentences and lengthen the short ones.

ICM : Intent and Conversational Mining from Conversation Logs

Sayantan Mitra, Roshni R. Ramnani, Sumit Ranjan and **Shubhashis Sengupta**
Accenture Labs, Bengaluru
{sayantan.a.mitra, roshni.r.ramnani,}@accenture.com
{sumit.b.ranjan, shubhashis.sengupta}@accenture.com

Abstract

Building conversation agents requires considerable manual effort in creating training data for intents / entities as well as mapping out extensive conversation flows. In this demonstration, we present ICM (Intent and Conversation Mining), a tool which can make the BOT build and update process much faster. ICM can be used to analyze existing conversation logs and help a bot designer to cluster, visualize and analyze customer intents; train custom intent models; and also to map and optimize conversation flows. The tool can be used for first time deployment or subsequent conversational flow updates in chatbots.

1 Introduction

In-spite of the proliferation of GUI based chatbot development environments and availability of open source and commercial tools with low code or no code environments, building chatbots remains a challenge. Many frameworks exist to help a non technical user build chatbots [1] [2], including mechanisms to enter the training data, and drag n drop methods for creating conversation flows. Similarly, research works focus on designing chat bots as end-to-end neural systems or using reinforcement learning based methods. However, limited work exists on techniques to automatically obtain and prepare training data by leveraging existing conversatons that can then be used by commercial tools building task-oriented chatbots.

In this paper , we discuss a tool that takes a user through a guided step by step process of clustering intents, reviewing the intent labels and conversation states, grouping of conversation flows, analyzing individual conversations and, finally exporting the training data for intents and conversation flow. This information can be used by non technical chatbot developers to create chatbots using any of the available chatbot building tools.

The rest of the paper is organized as follows: In section 2 we discuss some of the related work in this space. Section 3 high lights the key features in our tool. Section 4 discusses the technical details of the tool including the algorithms used. 5 discusses the key aspects of the demo. Finally, in section 6 we conclude the paper.

2 Related Work

Multiple methods have been proposed in using realistic data for developing chatbots.Wirén et al. (2007) suggest a modified version of the wizard of the oz approach by collecting transcripts of real conversations between service agents and customers. Many bots are being built to augment human service agents, and hence there is a rich set of information available as human (customer) - human (service agent) conversations.The tool Graph2Bot (Bouraoui et al., 2019) analyzes such existing conversations but fails to create a format that can be leveraged by commercial tools.

In the absence of human conversation logs, text in the form of emails and Service Now tickets can also provide insights about the queries that can possibly be handled by the chatbot. Mallinar et al. (2019) provide a mechanism to bootstrap conversational agents by helping select the necessary training samples.

Once a chatbot is deployed, there are existing tools that perform various forms of conversation analytics. However, these tools do not provide a direct mechanism to leverage these insights back into the enhancement of the chatbot.

3 Key Steps in Analyzing Intent and Conversations

ICM enables multiple people supporting all phases of the chatbot development process to identify the

[1] https://dialogflow.cloud.google.com
[2] https://botmock.com/

Proceedings of the SIGdial 2022 Conference, pages 403–406
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

key intents, the conversation flows, conversation flow analytics including volumetric and temporal analyses, the user sentiment and emotion as well as evolution of conversations over time. This is done in an offline manner by importing the conversation logs. The conversation logs may be human-human conversations captured at the beginning of the chatbot development life cycle, and / or human-bot based conversations at the run phase of the chatbot life cycle. The key features of the tool are as follows:

3.1 Intent Discovery

This is the mechanism by which an automatically extracted short description is used to cluster the conversations in a semi-supervised way. The user is then allowed to select or modify an automatically generated intent label and export the training examples applicable for each intent.

3.2 Intent Analysis

This screen allows the user to view detailed charts on the volumetric analysis (numbers, intensity), temporal (time-of-day, periodicity) characteristics etc. of the intents found.

3.3 Conversation Analysis

The user can view the combined conversation flows per intent or across intents. Through this screen, an analyst can analyze each conversation state, understand the most common flows through the system, identify bottle necks etc.

4 ICM : Technical Details

The tool contains a front end for labelling, analyzing and reviewing existing conversations, as well as the backend containing a rich set of clustering algorithm options, conversation summarization, sentiment and emotion detection options. Figure 1 shows the high level diagram of the tool.

4.1 Front End

The user can upload existing conversation logs or other text data in the form of emails etc via a simple CSV or excel file. The column containing the short description of the content must be identified. The short description, if not present, is generated by using the module described in Subsection **"Conversation Description"**. Privately Identifiable Information is anonymized separately using a custom

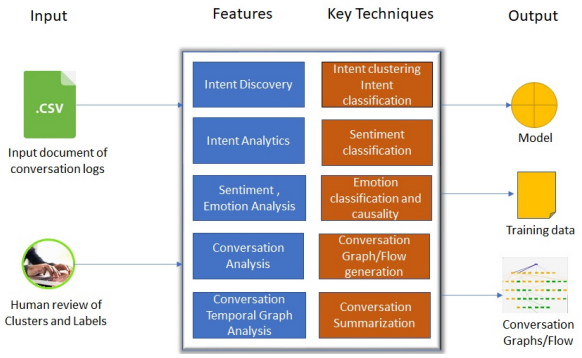


Figure 1: High Level Diagram of ICM

.

python script with regex and spaCy[3] NER model. At the front end, the user can select from a list of clustering algorithms, language models and clustering parameters. The clustered intents, conversation states and flows can be reviewed and labelled in the user interface. The user can export the generated labelled data and also conversation graphs and flows from the front end. Further, the front end also provides interactive visuals for comparison of conversation flows for the users.

4.2 Backend

4.2.1 Conversation Summarization

We use a summarization module based on BART (Lewis et al., 2019) trained on Samsum (Gliwa et al., 2019) data available in the transformers library.

4.2.2 Intent Clustering

The information uploaded into the system goes through three key steps: 1) The user must choose the clustering algorithm algorithm (ITER-DBSCAN (Chatterjee and Sengupta, 2021),HDBSCAN (McInnes et al., 2017)) , sentence embedding (BERT,USE, mBERT), select optional dimentionality reduction (UMAP) and other hyper parameters. The user can run the algorithm multiple times with different configurations and choose the best based on the coverage and the homogeneity of the clusters formed. 2) For each cluster the system provides label suggestions. These are done by using a combination of terms obtained using TD-IDF and the top 5 occurring skip3grams. 3) The system marks similar clusters by calculating the centroid of each cluster and finding the cosine similarity.

[3] https://github.com/explosion/spaCy

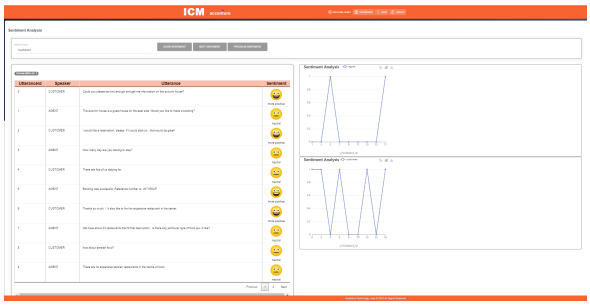


Figure 2: Screenshot of Sentiment analysis.

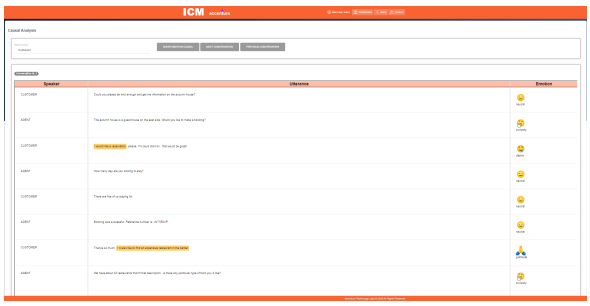


Figure 3: Screenshot of causal analysis.

4.2.3 Sentiment , Emotion and Causality

The tool can identify the user sentiment and emotion per utterance and the causality of each emotion. The sentiment analysis module classifies each utterance into positive, negative and neutral. We use the architecture described by Munikar in (Munikar et al., 2019) trained on the ScenarioSA dataset. The sentiment graph shows the change in sentiment throughout the conversation for Agent and Customer (Fig 2). The emotion analysis module identifies the emotion across 27 categories by using BERT trained on the GOEmotions dataset (Demszky et al., 2020). The causality of each emotion is determined by RECCON (Poria et al., 2020) (Fig. 3).

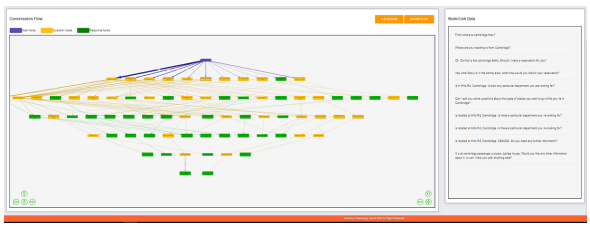


Figure 4: Screenshot of Conversation flow.

Figure 5: Screenshot of temporal conversation graph.

4.3 Conversation Graph and Flow Generation

This module is active only when the uploaded file has the conversation data[4]. Once the file is uploaded into the system, the backend uses a pretrained CRF based Dialogue Act Classifier (DAC) model to extract the relevant [5] AGENT and CUSTOMER utterances from each conversation. The extracted AGENT utterances are divided into two separate files, viz., questions and responses. Clustering and labelling are done on these two files. After this process, we have labels for each of the relevant AGENT utterances for every conversation.

To generate the conversation graph , we assign a edge between two AGENT state label if there is a transition. For example, the AGENT current state/utterance label is *ques-booking-enquiry* and the next available state is *res-booking-confirm*, so there is a edge between these two former labels and the edge value is the CUSTOMER utterances between these two states. It is a fully connected graph.

To generate a conversation flow (Fig.4), we generated a tree structure. We followed the similar approach as discussed above. The only difference is in the connections. In conversation graph, if the transition is from *ques-booking-enquiry* $\rightarrow$ *res-booking-confirm* $\rightarrow$ *ques-booking-enquiry*, we will end up with a loop. But in conversation flow the two *ques-booking-enquiry* are treated separately, that is *ques-booking-enquiry* in level 1 (say) of the tree is different from *ques-booking-enquiry* in level 2 (say). In conversation flow, we also calculate the weight of the edges . For examples, if for all the conversations there is 5 transition between two states then the weight of the edge becomes 5.

The tool can also generate temporal conversation flows (Fig.5) for each intent. This helps the end

[4]Here, we assume the conversation is between AGENT and CUSTOMER.

[5]Extracts only *{QUESTION, COMMAND, INFO}* type utterances and discard other types like *GREETINGS*

Industry Type*	Total Conversations	Identified intents	Note
Telecom1	14000	12	Client wanted to find out the initial conversational flows to increase the containment rate of the conversation. Conversation flow structure generated through ICM are validated by conversational designers of the client.
Telecom2	5000	9	Client shared 5k conversation to determine the intents of the conversation.
Consumer Health	27000	175	Client shared 27k conversation to determine the intents of the conversation.

Table 1: Statistics of ICM tool. *Due to company policy, client names are not disclosed.

user to understand the change in conversation flow either for a single intent over different time period or for different intents over same time period.

5 Demonstration

We will demonstrate ICM on an open source dataset, Multiwoz. During the demo, the audience will see how a user can: a) Upload a conversation dataset, select the appropriate clustering algorithm, language model, and other clustering parameters. b) Label, review and verify the cluster labels and conversation states c) View utterance level details like sentiment, emotion and emotion causality d) View the Conversation Graphs and Trees including the temporal analysis e) View training data and graphs can be exported.

6 Conclusion

In this paper, we highlight a key gap in the existing technology used to build chatbots - the ability to leverage existing data in the form of human-human or human-bot conversations automatically. We discuss a the tool that enables an end user to analyze this data, derive detailed and varied insights and export it in a form that can be leveraged by existing technology to build chatbots.

References

Jean Léon Bouraoui, Sonia Le Meitour, Romain Carbou, Lina M Rojas Barahona, and Vincent Lemaire. 2019. Graph2bots, unsupervised assistance for designing chatbots. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 114–117.

Ajay Chatterjee and Shubhashis Sengupta. 2021. Intent mining from past conversations for conversational agent.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Neil Mallinar, Abhishek Shah, Rajendra Ugrani, Ayush Gupta, Manikandan Gurusankar, Tin Kam Ho, Q Vera Liao, Yunfeng Zhang, Rachel KE Bellamy, Robert Yates, et al. 2019. Bootstrapping conversational agents with weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9528–9533.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11):205.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert.

Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu, Romila Ghosh, Niyati Chhaya, Alexander F. Gelbukh, and Rada Mihalcea. 2020. Recognizing emotion cause in conversations. *ArXiv*, abs/2012.11820.

Mats Wirén, Robert Eklund, Fredrik Engberg, and Johan Westermark. 2007. Experiences of an in-service wizard-of-oz data collection for the deployment of a call-routing application. In *Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, NAACL-HLT, Rochester, NY, April 2007.*, pages 56–63. Omnipress.

Entity-based De-noising Modeling for Controllable Dialogue Summarization

Zhengyuan Liu[†‡], **Nancy F. Chen**[†‡]
[†]Institute for Infocomm Research, A*STAR, Singapore
[‡]CNRS@CREATE, Singapore
{liu_zhengyuan,nfychen}@i2r.a-star.edu.sg

Abstract

Although fine-tuning pre-trained backbones produces fluent and grammatically-correct text in various language generation tasks, factual consistency in abstractive summarization remains challenging. This challenge is especially thorny for dialogue summarization, where neural models often make inaccurate associations between personal named entities and their respective actions. To tackle this type of hallucination, we present an entity-based de-noising model via text perturbation on reference summaries. We then apply this proposed approach in beam search validation, conditional training augmentation, and inference post-editing. Experimental results on the SAMSum corpus show that state-of-the-art models equipped with our proposed method achieve generation quality improvement in both automatic evaluation and human assessment.

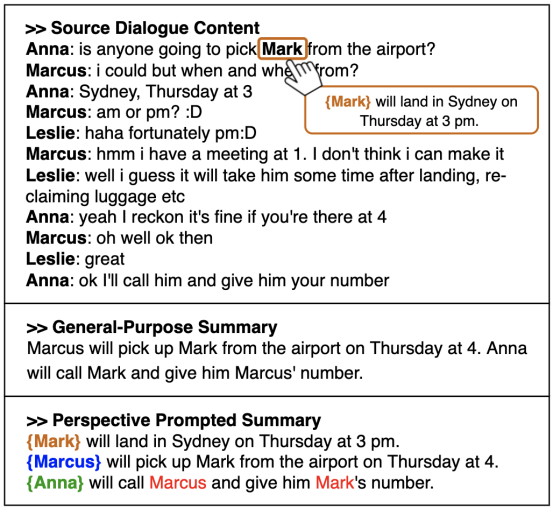


Figure 1: Dialogue summarization examples generated with a general purpose and perspective prompts (labeled in bracket). Note that controllable summaries start with the specified personal named entity's perspective.

1 Introduction

Abstractive dialogue summarization is an emerging research area (Goo and Chen, 2018; Chen et al., 2021). While the data size of available corpora is smaller than that for monological summarization (Carletta et al., 2005; Gliwa et al., 2019), neural approaches have shown promising potential to generate fluent outputs via fine-tuning large-scale contextualized language backbones (Chen and Yang, 2020; Feng et al., 2021). In most corpus constructed for text summarization, only one reference summary is annotated, and models trained via supervised learning on such corpora provide summaries in a general-purpose manner. However, in practice, the generic text summarizers cannot meet the requirements of certain applications and use cases (Fan et al., 2018; Goodwin et al., 2020). For instance, when generating minutes for meeting transcripts, users have their preferences on different personal perspectives. In this case, controllable summarization provides a flexible solution (He et al., 2020) since it allows users to obtain diverse generations. As the aim of dialogue summaries often focuses on *"who did what"* and their narrative flow usually starts with a subject (often persons), the generation process can be modulated by personal named entity planning or prompts (Liu and Chen, 2021). For example, as shown in Figure 1, a controllable system can produce different summaries based on the specific perspective prompts.[1]

However, neural abstractive models often suffer from hallucinations, which lower the reliability of automatic summarization (Zhao et al., 2020; Zhang et al., 2020). In dialogue summarization, this issue commonly involves misaligned personal named entity associations (Lee et al., 2021; Liu et al., 2021b). For instance, as shown in Figure 1, the model upon the prompt *'Anna'* generates the description *"Anna will call Marcus and give him Mark's number"*. While this sentence achieves a high score in word-

[1]Here we use 'prompt' (namely a text conditional signal) under conditional language generation, which is distinct from the task anchor formulated in few-shot/zero-shot 'prompt-based learning' (Liu et al., 2021a).

Proceedings of the SIGdial 2022 Conference, pages 407–418
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

overlapping metrics such as ROUGE (Lin, 2004), the semantic meaning it conveys is incorrect (according to the conversation, the personal named entities *'Mark'* and *'Marcus'* (colored in red) are misassigned). Such factual inconsistency, the inability to adhere to facts from the source, is a prevalent and unsolved problem (Kryscinski et al., 2019). This limitation is more substantial in controllable scenarios, as models are *required* to condense and paraphrase important contextual information from various personal perspectives.

In this work, we focus on improving the accuracy of personal named entity assignment. Given a source dialogue content, detecting and correcting the errors in a generated summary is similar to the de-noising process adopted in sequence-to-sequence language modeling schemes (Lewis et al., 2020). Therefore, we build an entity-based de-noising model for dialogue summarization via reference summary perturbation and recovery. We then leverage this de-noising model to improve controllable dialogue summarization: (1) At the training stage, we use the de-noising model as a discriminator, to validate beam search candidates under different prompts, and generate factually consistent summaries. Then the validated summaries are added to the training set, which serves as conditional training augmentation. (2) At the inference stage, we use the de-noising model as a corrector, to amend the generated summaries via post-editing. This approach can also be applied to other generic and controllable dialogue summarizers. Experiments are conducted on SAMSum (Gliwa et al., 2019), which consists of multi-turn dialogues and human-written summaries. Empirical results show that our proposed method reduces personal named entity misassignment and achieves improved generation quality on both automatic measures and human evaluation.

2 Related Work

Text summarization is studied in extractive and abstractive paradigms (Gehrmann et al., 2018). In extractive studies, non-neural approaches utilize various linguistic and statistical features via lexical (Kupiec et al., 1995) and graph-based modeling (Erkan and Radev, 2004), and neural approaches bring about substantial improvements via feature-rich distributional representation and hierarchical context modeling (Nallapati et al., 2017; Kedzie et al., 2018). In contrast, abstractive approaches are expected to generate more concise and fluent summaries, which brings about different technical challenges. To foster end-to-end data-driven methods, corpora in news domain (e.g., CNN/Daily Mail (Hermann et al., 2015), NYT (Sandhaus, 2008)) are constructed, and sophisticated neural architectures for abstractive summarization are proposed, such as LSTM-based encoding-decoding (Rush et al., 2015), pointer-generator networks (See et al., 2017), hybrid extractive-abstractive summarizer Gehrmann et al. (2018), and fine-tuning large-scale pre-trained language models (Liu and Lapata, 2019; Lewis et al., 2020). Recently, datasets for summarizing conversations are constructed from meetings (Zhong et al., 2021) or daily chats (Gliwa et al., 2019). Based on the linguistic features of human conversations, many studies pay attention to utilizing conversational analysis for dialogue summarization, such as leveraging dialogue acts (Goo and Chen, 2018), multi-modal features (Li et al., 2019), topic information (Liu et al., 2019), coreference (Liu et al., 2021b), and fine-grained view segmentation with hierarchical modeling (Chen and Yang, 2020).

Controllable language generation introduces auxiliary signals to obtain diverse or task-specific outputs. Such tasks include text style transfer (Shen et al., 2017) and paraphrasing (Iyyer et al., 2018). There are various conditional signal formats, such as categorical labels (Hu et al., 2017), latent representations, semantic or syntactic exemplars (Gupta et al., 2020), and keyword planning (Hua and Wang, 2020). For controllable text summarization, He et al. (2020) and Dou et al. (2021) proposed two generic frameworks in news domain with length constraint and question/entity indicators, and Liu and Chen (2021) proposed personal named entity planning by leveraging the common narrative flow of dialogue summarization.

Tackling hallucinations in abstractive summarization is an essential research topic in making such summaries applicable to real-world scenarios (Kryscinski et al., 2019; Zhao et al., 2020). Reinforcement approaches proposed using factual consistency as optimization reward (Zhang et al., 2020) and post-editing approaches (Kryscinski et al., 2020) focus on correcting summary of general news corpora or facts extracted from an external knowledge base (Iso et al., 2020). For dialogue summarization, Liu and Chen (2021) proposed a binary classifier to detect personal named entity in-

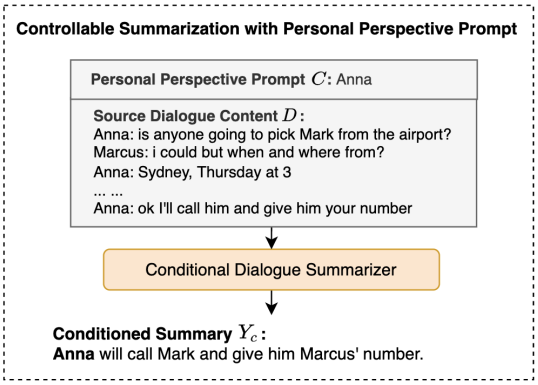


Figure 2: Overview of controllable summarization process. One specific personal named entity is fed to the summarizer as conditional signal.

consistency. Recently, Lee et al. (2021) proposed a post-correction model that can discriminate which type of speaker inconsistency, and revise the output accordingly. In this work, to the best of our knowledge, we are the first to exploit an entity-based de-noising model for abstractive dialogue summarization in both training and inference stages.

3 Controllable Dialogue Summarization

3.1 Task Definition

Here we assume that the input consists of two entries in the controllable setting: a source dialogue D, and a prompt C. The output is the summary text Y, which is a condensed version of the source content D, and starts with the prompt C. Unlike the general-purpose summarization task (Hermann et al., 2015; Gliwa et al., 2019), given one instance of D, the summary Y can be manifested as various outputs conditioned on different choices of C, and are expected to be fluent and factually correct.

3.2 Conditional Entity-based Prompt

In previous studies on controllable document summarization, conditional signals in the form of keywords or descriptive prompts are investigated, and extracted from the source document (He et al., 2020). To summarize multi-turn dialogues, personal named entities that occur in the conversation can be used to form the prompt C for conditional generation (Liu and Chen, 2021). For instance, when writing meeting minutes, with a controllable system, users can obtain diverse generations by choosing different personal named entities, as shown in Figure 1.

In this work, we use the **single entity prompt** for controllable dialogue summarization, as shown in Figure 2. In our reference summary analysis of the SAMSum corpus (Gliwa et al., 2019), the average number of personal named entities (e.g., speaker roles, mentioned persons) in a source dialogue is 2.89. Among these dialogues, 90% human-written summaries start with a personal named entity. In particular, we observed that there is a positional correlation between entity prompts in reference summaries and their occurrence in the source content. As shown in Figure 3, 62% reference summaries start with the first occurred personal named entities in the conversation. This number reaches 94% when we count the first two personal named entities. Therefore, the general-purpose summarizer will follow the same narrative style (namely start with the first speaker or mentioned person), which shares a similar parallel with the position-bias phenomenon studied in news summarization (Kryscinski et al., 2019). Moreover, this positional distribution demonstrates the limited annotation diversity if we only use the reference summary for conditional training.

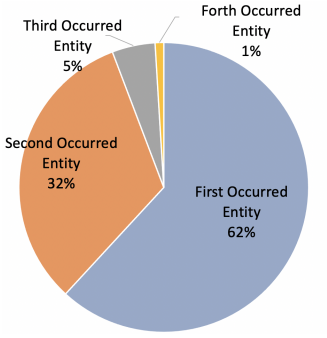


Figure 3: Positional distribution of the personal named entity prompt of reference summaries and their occurrence in the source content.

3.3 Controllable Neural Summarizer

A neural sequence-to-sequence network is applied to build the controllable dialogue summarizer. Its base architecture is a Transformer-based encoding-decoding model, since Transformer (Vaswani et al., 2017) is widely adopted in various natural language processing tasks due to its superior generation performance (Devlin et al., 2019; Lewis et al., 2020).

Encoder: The encoder consists of a stack of Transformer layers. Each layer has two sub-components: a multi-head layer with a self-attention mechanism, and a position-wise feed-forward layer (Equation 1). A residual connection is employed between each pair of the two sub-components, followed by layer normalization (Equation 2).

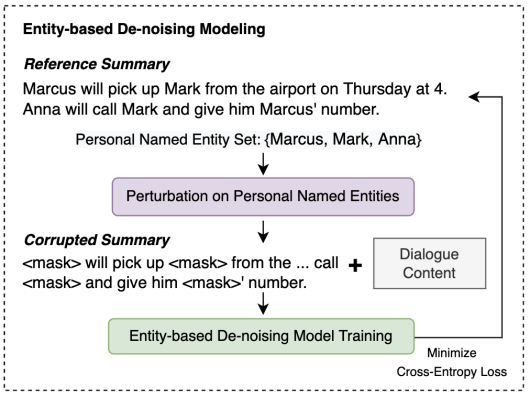


Figure 4: Overview of the entity-based de-noising model. Entity-based text perturbation is conducted on the reference summaries.

$$\widetilde{h}^l = \text{LayerNorm}(h^{l-1} + \text{MHAtt}(h^{l-1})) \quad (1)$$

$$h^l = \text{LayerNorm}(\widetilde{h}^l + \text{FFN}(\widetilde{h}^l)) \quad (2)$$

where l represents the depth of stacked layers, and h^0 is the embedded input sequence. MHAtt, FNN, LayerNorm are multi-head attention, feed-forward and layer normalization components, respectively.

Decoder: The decoder is also a stack of Transformer layers. Aside from the two sub-components in encoding layers, the decoder has another component that performs a multi-head attention over hidden representations from the last encoding layer. Then, the decoder generates tokens from left to right in an auto-regressive manner. Full neural architecture and formula details are described in (Vaswani et al., 2017).

At the training stage, the prompt $C = \{c_0, c_1, .., c_m\}$[2] is concatenated with the source content $D = \{w_0, w_1, ..., w_n\}$ as input, and it is represented as *[<BOS>, C, <EOS>, <BOS>, D, <EOS>]*.[3] To better model utterance boundary representation, we added a special token *'<u>'* as the utterance delimiter in D. [4] The summarizer learns to generate the ground truth $Y = \{y_0, y_1, ..., y_t\}$ by condensing the information of dialogue context conditioned on the prompt. The loss of maximizing the log-likelihood on the ground truth is formulated as:

$$loss(\theta) = -\Sigma_i \log(p(y_i|y_{<i}, D, C; \theta)) \quad (3)$$

[2]In our setting, while the prompt is a single personal named entity, it can be multiple tokens after the subword tokenization.

[3]Tokens of <BOS> and <EOS> defined in *'BART-large'* are <s> and </s> respectively, and can be changed according to other language backbones.

[4]The special token *'<u>'* is added to the vocabulary, and we initialize its token embedding by averaging the embedding vectors of '<s>', comma, and period.

Sample Type	Number
Training Set (14732 Samples)	
Mean/Std. of Dialogue Turns	11.7 (6.45)
Mean/Std. of Dialogue Length	124.5 (94.2)
Mean/Std. of Summary Length	23.44 (12.72)
Validation Set (818 Samples)	
Mean/Std. of Dialogue Turns	10.83 (6.37)
Mean/Std. of Dialogue Length	121.6 (94.6)
Mean/Std. of Summary Length	23.42 (12.71)
Test Set (819 Samples)	
Mean/Std. of Dialogue Turns	11.25 (6.35)
Mean/Std. of Dialogue Length	126.7 (95.7)
Mean/Std. of Summary Length	23.12 (12.20)

Table 1: Data Statistics of the dialogue summarization dataset SAMSum (Gliwa et al., 2019).

where D, C, y, θ denotes the dialogue content, conditional prompt, targeted summary sequence, and the trainable parameter set, respectively. i is decoding time-step, and ranges from 1 to t. During inference, the model creates a summary based on a specific perspective prompt, and is coherent with the context of the input conversation.

4 Entity-based De-noising Modeling

While existing abstractive neural models achieve state-of-the-art performance on quantitative evaluation, factual inconsistency remains a prevalent and unsolved problem (Kryscinski et al., 2019; Zhang et al., 2020). In both document and dialogue summarization, it has been demonstrated that a certain proportion of abstractive summaries contain hallucinated statements (Zhao et al., 2020; Khalifa et al., 2021). Such hallucinations raise concerns about the usefulness and reliability of automatic summarization, and are challenging to eradicate in neural approaches due to the implicit nature of learning representations.

In dialogue summarization, the misassignment of personal named entities significantly affects generation quality (Lee et al., 2021; Liu and Chen, 2021). Inspired by the de-noising sequence-to-sequence pre-training schemes (Lewis et al., 2020; Raffel et al., 2020), here we propose an entity-based de-noising model to detect and recover the incorrect personal named entity tokens. Compared with the binary classifier for factual inconsistency (Liu and Chen, 2021), the sequence-to-sequence framework supports revising the summaries via post-editing.

4.1 De-noising Sample Construction

To construct training samples for entity-based de-noising, we conduct text perturbation on the ref-

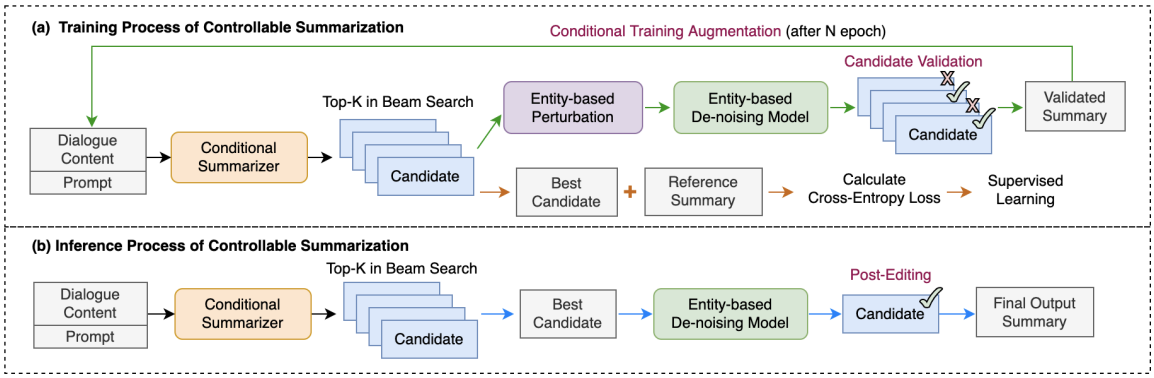


Figure 5: Overview of the controllable summarization framework equipped with entity-based de-noising modeling: (a) Training process with supervised learning (in orange arrow), and with beam search validation and conditional sample augmentation (in green arrow); (b) Inference process with post editing (in blue arrow).

erence summaries. As shown in Figure 4, given a reference summary Y, we obtain a corrupted version $\widetilde{Y}$ via entity masking and substitution. More specifically, we first extract the full list of personal named entities from each source dialogue, and then mask them or replace them with another entity at a random rate (p_{noise}=0.5). Additionally, to reduce the positional imbalance caused by labeling correlation described in Section 3.2, we shuffle the summary sentences at a random rate (p_{shuffle}=0.5) .

4.2 De-noising Model Training

We fine-tuned the sequence-to-sequence language backbone *BART-large* (Lewis et al., 2020) for de-noising modeling. Given a dialogue D and a corrupted summary $\widetilde{Y}$, the input is represented as *[*<BOS>, $\widetilde{Y}$, <EOS>, <BOS>, D, <EOS>*]*.[5] As shown in Figure 4, the model is applied to generate the reference summary Y, and is optimized by minimizing the cross-entropy loss. Since the text perturbation is conducted especially on personal named entities, it encourages the de-noising backbone to model features such as *"who-did-what"* and speaker interactions. Moreover, unlike the left-to-right auto-regressive summary generation, the de-noising backbone can utilize the bi-directional context of both dialogue and summary sequence, and it achieves a 0.92 sample-level accuracy on the validation set, which is a reasonable performance for follow-up steps.

5 Leveraging De-noising Modeling

In this section, we then elaborate on how to leverage the entity-based de-noising model for controllable dialogue summarization.

5.1 Beam Search Candidate Validation

The de-noising model can be used as a reference-free discriminator to validate the beam search candidates. Following previous work on two-stage summary ranking (Liu and Liu, 2021), we use diverse beam search (Vijayakumar et al., 2016) as the sampling strategy. As shown in Figure 5 (a) and Figure 6, for each candidate generated in beam search, we mask all the personal named entities, and feed it to the de-noising model. If the recovered output is identical to the unaltered candidate, it is regarded as a validated summary without any personal named entity misassignment. Moreover, given a pair of names concatenated with '*and*', we consider their permutations are the same (e.g, '*Tom and John*', '*John and Tom*').

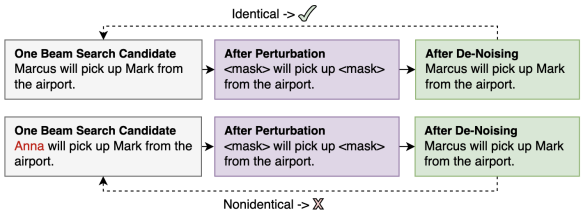


Figure 6: One example of beam search candidate validation. Two beam search candidates are validated by the de-noising discriminator, and one misassignment is detected ('Anna' in red).

5.2 Conditional Training Augmentation

One major challenge of training models for controllable dialogue summarization is the lack of diverse annotation, as each source content only has one reference summary in existing corpora. Moreover, due to the positional correlation of entity prompts

[5]Tokens of <BOS> and <EOS> defined in *'BART-large'* are <s> and </s> respectively, and can be changed according to other language backbones.

in human-written summaries (Section 3.2), their corresponding conditional samples will present an imbalanced prompt distribution, and cause unnecessary inductive bias in data-driven approaches.

In this work, we address this issue by introducing weak self-supervision (Karamanolakis et al., 2021), and use the summarizer's intermediate generation as additional training samples. In other controllable language generation studies like text style transfer, self-supervised sample selection adopts metrics such as sentiment polarity score (Luo et al., 2019); here we use the entity-based consistency. As shown in Figure 5 (a), after N training epoch, we re-run the model on the original training set, obtain summaries upon perspective prompts which are distinct from that of the reference, and validate them by the de-noising model (as in Section 5.1), then the validated samples which rank highest in beam search are used as additional training data. In our experiments on SAMSum, we conducted the augmentation from the third epoch (when the summarizer produces reasonable results with automatic metrics), and 30% of the training set contribute a conditional augmented sample.

5.3 Inference with Post-Editing

In addition, since the de-noising model learns to correct the entity-based perturbation, it can also be used for summary post-editing, which is an effective method to improve the generation quality commonly applied in machine translation (Popović and Arčan, 2016). As shown in Figure 5 (b), at the inference stage, the best candidate selected from beam search is fed to the de-noising model, then we obtain the final summary where the misassigned entities are corrected. It is noteworthy to mention that, since the post-editing here focuses on personal named entity correction, it is not straightforward to observe the performance improvement via automatic evaluation metrics such as ROUGE, and we thus conduct a human evaluation. Moreover, the post-editing is a general process to extend to other dialogue summarization systems (see experimental results in Section 6.6).

6 Experiments and Results

6.1 Experimental Corpus

Experiments are conducted on SAMSum (Gliwa et al., 2019), which contains multi-turn daily conversations with human-written summaries in a general-purpose manner. Details of the dataset are shown in Table 1. We retain the original text content of conversations such as cased words, emoticons, and special tokens, and pre-process them using sub-word tokenization (Lewis et al., 2020). Since the positional embedding of our Transformer-based model can support 1,024 input length, none of the samples are truncated.

6.2 De-noising Model Configuration

The *'BART-large'* (Lewis et al., 2020) is used to build the entity-based de-noising model. The number of encoder layers, decoder layers, input, and hidden dimension and $12/12/1024$, respectively. The learning rate was set at $2e{-}5$. *AdamW* optimizer (Loshchilov and Hutter, 2019) was used with weight decay of $1e{-}3$ and a linear scheduler. Drop-out (Srivastava et al., 2014) ($rate$=0.1) was set as in the original *BART* configuration. Text perturbation described in Section 4.1 is conducted on the SAMSum dataset for training and validation.

6.3 Summarization Model Configuration

For controllable dialogue summarization, the language backbone *BART* (Lewis et al., 2020) is applied. The number of encoder layers, decoder layers, input and hidden dimension are $6/6/768$ for the *'BART-base'*, and $12/12/1024$ for the *'BART-large'* and *'CTRLsum'*. *AdamW* optimizer (Loshchilov and Hutter, 2019) was used with learning rate of $3e{-}5$, weight decay of $1e{-}3$, and a linear learning rate scheduler. Drop-out (Srivastava et al., 2014) rate was set at 0.1. Diverse beam search (Vijayakumar et al., 2016) is adopted with group number 5 and beam size 10. For augmentation samples, we added a weighted loss (λ=0.15).

The trainable parameter size is 139M of the *'BART-base'*, and 406M of the *'BART-large'*. Batch size and epoch number were set at 8. Best checkpoints were selected based on validation results of ROUGE-2 F1 score. All models were implemented with PyTorch (Paszke et al., 2019) and HuggingFace Transformers[6]. All experiments were running on a single Tesla A100 GPU with 40G memory.

6.4 Evaluation Metrics

Extensive metrics are used for quantitative evaluation: (1) We adopt **ROUGE-1**, **ROUGE-2**, and **ROUGE-L** (Lin, 2004), which are customary in summarization tasks via counting lexical overlap, and `Py-rouge` package is employed following

[6]https://github.com/huggingface/transformers

	ROUGE-1			ROUGE-2			ROUGE-L			SimCSE	B-Score	EACC
	F	P	R	F	P	R	F	P	R			
CTRLsum BART$_{large}$ (CNN/DM)	33.8	43.5	32.6	10.5	14.2	10.1	33.6	41.1	32.1	61.8	-7.19	71.2
CTRLsum BART$_{large}$ (SAMSum)	53.8	55.9	57.6	29.9	31.2	31.9	52.4	53.6	55.1	79.2	-4.79	84.2
+ Beam Search Valid-Augment	54.2	57.7	56.4	30.3	32.9	31.3	52.9	55.4	54.3	79.4	-4.71	88.7
Conditional BART-base	51.3	57.1	51.8	27.2	30.4	27.5	50.4	54.6	50.3	76.2	-5.36	74.4
+ Beam Search Valid-Augment	51.5	58.4	50.7	27.6	31.4	27.3	50.7	56.1	49.7	76.5	-5.20	79.8
Conditional BART-large	53.8	61.6	52.6	30.2	35.1	29.4	52.9	59.1	51.5	78.4	-5.23	84.7
+ Beam Search Valid-Augment	54.2	58.8	55.2	30.4	33.4	30.8	53.0	56.5	53.5	78.9	-4.98	86.2

Table 2: **Results on reference prompt generation** (matching training and test condition). *F*, *P*, *R* are F1 measure, precision, and recall. *B-Score* and *EACC* denotes BARTScore (Yuan et al., 2021) and entity-based accuracy. *CTRLsum* is a generic controllable summarizer for news (He et al., 2020), and we further fine-tuned it on SAMsum.

	ROUGE-1			ROUGE-2			ROUGE-L			SimCSE	B-Score	EACC
	F	P	R	F	P	R	F	P	R			
CTRLsum BART$_{large}$ (CNN/DM)	34.8	50.1	32.7	10.7	17.7	9.7	31.7	42.8	29.2	60.0	-7.99	71.7
CTRLsum BART$_{large}$ (SAMSum)	54.3	56.9	57.8	28.0	29.3	29.9	50.5	52.1	52.9	78.8	-4.85	67.8
+ Beam Search Valid-Augment	55.1	58.2	56.5	28.7	30.7	29.4	50.7	52.8	51.7	79.0	-4.82	77.3
Conditional BART-base	51.5	58.2	50.1	29.4	29.7	25.4	50.9	56.4	48.5	75.0	-5.44	64.1
+ Beam Search Valid-Augment	52.6	59.4	50.7	28.0	31.5	27.1	50.4	56.3	49.1	75.8	-5.23	72.2
Conditional BART-large	55.1	62.3	53.3	29.5	33.5	29.3	53.2	59.2	52.3	78.2	-5.04	68.3
+ Beam Search Valid-Augment	55.7	61.5	56.2	30.2	32.5	30.5	53.6	57.1	53.7	79.5	-4.91	77.2

Table 3: **Results on distinct prompt generation** (simulating a practical use case). *F*, *P*, *R* are F1 measure, precision, and recall. *B-Score* and *EACC* denotes BARTScore (Yuan et al., 2021) and entity-based accuracy.

	ROUGE-1			ROUGE-2			ROUGE-L			SimCSE	B-Score	EACC
	F	P	R	F	P	R	F	P	R			
CTRLsum BART$_{large}$ (CNN/DM)	32.4	42.7	30.7	9.5	13.3	9.0	32.0	39.9	30.1	59.9	-7.57	77.9
CTRLsum BART$_{large}$ (SAMSum)	52.2	53.2	57.1	27.0	27.8	29.5	48.7	49.2	52.2	77.7	-4.79	84.4
+ Beam Search Valid-Augment	52.2	54.0	56.6	27.3	28.7	29.3	48.6	49.7	51.7	77.2	-4.91	87.4
General-Purpose BART-base	50.5	54.8	51.9	25.2	27.4	26.1	47.7	50.7	48.5	75.1	-5.25	78.0
Conditional BART-base	50.0	55.2	50.5	24.8	27.5	25.1	47.1	50.8	47.2	74.0	-5.35	72.9
+ Beam Search Valid-Augment	49.4	55.6	49.1	24.7	28.1	24.7	46.9	51.5	46.4	73.4	-5.45	78.2
General-Purpose BART-large	53.0	57.2	54.5	28.1	30.8	28.7	49.8	53.1	50.5	77.6	-5.11	88.5
Conditional BART-large	52.6	58.1	53.3	27.7	30.9	27.6	49.1	53.1	49.0	76.3	-5.28	85.3
+ Beam Search Valid-Augment	51.9	54.7	54.4	27.4	29.3	28.6	48.1	50.1	49.7	75.9	-5.05	88.1

Table 4: **Results on generation without prompt** (simulating the non-conditional scenario). *F*, *P*, *R* are F1 measure, precision, and recall. *B-Score* and *EACC* are BARTScore (Yuan et al., 2021) and entity-based accuracy.

(Gliwa et al., 2019; Feng et al., 2021). (2) We use the **SimCSE** (Gao et al., 2021) and **BARTScore** (Yuan et al., 2021) as semantic-oriented metrics via counting embedding similarity, which have empirically shown to match the human perception. (3) We calculate the **Entity-based Accuracy** by applying a separate de-noising model on final outputs, which is formulated as the proportion of samples that all personal named entities are correctly generated.

6.5 Quantitative Evaluation Results

6.5.1 Reference Prompt Generation

Firstly, to evaluate the controllable generation under the matched training and test condition, we obtain summaries with the same entity prompt as reference summaries, and this matched condition serves as the benchmarked performance for conditional learning (Liu and Chen, 2021). As shown in Table 2, adding the conditional data augmentation to *'BART-base'* and *'BART-large'* brings improvement on both lexical and semantic metrics, which significantly improves entity-based accuracy. We also evaluate a generic controllable model *'CTRLsum'* (He et al., 2020) that provides state-of-the-art results on the news domain. We observed that *'CTRLsum'* benefits from domain adaptation from news to dialogue (on all fronts) when running experiments on the SAMSum corpus, and we obtained further improvements when adopting the proposed training augmentation.

Source Dialogue Content: Pamela: Have you tried applying for the RGS grant? <u> Zoe: I did last year. <u> Zoe: Wasn't lucky... <u> Xiara: Me neither <u> Pamela: Do you know if I have to be a member? <u> Zoe: It depends on the award <u> Pamela: Ok I'll have a look at their website.
Reference Summary: **Pamela** will look at their website to apply for the RGS grant. Zoe and Xiara applied last year but they weren't lucky. Sometimes you have to be a member to apply. **Generation with General-Purpose (BART-large):** **Zoe** applied for the RGS grant last year, but didn't get lucky. Pamela will look at their website. **Generation with Prompt (BART-large):** **Pamela**, **Zoe** and **Xiara** didn't apply for the RGS grant last year. **Generation with Prompt (BART-large + Valid-Augment):** **Pamela** will have a look at the RGS website to apply for the grant.
Source Dialogue Content: Ivan: hey eric <u> Eric: yeah man <u> Ivan: so youre coming to the wedding <u> Eric: your brother's <u> Ivan: yea <u> Eric: i dont know mannn <u> Ivan: YOU DONT KNOW?? <u> Eric: i just have a lot to do at home, plus i dont know if my parents would let me <u> Ivan: ill take care of your parents <u> Eric: youre telling me you have the guts to talk to them XD <u> Ivan: thats my problem <u> Eric: okay man, if you say so <u> Ivan: yea just be there <u> Eric: alright.
Reference Summary: **Eric** doesn't know if his parents let him go to Ivan's brother's wedding. Ivan will talk to them. **Generation with General-Purpose (BART-large):** **Ivan** is going to **Eric**'s brother's wedding. Eric doesn't know if he can come because he has a lot to do at home. Ivan will talk to his parents. **Generation with General-Purpose (BART-large + Post-Editing):** **Eric** is going to Ivan's brother's wedding. Eric doesn't know if he can come because he has a lot to do at home. Ivan will talk to his parents.

Table 5: Two examples of dialogues in SAMSum, and summaries generated by different models. <u> is the utterance delimiter. Text in blue denotes beginning or prompt entities. Text in red denotes the factual inconsistency.

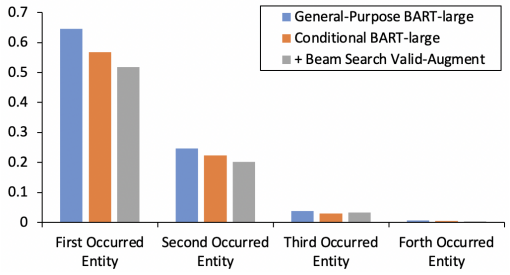


Figure 7: Positional distribution of start entity generated by different models without prompt.

6.5.2 Distinct Prompt Generation

As the single reference summary cannot be readily used for diverse conditional evaluation, to simulate the practical controllable generation scenario, we build a sub-set with distinct prompts (119 of 819 test samples), where the generation by a general-purpose *'BART-large'* and reference summaries start with different personal named entities. As shown in Table 3, all models (*'BART-base'*, *'BART-large'*, and *'CTRLsum'*) with the proposed method achieve higher performance on all fronts, and their entity-based consistency has a relative 12% gain.

6.5.3 Generation without Prompt

While controllable summarizers require a prompt as part of the input, we also obtained summaries without any entity indicator to simulate the general-purpose summarization scenario. As shown in Table 4, models trained in a conditional manner achieve comparable but slightly lower scores. As shown in Figure 7, we speculate that this is because summaries generated by conditionally-trained models present a more balanced entity distribution.

Model	Error Rate
Conditional BART-large	0.37
+ Beam Search Valid-Augment	0.27
+ Inference Post-Editing	0.23

Table 6: Human assessment on entity-based factual consistency of distinct prompt generation.

6.5.4 Results after Inference Post-Editing

For all three generation types shown in Table 2, Table 3, and Table 4), we observed that adopting inference post-editing does not affect the lexical and semantic scores (as it only changes a few tokens), but this post-processing step can improve entity-based consistency by 7% relatively.

Moreover, following previous work (Liu and Chen, 2021), we incorporated dialogue coreference information for the controllable generation, and it is effective to improve the generation quality such as entity accuracy (see results in Appendix).

6.6 Human Assessment on Entity-based Factual Consistency

We further conducted two qualitative evaluations via human assessment. At each time, 30 samples are randomly chosen from the test set and their corresponding summaries from different summarizers. Participants are asked to read the dialogue and summaries, and judge if any personal named entity is misassigned.

For controllable summarization, we evaluate the outputs upon the **distinct prompt generation** (de-

Model	Error Rate
General-Purpose BART-large	0.33
+ Inference Post-Editing	0.26

Table 7: Human assessment on entity-based factual consistency of general-purpose models.

scribed in Section 6.5.2). As shown in Table 6, we observe that the sample-level error rate drops from 0.37 to 0.27 (22% relatively) with the conditional training augmentation, and this is consistent with automatic entity-based accuracy results (see examples in Table 5), and it further drops to 0.23 after the post-editing.

Since the inference post-editing described in Section 5.3 can also be adopted on general-purpose summarizers, we conduct a human assessment on the **non-conditional generation**: we fine-tune a *'BART-large'* which serves as the state-of-the-art baseline on the original SAMSum corpus, and feed its generation to the entity-based de-noising model for post-editing. As shown in Table 7, we observe that the sample-level error rate drops from 0.33 to 0.26 (25% relatively) with the post-editing (see examples in Table 5).

7 Conclusion

In this paper, we focused on reducing incorrect assignments of personal named entities in dialogue summarization. We proposed an entity-based denoising model, and applied it to beam search validation, conditional training augmentation, and inference post-editing (which can be used for non-conditional and conditional summarization). Experimental results demonstrated that our proposed method improves performance in both lexical and semantic evaluation metrics and is beneficial to entity-based factual consistency in both automatic and human evaluations. Future work can be extending it to pronoun tokens and other entity types.

Acknowledgments

This research was supported by funding from the Institute for Infocomm Research (I2R) under A*STAR ARES, Singapore, and by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. We thank Ai Ti Aw for the insightful discussions. We also thank the anonymous reviewers for their precious feedback to help improve and extend this piece of work.

References

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of EMNLP 2020*, pages 4106–4118. Association for Computational Linguistics.

Yulong Chen, Yang Liu, Liang Chen, and Yue Zhang. 2021. DialogSum: A real-life scenario dialogue summarization dataset. In *Findings of ACL-IJCNLP 2021*, pages 5062–5074, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL 2019*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou, Pengfei Liu, Hiroaki Hayashi, Zhengbao Jiang, and Graham Neubig. 2021. Gsum: A general framework for guided neural abstractive summarization. In *Proceedings of NAACL 2021*, pages 4830–4842. Association for Computational Linguistics.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54. Association for Computational Linguistics.

Xiachong Feng, Xiaocheng Feng, Libo Qin, Bing Qin, and Ting Liu. 2021. Language model as an annotator: Exploring DialoGPT for dialogue summarization. In *Proceedings of ACL 2021*, pages 1479–1491. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of EMNLP 2021*, pages 6894–6910.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of EMNLP 2018*, pages 4098–4109, Brussels, Belgium. Association for Computational Linguistics.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New*

Frontiers in Summarization, pages 70–79. Association for Computational Linguistics.

Chih-Wen Goo and Yun-Nung Chen. 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 735–742. IEEE.

Travis Goodwin, Max Savery, and Dina Demner-Fushman. 2020. Towards zero shot conditional summarization with adaptive multi-task fine-tuning. In *Proceedings of EMNLP 2020*, pages 3215–3226.

Prakhar Gupta, Jeffrey P Bigham, Yulia Tsvetkov, and Amy Pavel. 2020. Controlling dialogue generation with semantic exemplars. *arXiv preprint arXiv:2008.09075*.

Junxian He, Wojciech Kryściński, Bryan McCann, Nazneen Rajani, and Caiming Xiong. 2020. Ctrl-sum: Towards generic controllable text summarization. *arXiv preprint arXiv:2012.04281*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Proceedings of NeurIPS 2015*, pages 1693–1701.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *Proceedings of ICML 2017*, pages 1587–1596.

Xinyu Hua and Lu Wang. 2020. PAIR: Planning and iterative refinement in pre-trained transformers for long text generation. In *Proceedings of EMNLP 2020*, pages 781–793. Association for Computational Linguistics.

Hayate Iso, Chao Qiao, and Hang Li. 2020. Fact-based Text Editing. In *Proceedings of ACL 2020*, pages 171–182, Online. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of NAACL 2018*, pages 1875–1885.

Giannis Karamanolakis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan Awadallah. 2021. Self-training with weak supervision. In *Proceedings of NAACL 2021*, pages 845–863, Online. Association for Computational Linguistics.

Chris Kedzie, Kathleen McKeown, and Hal Daume III. 2018. Content selection in deep learning models of summarization. In *Proceedings of EMNLP 2018*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Muhammad Khalifa, Miguel Ballesteros, and Kathleen McKeown. 2021. A bag of tricks for dialogue summarization. In *Proceedings of EMNLP 2021*, pages 8014–8022, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. In *Proceedings of the EMNLP 2019*, pages 540–551, Hong Kong, China. Association for Computational Linguistics.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of EMNLP 2020*, pages 9332–9346.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, page 68–73, New York, NY, USA. Association for Computing Machinery.

Dongyub Lee, Jungwoo Lim, Taesun Whang, Chanhee Lee, Seungwoo Cho, Mingun Park, and Heuiseok Lim. 2021. Capturing speaker incorrectness: Speaker-focused post-correction for abstractive dialogue summarization. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 65–73, Online and in Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of ACL 2020*, pages 7871–7880. Association for Computational Linguistics.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J. Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of ACL 2019*, pages 2190–2196, Florence, Italy. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of EMNLP 2019*, pages 3721–3731, Hong Kong, China. Association for Computational Linguistics.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. In *Proceedings of ACL-IJCNLP 2021*, pages 1065–1072.

Zhengyuan Liu and Nancy Chen. 2021. Controllable neural dialogue summarization with personal named entity planning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 92–106, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengyuan Liu, Angela Ng, Sheldon Lee, Ai Ti Aw, and Nancy F Chen. 2019. Topic-aware pointer-generator networks for summarizing spoken conversations. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 814–821. IEEE.

Zhengyuan Liu, Ke Shi, and Nancy Chen. 2021b. Coreference-aware dialogue summarization. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 509–519, Singapore and Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *The International Conference on Learning Representations (ICLR 2019)*.

Fuli Luo, Peng Li, Jie Zhou, Pengcheng Yang, Baobao Chang, Xu Sun, and Zhifang Sui. 2019. A dual reinforcement learning framework for unsupervised text style transfer. In *IJCAI*.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of AAAI 2017*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS 2019*, pages 8026–8037.

Maja Popović and Mihael Arčan. 2016. PE2rr corpus: Manual error annotation of automatically pre-annotated MT post-edits. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 27–32, Portorož, Slovenia. ELRA.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of EMNLP 2015*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Proceedings of NeurIPS 2017*, pages 6830–6841.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeurIPS 2017*, pages 5998–6008.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34.

Yuhao Zhang, Derek Merck, Emily Tsai, Christopher D. Manning, and Curtis Langlotz. 2020. Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of ACL 2020*, pages 5108–5120. Association for Computational Linguistics.

Zheng Zhao, Shay B. Cohen, and Bonnie Webber. 2020. Reducing quantity hallucinations in abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2237–2249. Association for Computational Linguistics.

Ming Zhong, Da Yin, Tao Yu, Ahmad Zaidi, Mutethia Mutuma, Rahul Jha, Ahmed Hassan Awadallah, Asli Celikyilmaz, Yang Liu, Xipeng Qiu, and Dragomir Radev. 2021. QMSum: A new benchmark for query-based multi-domain meeting summarization. In *Proceedings of the 2021 Conference of the NAACL*, pages 5905–5921, Online. Association for Computational Linguistics.

A Appendix

	ROUGE-1			ROUGE-2			ROUGE-L			SimCSE	B-Score	EACC
	F	P	R	F	P	R	F	P	R			
Conditional BART-large	53.8	61.6	52.6	30.2	35.1	29.4	52.9	59.1	51.5	78.4	-5.23	84.7
+ Coreference Information	54.3	57.9	56.8	30.2	32.6	31.5	52.6	55.3	54.3	79.5	-4.72	85.8
+ Beam Search Valid-Augment	54.1	56.0	58.4	30.5	32.1	32.4	52.3	53.5	55.3	79.2	-4.69	86.8
+ Inference Post-Editing	54.2	56.6	57.7	30.3	32.1	32.3	52.5	54.2	55.1	80.2	-4.59	98.2

Table 8: **Additional experimental results on reference prompt generation** (matching training and test condition). We incorporated dialogue coreference information following previous work (Liu and Chen, 2021). *F*, *P*, *R* are F1 measure, precision, and recall. *B-Score* and *EACC* denotes BARTScore and entity-based accuracy.

	ROUGE-1			ROUGE-2			ROUGE-L			SimCSE	B-Score	EACC
	F	P	R	F	P	R	F	P	R			
Conditional BART-large	55.1	62.3	53.3	29.5	33.5	29.3	53.2	59.2	52.3	78.2	-5.04	68.3
+ Coreference Information	54.9	58.5	57.2	29.9	32.1	31.0	51.5	53.6	53.1	79.2	-4.90	76.1
+ Beam Search Valid-Augment	55.5	55.8	59.5	29.1	29.5	31.1	51.7	51.8	54.5	78.5	-4.58	77.7
+ Inference Post-Editing	55.1	55.3	59.1	27.6	27.7	29.8	50.0	49.4	52.5	78.5	-4.57	97.9

Table 9: **Additional experimental results on distinct prompt generation** (simulating a practical use case). We incorporated dialogue coreference information following previous work (Liu and Chen, 2021). *F*, *P*, *R* are F1 measure, precision, and recall. *B-Score* and *EACC* denotes BARTScore, and entity-based accuracy.

	ROUGE-1			ROUGE-2			ROUGE-L			SimCSE	B-Score	EACC
	F	P	R	F	P	R	F	P	R			
Conditional BART-large	52.6	58.1	53.3	27.7	30.9	27.6	49.1	53.1	49.0	76.3	-5.28	85.3
+ Coreference Information	52.2	51.7	58.4	27.1	27.0	30.5	47.9	47.8	52.8	77.3	-4.75	87.2
+ Beam Search Valid-Augment	52.0	51.3	59.1	26.9	26.8	30.6	47.6	46.6	52.8	77.1	-4.69	85.3
+ Inference Post-Editing	51.9	51.2	59.0	26.9	26.9	30.5	47.6	46.7	52.7	76.9	-4.38	98.1

Table 10: **Additional experimental results on generation without prompt** (simulating the non-conditional scenario). We incorporated dialogue coreference information following previous work (Liu and Chen, 2021). *F*, *P*, *R* are F1 measure, precision, and recall. *B-Score* and *EACC* are BARTScore, and entity-based accuracy.

iEval: Interactive Evaluation Framework for Open-Domain Empathetic Chatbots

Ekaterina Svikhnushina, Anastasiia Filippova and **Pearl Pu**
School of Computer and Communication Sciences
EPFL, Lausanne, Switzerland
{ekaterina.svikhnushina,anastasiia.filippova,pearl.pu}@epfl.ch

Abstract

Building an empathetic chatbot is an important objective in dialog generation research, with evaluation being one of the most challenging parts. By empathy, we mean the ability to understand and relate to the speakers' emotions, and respond to them appropriately. Human evaluation has been considered as the current standard for measuring the performance of open-domain empathetic chatbots. However, existing evaluation procedures suffer from a number of limitations we try to address in our current work. In this paper, we describe iEval, a novel interactive evaluation framework where the person chatting with the bots also rates them on different conversational aspects, as well as ranking them, resulting in greater consistency of the scores. We use iEval to benchmark several state-of-the-art empathetic chatbots, allowing us to discover some intricate details in their performance in different emotional contexts. Based on these results, we present key implications for further improvement of such chatbots. To facilitate other researchers using the iEval framework, we will release our dataset consisting of collected chat logs and human scores.[1]

1 Introduction

Development of open-domain chatbots endowed with social and emotional intelligence is a crucial task in natural language research (Rashkin et al., 2019). Empathetic chatbots are expected to engage in a conversation with the users and demonstrate understanding and appropriate handling of users' feelings. While many strategies for generating empathetic responses have been described, there is still little consensus on their evaluation. For dialog generation, automatic metrics do not show consistency in correlations with human judgement (Liu et al., 2016; Tao et al., 2018), leading to their limited adoption. Therefore, most of existing works rely on human evaluation. It may happen in either *static* or *interactive* setting (Adiwardana et al., 2020). In the former case, a human judge rates chatbot's responses, generated from a fixed set of

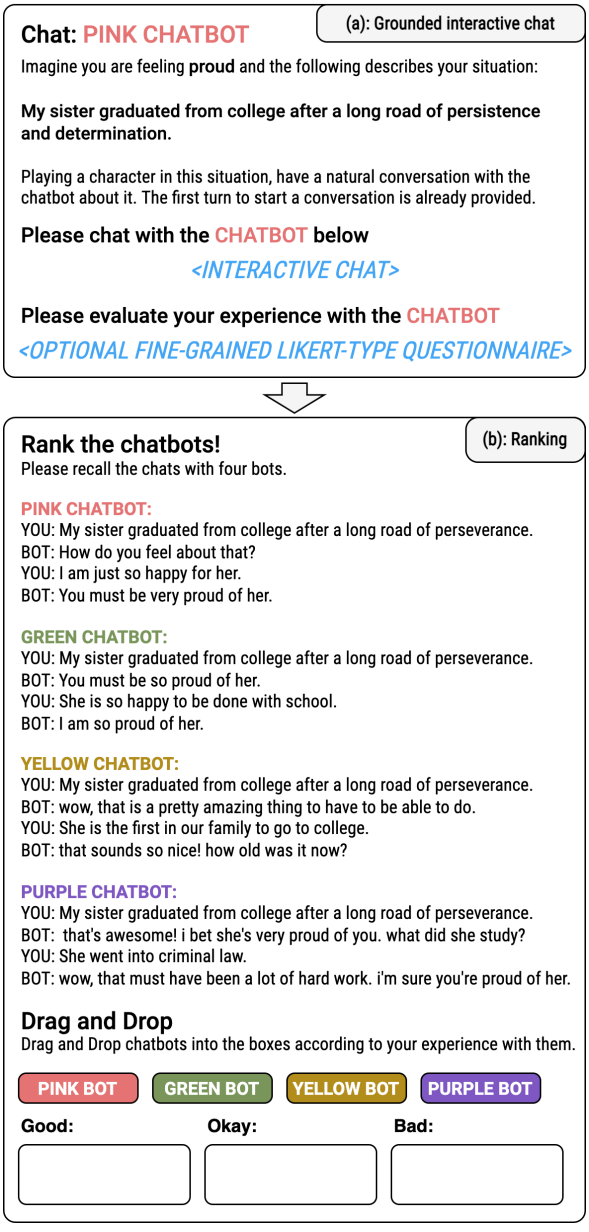


Figure 1: iEval framework.

[1]Our annotated dataset is publicly accessible at https://github.com/Sea94/ieval.

Proceedings of the SIGdial 2022 Conference, pages 419–431
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

contexts. In the latter case, dialogs for evaluation are collected as humans' multi-turn chats with the model.

Recently, two comprehensive approaches based on interactive multi-turn human evaluation were proposed. Adiwardana et al. (2020) described a metric called Sensibleness and Specificity Average, which measures these two aspects of chatbot's responses. Human judges give Likert-type scores to each chatbot's turn in a dialog, which are further averaged to obtain a final score. As Likert-type scores may exhibit differing bias and variance per annotator, associated with the lack of sensitivity, Li et al. (2019) suggested an alternative evaluation strategy based on pairwise comparisons. According to their method, human judges indicate their preference of one chatbot over another by comparing two dialog logs with these chatbots. This procedure is more robust, but become very costly when the number of compared models goes up.

Both of these approaches differentiate humans who interact with the models and humans who judge them. They probably opt for this design choice due to such considerations as workers' fatigue. However, according to findings in cognitive psychology, our emotional experiences are highly subjective. Barrett et al. (2007) points out that only the experiencers can reveal the full complexity of emotions that they feel. For example, if a client complains about a hotel room being too cold, a third-party observer might underestimate the gravity of the issue, especially if he enjoys indoor coolness. This fact argues for the necessity of a new evaluation approach of chatbots, which would ensure that both emotional interaction and evaluation of a chatbot are accomplished by the same human actor. To help these humans share their emotional experiences, asking them to role-play a relatable scenario is a frequently used procedure in social sciences (Walther et al., 2005; Hancock et al., 2007).

In this work, we introduce iEval, an interactive evaluation framework for open-domain empathetic chatbots, which mitigates the issue of separating an experiencer and an evaluator. To combine the benefits of Likert scales, allowing to evaluate many chatbots in a single stretch of time, and pairwise comparisons, offering greater reliability and cross-experiment robustness, we propose a novel ranking-based approach. According to iEval, a human first converses with all chatbots, having all chats grounded in an emotional scenario (Figure 1 (a)). Then, the same human ranks the models by dragging-and-dropping them into corresponding categories (Figure 1 (b)). Our experiments demonstrate that iEval can reveal subtle but significant differences in chatbots' performance across emotional contexts.

Overall, our contributions include the following. 1) We describe a new evaluation framework to measure chatbots' abilities to respond appropriately in sensitive contexts. 2) We demonstrate a rigorous procedure for preparing grounding scenarios for the given evaluation task. 3) We benchmark several state-of-the-art empathetic chatbots, which have never been compared before. 4) Based on the analysis of the benchmark results, we discuss implications for the future development of empathetic chatbots. 5) Finally, we release the data from our experiments to facilitate future research endeavors.

2 Related Work

Most works focusing on the development of empathetic chatbots couple automatic evaluation with human judgement. Automatic metrics usually include perplexity, approximating the model's language modeling ability (Roller et al., 2021; Xie and Pu, 2021; Li et al., 2020), and may incorporate other scores, depending on the specific focus of the work. Some frequently used examples are BLEU score (Lin et al., 2019; Majumder et al., 2020), diversity metrics (Xie and Pu, 2021; Li et al., 2020), and F-1 score or accuracy of emotion detection (Lin et al., 2019; Xie and Pu, 2021; Li et al., 2020).

Since the appropriateness of automatic metrics for open-domain dialog is still ambiguous, all works de facto rely on human judgement. Most commonly, researchers employ single-turn static evaluation, where a fixed emotionally-colored context is shown to a judge along with the responses generated by different chatbots. The judges are asked to rate how empathetically appropriate the responses are, and the assessment may come either as Likert-type scores (Hu et al., 2018; Lin et al., 2019; Majumder et al., 2020; Li et al., 2020) or ranking (Xie and Pu, 2021). Although this approach is widespread due to the ease of implementation, it fails to capture issues emerging in multi-turn chats, such as repetitiveness or deterioration of semantic coherence in long-range contexts (See et al., 2019).

Few works that focus on integrating empathetic abilities into chatbots started adopting interactive evaluations. Roller et al. (2021) employed ACUTE-

Eval (Li et al., 2019) framework based on pairwise comparisons to assess engaginess and humanness of their models. Ghandeharioun et al. (2019) defined their own evaluation protocol to collect Likert-type scores for a series of dimensions measuring chatbot's performance. However, in both of these studies, the evaluated data points were open-ended chats that began with a generic greeting. Based on the provided examples of conversations, these exchanges generally developed as light small-talk, maintaining neutral or positive sentiments. Therefore, it remains unclear how well the collected scores reflect empathetic abilities of the chatbots, which should ideally succeed over a range of emotions. Our framework addresses this limitation by grounding the chats in diverse emotional scenarios.

3 Method: iEval

To compare empathetic abilities of several chatbots, iEval suggests that at first a human makes an emotionally-grounded conversion with each bot in a randomized order. If necessary, fine-grained Likert-type assessments of specific chatbot's performance aspects may be collected after each conversation. As the next step, the same human is asked to rank the chatbots according to her experience with them. An example of this flow is given in Figure 1. Finally, appropriate statistical instruments should be applied to compare the chatbots.

3.1 Emotionally-grounded Chats

To make sure that humans experience the full extent of chatbots' empathetic abilities, we condition each conversation with a short emotional scenario, instructing the humans to imagine themselves feeling a particular emotion in a given situation. They are further asked to role-play a character in this scenario and chat about it with the models. The first dialog turn is provided to the humans to facilitate the process of their getting into the assigned role.

Careful conditioning of the experiment is essential to ensure that it adequately represents chatbots' abilities in a vast range of topics and emotions. We noticed that some dialogs from the EmpatheticDialogues dataset (Rashkin et al., 2019), a popular dataset for building empathetic models, form large clusters in terms of the similarity of discussed situations (see Appendix A). It may lead to models' shifted performance on specific topics. Therefore, one should control for topical diversity when defining conditioning scenarios for iEval.

Besides, previous results pointed out that the same model may receive different appraisals depending on the emotional polarity of the chats (Majumder et al., 2020). This may be linked to the existing difference between humans' empathetic responding in positive and negative scenarios (Aue et al., 2021), and hence difference in expectations. Thus, we argue for the importance of balancing and studying the role of emotional polarity within iEval.

Finally, ensuring sufficient interaction experience with the models is necessary before asking humans for their judgements. Previous works required between 3 and 14 chatbot's turns per dialog. We find 3 turns to be enough, given that the dialog starts with a specific input.

3.2 Ranking

The concluding step of iEval requests a human to recall the conversations with the chatbots and rank them by assigning the bots into three categories: *Bad*, *Okay*, and *Good*. Several chatbots can be assigned to the same category, indicating equal rank. This approach allows moving away from inter-annotator variability associated with Likert scales (Li et al., 2019; Kulikov et al., 2019), while preserving the benefits of relative comparisons. To obtain the final standing of the chatbots, we propose converting the resulting rank into an ordinal rating (*Bad* $\rightarrow$ 1, *Good* $\rightarrow$ 3) and running non-parametric ANOVA to compare the mean ratings.

3.3 Annotation Quality

According to iEval framework, one human should chat with and evaluate several models. As human's short-term mental storage capacity is limited to several informational chunks, we recommend keeping the number of evaluated models between 3 and 7, giving preference to lower values (Cowan, 2001).

To meet the requirements of randomized controlled experiments, it is also advisable to allow each human to complete only one evaluation task to eliminate anchoring effects. For the same reason, the order in which humans interact with the chatbots should be randomized and counterbalanced across tasks. To distinguish different models without revealing their names to the humans, we suggest color-coding them to avoid any fixation effects which could be caused by aliases that reflect order.

Finally, we use crowdsourcing for our experiment. To decrease the probability of encountering

fraudulent or inattentive workers, human intelligent task design and configuration should follow the quality control recommendations of the platform in combination with other attention checks.

4 Experiment

To demonstrate how iEval works in practice, we apply the framework to benchmark several state-of-the-art empathetic chatbots, which have never been compared against each other in an interactive setting. The details and analysis are outlined below.

4.1 Measures

We use the final ranking of the chatbots, converted into ordinal ratings, as our main metric. To better understand which factors play a principal role in defining overall ranking, we also ask human workers for fine-grained Likert-type scores to a number of chatbots' qualities on a 1-5 scale. These questions were derived as a combination of the established key qualities for conversational chatbots (Svikhnushina and Pu, 2021) and other critical aspects related to their language modeling abilities (See et al., 2019). We measured chatbots' perceived politeness, empathy, likability, repetitiveness, and whether their responses make sense.

4.2 Models

We benchmarked four models, as this corresponds to an average number of informational chunks that humans can store in short-term memory (Cowan, 2001). We chose between the top-performing chatbots available at the moment of preparing our experiment in Q4 2021. We selected the models, which use distinct approaches for generating empathetic responses. Only one of them participated in an interactive evaluation previously, but it was not targeted at its empathetic skills. The four models with assigned color-codes are as follows.

Blender is a large model employing a standard Seq2Seq Transformer architecture with ≈90M parameters (Roller et al., 2021). Blender was pre-trained on ≈1.5B comments from Reddit discussions and fine-tuned on EmpatheticDialogues dataset (Rashkin et al., 2019).

MIME is a relatively small model with ≈18M parameters also based on Seq2Seq Transformer with additional stochastic emotion grouping and mimicry mechanism (Majumder et al., 2020). Without pretraining, MIME was directly initialized with GloVe embeddings (Pennington et al., 2014) and fine-tuned on EmpatheticDialogues.

MEED is a middle-size Seq2Seq Transformer-based model with ≈40M parameters, which incorporates extra controllability of response generation achieved through modeling fine-grained empathetic intents. The model was pre-trained on ≈1M dialogs from OpenSubtitles (Lison and Tiedemann, 2016) and fine-tuned on EmpatheticDialogues.

Plain is a basic Seq2Seq Transformer-based model with ≈40M parameters, which followed the same training pipeline as MEED. Plain serves as a baseline in our experiment.

All models were adapted to operate in an interactive setting so that for generating each next response, all previous dialog history was passed to the models as input.

4.3 Grounding Scenarios

As EmpatheticDialogues (Rashkin et al., 2019) is the mainly used benchmarking dataset for empathetic chatbots, we employed its test set to create grounding scenarios. This dataset contains 24,850 dialogs associated with emotional contexts (out of which 2,547 dialogs comprise the test set). To create the dataset, (Rashkin et al., 2019) connected two types of crowdworkers, speakers and listeners, to have conversations with each other. Speakers first had to select one of the 32 emotional labels (e.g., *sad*, *joyful*, *proud*) and describe a situation when they felt that way. Then they proceeded to have a conversation with the listeners using the outlined situations as guiding prompts. We utilized these attributes (32 emotional labels and prompts describing the speakers' situations) to describe our grounding scenarios and kept the first turn from each selected dialog as a starting turn for the worker in our evaluation task.

To ensure comprehensibility of the task for crowdworkers, this selection of grounding prompts and opening utterances was organized very carefully. Firstly, we selected dialogs where the length of the associated prompt falls between the first and third quantiles in terms of the number of tokens to ensure it provides sufficient details about the speaker's situation. Secondly, we computed Vader sentiment scores (Hutto and Gilbert, 2014) of the first utterance in each dialog and only kept those that had a clear emotional coloring. These steps produced 527 data points, which we finally proofread and annotated with emotional polarity labels (negative or positive). Note that we used the

original 32 emotional labels to show them to crowdworkers to ground their interaction with the chatbots, while the polarity labels were needed for the analysis part. We further narrowed the set of 527 data points down to 480 prompts with utterances to meet our experimental design requirements (§4.4). The discarded data points were chosen manually in order to diversify the topics in the main set. The distribution of emotional labels in the resulting evaluation set is shown in Figure 8 in Appendix B. Some examples of grounding scenarios (emotional labels and prompts) are provided in Figures 4, 5, and 6.

4.4 Experiment Design

We aimed at evaluating the performance of the participating chatbots, while also contrasting their abilities in negative and positive emotional contexts. To maintain a manageable number of human intelligence tasks (HIT), we decided to ask each crowdworker to interact with all chatbots in both conditions. Therefore, our experiment was a 2×4 within-subject factorial design. By designing our study as a factorial experiment, we were able to examine both main effects and interactions among chatbots and emotional contexts. We used G*Power software to estimate the required sample size to achieve "medium" effect size (Faul et al., 2007). As the recommended sample size was about 200, we ran 240 experimental tasks to achieve a full counterbalance of the order of chatbots and emotional contexts across subjects. We analyzed ranking of the chatbots using the nonparametric Aligned Rank Transform (ART) procedure (Wobbrock et al., 2011). Quartile-quartile plots of the fitted residuals of our the model showed that they were normally distributed, indicating the appropriateness of this model for our analysis.

4.5 Running the Experiment

We ran our experiment on Amazon Mturk, requiring one US-based worker per each of the 240 HITs. Our workers spent on average 20.6 minutes to complete a HIT and their reward was $2.5 per HIT, which agrees with the US minimum wage standards. Following Mturk recommendations,[2] we required the workers to have 98% approval rate and 10,000 approved HITs. We further rejected the workers whose average HIT completion time,

[2] https://blog.mturk.com/qualifications-and-worker-task-quality-best-practices-886f1f4e03fc

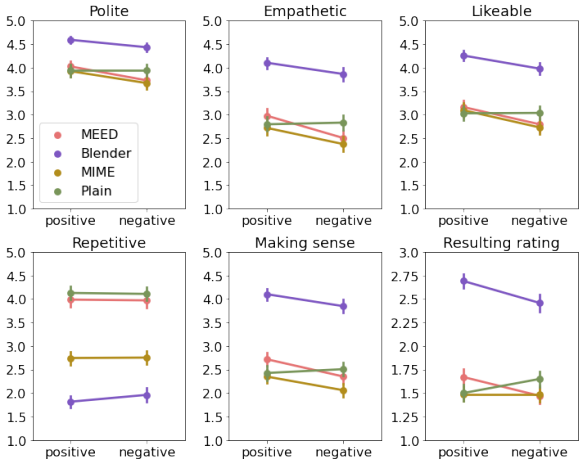


Figure 2: Benchmarking results of the four chatbots.

length of chat responses, or number of contradictory responses to reverse-scaled questions in the Likert-type questionnaire stood out as outliers.

5 Analysis of Results

Below, we describe the eventual ranking of the models and consider the aspects that likely explain the observed results.

5.1 Benchmarking of Empathetic Chatbots

We used the nonparametric ART procedure to analyze ranking of the chatbots. As described above (§3.2), for this analysis we converted the resulting rank into an ordinal rating for more straightforward interpretation (the higher, the better). Results show a main effect of chatbot ($F_{3,1673} = 257.92$, $p < 0.001$) and of emotional context ($F_{1,1673} = 43.17$, $p < 0.001$) on the rating, and of their interaction ($F_{1,1673} = 9.80$, $p < 0.001$) as illustrated in the lower right subplot of Figure 2. Interaction results revealed several interesting relationships. Blender is consistently rated significantly higher than the other three chatbots, and it also performs significantly better in positive contexts than in negative ($p < 0.01$). MIME is rated the lowest, while for MEED and Plain a shift in the ratings emerges depending on emotional context. MEED significantly outperforms Plain in positive contexts ($p < 0.05$) while the diametrically opposite result manifests for negative contexts ($p < 0.05$).

5.2 Aspects Explaining the Ranking

We fitted an ordinal regression model to identify which of the factors measured by our Likert-type questionnaire correlate strongest with the assigned

ratings (McFadden's pseudo-$R^2 = 0.37$). The statistical model was chosen due to the ordinal nature of the dependent variable. All evaluated qualities exhibit significant influence on chatbots' ratings. Making sense ($\beta = 1.01, p < 0.001$), empathy ($\beta = 0.35, p < 0.001$), and repetitiveness ($\beta = -0.32, p < 0.001$) are the strongest predicting factors, followed by politeness ($\beta = 0.21, p < 0.01$) and likability ($\beta = 0.18, p < 0.05$) (Figure 3).

The leading factor suggests that the language modeling abilities of the chatbots define their ranking at large. This is understandable as language fluency is essential for the comprehensive handling of emotions in chat. Blender, being a massive model pre-trained on a dataset that is 1000-time larger than the one for MEED or Plain, is capable of generating considerably longer fluent responses than any other chatbot (Figure 9 in Appendix C), which ensures its highest rank. Meanwhile, MIME is the smallest model, which did not undergo any pre-training. It responds reasonably well to the first speaker's utterance, but as context gets longer its ability to produce semantically coherent responses quickly deteriorates (Figure 4), causing its poor performance regardless of the emotional context.

For chatbots with better language modeling skills, the next most decisive factors bring about curious shift in chatbots' rankings, depending on the emotional polarity of conversations. We analyze these phenomena in the next section, focusing on empathy and repetitiveness (diversity) revealed in chatbots' responses. While considering empathy of the models, we pay special attention to the role of questions that they ask as it has been established to be the most prominent intent of human empathetic listeners (Welivita and Pu, 2020). Moreover,

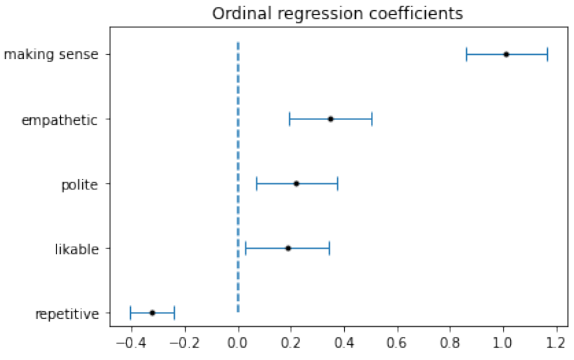


Figure 3: Results of ordinal regression on rank. 95% confidence intervals are approximated as two standard errors. All coefficients are significant ($p < 0.05$).

MIME (context: Positive, rank: Okay)

Emotion: joyful

Prompt: was delighted the other day when I got to see a friend that I had not seen in 20 years.

***Speaker**: I was delighted the other day when I got to see a friend that I had not seen in 20 years.*

***MIME**: that is awesome. i hope you guys get some great news.*

***Speaker**: Yes, she seemed to be doing well. It was so nice to see her again.*

***MIME**: that is a good thing. i bet you were happy.*

***Speaker**: I was, yes.*

***MIME**: what did he do?*

MIME (context: Negative, rank: Bad)

Emotion: afraid

Prompt: It is so dark in my house. I keep hearing creepy noises.

***Speaker**: I keep hearing strange noises in my house. My husband is out of town and I'm all alone. There has been reports of break ins on my street.*

***MIME**: oh my goodness, that is not good, how long have you had to do?*

***Speaker**: I've been hearing strange noises for about an hour now. I'm scared.*

***MIME**: that is a good thing to be in the world. i hope it is not too late to get the right now.*

***Speaker**: What do you mean?*

***MIME**: i am sure that you are a good feeling.*

Figure 4: Examples of chat logs with MIME.

question mark appears in the top-15 most frequent tokens in the responses of all models studied in our experiment (Table 3 in Appendix C).

5.3 Analysis of Interaction Effects

5.3.1 Intricacies between MEED and Plain

Both MEED and Plain have moderate language modeling abilities compared to the other two counterparts. To reason about why these models' rankings swapped depending on the emotional polarity, we make two noteworthy observations. First, even though the gap in scores is not huge, Plain is rated significantly more repetitive than MEED (Figure 2). Second, as it can be seen from Table 1, both chatbots actively ask questions in their responses, but MEED asks significantly more questions than Plain in negative contexts (independent t-test $p < 0.01$).

It is expected that MEED establishes greater diversity and poses questions as it models multiple listeners' empathetic intents, where questioning makes up the largest class. However, as we observed during qualitative inspection of the dialog logs, MEED falls into a typical trap of neural-

	Dialog level			Turn-level: Positive			Turn-level: Negative		
	Overall	Positive	Negative	Bot's turn 1	Bot's turn 2	Bot's turn 3	Bot's turn 1	Bot's turn 2	Bot's turn 3
MEED	1.83 (SD=0.96)	1.78 (SD=0.98)	1.89 (SD=0.94)	0.91 (SD=0.29)	0.56 (SD=0.51)	0.3 (SD=0.46)	0.93 (SD=0.26)	0.63 (SD=0.48)	0.33 (SD=0.47)
Blender	1.12 (SD=0.94)	1.25 (SD=0.92)	0.99 (SD=0.93)	0.73 (SD=0.55)	0.21 (SD=0.43)	0.32 (SD=0.52)	0.73 (SD=0.65)	0.11 (SD=0.35)	0.15 (SD=0.37)
MIME	1.26 (SD=0.91)	1.26 (SD=0.94)	1.25 (SD=0.88)	0.34 (SD=0.48)	0.46 (SD=0.54)	0.47 (SD=0.55)	0.35 (SD=0.48)	0.42 (SD=0.51)	0.48 (SD=0.53)
Plain	1.72 (SD=1.08)	1.78 (SD=1.07)	1.68 (SD=1.09)	0.83 (SD=0.38)	0.52 (SD=0.50)	0.43 (SD=0.50)	0.82 (SD=0.39)	0.53 (SD=0.50)	0.34 (SD=0.47)

Table 1: Average number of questions with standard deviation (in the parentheses) asked by different chatbots.

MEED (context: Positive, rank: Okay)	**Plain** (context: Positive, rank: Bad)
Emotion: anticipating	**Emotion**: anticipating
Prompt: I just ordered a new cookery book and I can't wait for it to arrive. I love to cook and will really enjoy trying out the recipes.	**Prompt**: I just ordered a new cookery book and I can't wait for it to arrive. I love to cook and will really enjoy trying out the recipes.
Speaker: I just ordered a new cookery book and am eagerly awaiting its arrival. I want to delve in and try lots of new recipes!	***Speaker:*** I just ordered a new cookery book and am eagerly awaiting its arrival. I want to delve in and try lots of new recipes!
MEED: What kind of dish is it?	***Plain:*** That sounds like a good idea. What kind of recipes do you like?
Speaker: Well I want to find something with beef first.	***Speaker:*** I love deserts the most. I want to try a new chocolate cake recipe.
MEED: That sounds like a good recipe.	***Plain:*** That sounds like a good idea. What kind of recipe is it?
Speaker: Beef always is. I think I will try to find a good casserole to try.	***Speaker:*** I don't know yet. i need to look but definitely chocolate.
MEED: That sounds delicious! I'm sure you will find a good casserole.	***Plain:*** That sounds like a good idea. I'm sure you'll like it.
MEED (context: Negative, rank: Bad)	**Plain** (context: Negative, rank: Okay)
Emotion: terrified	**Emotion**: terrified
Prompt: I am so scared to live in my neighborhood. There is people that come around shooting their guns.	**Prompt**: I am so scared to live in my neighborhood. There is people that come around shooting their guns.
Speaker: I am so scared to live in my neighborhood. There is people that come around shooting their guns.	***Speaker:*** I am so scared to live in my neighborhood. There is people that come around shooting their guns.
MEED: What kind of guns?	***Plain:*** What kind of guns do you have?
Speaker: I don't know the kind of gun, but it is upsetting, and they are shooting close to my house.	***Speaker:*** I don't have any guns.
MEED: What kind of gun do you think they are shooting close to your house?	***Plain:*** I don't have any guns either. I don't like guns either.
Speaker: I don't know. A hand gun? What does that matter?	***Speaker:*** So what should I do to feel safer in my neighborhood?
MEED: Do you think they are shooting close to your house?	***Plain:*** I don't have any guns either. I don't like guns either.

Figure 5: Examples of chat logs with MEED and Plain.

based chatbots by learning the most safe and general type of response, in this case question, e.g., *"What type of... was it?"* While asking such general information-seeking questions is an acceptable strategy for positive contexts, other questioning behaviors were shown to be more effective in delivering meaningful emotional regulation in negative scenarios (Svikhnushina et al., 2022).

We further combined these observations with the fact that correlation between these chatbots' repetitiveness scores and overall ratings is slightly lower in negative scenarios (Pearson's $r = -0.42$ ($p < 0.001$)) than in positive (Pearson's $r = -0.51$ ($p < 0.001$)). It suggests one plausible explanation to the observed phenomenon. In positive contexts, human speakers value chatbots' diversity and active engagement demonstrated via questioning, and are more forgiving even if the chatbot's response is slightly misaligned with the context. In negative scenarios, speakers feel much more vulnerable and

expect greater attention. Consequently, they prefer a generic, but safe response over the one which is somewhat unrelated or diverting attention from the speaker's emotional state. Figure 5 provides examples illustrating these observations.

5.3.2 Decline of Blender in Negative Contexts

To study the possible reasons of Blender's lower performance in negative contexts, we started with qualitative inspection of dialog logs. While Blender asks fewer questions than MEED or Plain, they still appear frequently in its responses (Table 1) and the same issue of asking overly general questions, failing to address speaker's emotional needs in negative contexts, preserves also for this chatbot.

More interestingly, we observed that Blender is the only chatbot in our experiment that persistently shares its own experiences and views with the speakers. To get a better idea of this behavior, we randomly sampled 50 chat logs (25 per emotional polarity) for each chatbot and annotated how many of their responses were oriented towards the speaker (other), reflecting the chatbot's own experience (self), or both. Blender is the only chatbot, for whom the amount of responses oriented towards self and both largely exceeds zero in the later dialog turns, almost reaching the proportion of purely other-oriented responses (Table 2 in Appendix C).

Self-disclosure is frequent in peer support discussion forums (Barak and Gluck-Ofri, 2007). This likely explains Blender's tendency to share own perspective as it was pre-trained on Reddit conversations, where peer support is actively practiced. However, human attitude to chatbot's sharing about self is unclear, especially in negative scenarios. Even in human-human interaction, positive disclosure is appreciated more than negative (Caltabiano and Smithson, 1983). Moreover, in counselling practice, therapist self-disclosure is usually portrayed as a mistake (Henretty and Levitt, 2010). We could not find studies about users' preferences for the degree of chatbot's self-oriented responses, but some previous findings about embodied computer agents reveal that their empathetic other-oriented emotions lead to more positive ratings of the agent (Brave et al., 2005). We, therefore, hypothesize that pulling attention to self too quickly in negative conversations might have resulted in Blender's poorer performance in this emotional polarity, which is demonstrated with an example in Figure 6.

Blender (context: Positive, rank: Good)

Emotion: nostalgic

Prompt: I get sentimental during the holidays. I think back to the ones i had as a kid.

***Speaker**: I love the holidays, that make think back to being a kid.*

***Blender**: i agree, it's a great time to spend with family and friends. What do you do for fun?*

***Speaker**: We bake cookies, listen to christmas songs. What about you?*

***Blender**: i like to play video games and spend time with my family. what kind of cookies do you bake?*

***Speaker**: pecan, chocolate chip, yum!*

***Blender**: what is your favorite video game to play? i've never played video games, but i love video games.*

Blender (context: Negative, rank: Bad)

Emotion: devastated

Prompt: I was so upset when I failed my math test, I was only 1 percent off!

***Speaker**: I was one percent off from passing my math test, I was devastated.*

***Blender**: i'm sorry to hear that. were you able to get a better grade on the test?*

***Speaker**: No, I am just upset.*

***Blender**: i think i am going to go back to school. i am not sure what i will do.*

***Speaker**: Make sure to study.*

***Blender**: i am sure you will do great. i hope you get a good grade on your test. good luck!*

Figure 6: Examples of chat logs with Blender.

6 Discussion

6.1 Implications for Chatbot Development

Most of the chatbots in our experiment were trained to model short-context conversations and did not support the interactive chat mode by default, which also applies to other dialog models, e.g. (Hu et al., 2018; Lin et al., 2019). Nevertheless, being able to maintain continuous engaging conversation is an ultimate goal for empathetic chatbots. Thus, more attention should be paid to adapting training procedures and architectures to track longer-term dialog history and evolution of speaker's emotions.

Our findings demonstrate that users' emotional needs differ in positive and negative scenarios, and that they do not necessarily expect a strong emotional reaction to their inputs. Raising a question may be an appropriate response. According to our results, chatbots should dwell longer on speakers' negative situations, employing meaningful questioning strategies, which can possibly be achieved by modeling fine-grained empathetic questioning

intents (Svikhnushina et al., 2022). In addition, more research on the amount of chatbots' self-disclosure would further help tailor chatbots' responses to users' expectations.

6.2 Next Steps

While human evaluation is the current standard to assess chatbots' performance, developing an automated metric to approximate human judgement is an important milestone that would considerably facilitate the developmental cycle. Some attempts towards this goal have been made (Yeh et al., 2021), but very few of these metrics try to capture empathetic abilities of chatbots. Our analysis suggests that all dimensions evaluated in our Likert-type questionnaire constitute significant predictors of the overall human satisfaction (§5.2). Therefore, to develop a stronger automatic proxy for human evaluation, we consider creating rationale heuristics approximating those dimensions and identifying a meaningful way to combine them into a single score. The dataset of collected chat logs and human scores from our experiment should streamline the construction and calibration of such a metric.

7 Limitations

In our work, we applied iEval framework to benchmark four empathetic agents. We did not compare them against human-human interaction, as synchronizing two crowdworkers for conducting several chats between each other entails more logistical difficulties. More importantly, we were mainly interested in measuring how existing chatbots address users' emotional needs, rather than checking if they are indistinguishable from human interlocutors.

Our results show that bigger models rank higher in the evaluation task. It raises the subsequent question about to what extent the proposed framework measures differences in models' empathetic abilities compared to their underlying language model performances. We believe that iEval is an effective framework for evaluating chatbots' empathy as it succeeded in registering intricate differences in the performances of MEED and Plain, two models of comparable sizes and pre-training pipelines, as well as distinguishing the performance of Blender in emotional contexts of different polarity. To further disentangle the role of language modeling and empathetic abilities, one can consider running the iEval evaluation experiment to compare equal-size models with and without fine-tuning for empathetic response generation (e.g., Blender, which was only pre-trained on Reddit, and Blender, which was further fine-tuned on the EmpatheticDialogues dataset). However, this was not the main objective of our study and we leave it for future work.

Finally, we propose to use ranking as a way of expressing the appraisals of the chatbots, as it affords advantages of both Likert scales and pairwise comparisons. Ranking may be less robust for comparing results across experiments with mismatched sets of chatbots. Applying rank aggregation techniques can be useful to tackle such cases (Sculley, 2007).

8 Conclusion

Our paper introduced iEval, a novel evaluation framework for open-domain chatbots that can detect humans' personal perceptions of social interaction, manifesting in emotional dialogs. We used iEval to benchmark four recent empathetic chatbots. Further analysis revealed several limitations in empathetic response generation approaches of these models, which came out due to their uneven abilities in handling positive and negative conversational scenarios. Based on our findings, we formulated implications informing future efforts in the development and evaluation of such chatbots. We also publicly release the data from our experiment to expedite future research in these directions.

Acknowledgements

This project has received funding from the Swiss National Science Foundation (Grant No. 200021_184602). The authors also express gratitude to Francesco Posa for the help with configuring the iEval application.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a Human-like Open-Domain Chatbot. *arXiv e-prints*, page arXiv:2001.09977.

Tatjana Aue, Stephanie Bührer, Boris Mayer, and Mihai Dricu. 2021. Empathic responses to social targets: The influence of warmth and competence perceptions, situational valence, and social identification. *PloS one*, 16(3):e0248562.

Azy Barak and Orit Gluck-Ofri. 2007. Degree and reciprocity of self-disclosure in online forums. *CyberPsychology & Behavior*, 10(3):407–417.

Lisa Feldman Barrett, Batja Mesquita, Kevin N. Ochsner, and James J. Gross. 2007. The experience of emotion. *Annual Review of Psychology*, 58(1):373–403. PMID: 17002554.

Scott Brave, Clifford Nass, and Kevin Hutchinson. 2005. Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. *International Journal of Human-Computer Studies*, 62(2):161–178. Subtle expressivity for characters and robots.

Marie Louise Caltabiano and Michael Smithson. 1983. Variables affecting the perception of self-disclosure appropriateness. *The Journal of Social Psychology*, 120(1):119–128.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114.

Franz Faul, Edgar Erdfelder, Albert-Georg Lang, and Axel Buchner. 2007. G*power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences. *Behavior Research Methods*, 39(2):175–191.

Asma Ghandeharioun, Judy Hanwen Shen, Natasha Jaques, Craig Ferguson, Noah Jones, Agata Lapedriza, and Rosalind Picard. 2019. Approximating interactive human evaluation with self-play for open-domain dialog systems. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Jeffrey T. Hancock, Christopher Landrigan, and Courtney Silver. 2007. Expressing emotion in text-based communication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 929–932, New York, NY, USA. Association for Computing Machinery.

Jennifer R. Henretty and Heidi M. Levitt. 2010. The role of therapist self-disclosure in psychotherapy: A qualitative review. *Clinical Psychology Review*, 30(1):63–77.

Tianran Hu, Anbang Xu, Zhe Liu, Quanzeng You, Yufan Guo, Vibha Sinha, Jiebo Luo, and Rama Akkiraju. 2018. Touch your heart: A tone-aware chatbot for customer care on social media. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA. Association for Computing Machinery.

C. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Ilia Kulikov, Alexander Miller, Kyunghyun Cho, and Jason Weston. 2019. Importance of search and evaluation strategies in neural dialogue modeling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 76–87, Tokyo, Japan. Association for Computational Linguistics.

Margaret Li, Jason Weston, and Stephen Roller. 2019. Acute-eval: Improved dialogue evaluation with optimized questions and multi-turn comparisons. *arXiv preprint arXiv:1909.03087*.

Qintong Li, Hongshen Chen, Zhaochun Ren, Pengjie Ren, Zhaopeng Tu, and Zhumin Chen. 2020. EmpDG: Multi-resolution interactive empathetic dialogue generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4454–4466, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhaojiang Lin, Andrea Madotto, Jamin Shin, Peng Xu, and Pascale Fung. 2019. MoEL: Mixture of empathetic listeners. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 121–132, Hong Kong, China. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. MIME: MIMicking emotions for empathetic response generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8968–8979, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Robert Plutchik. 1991. *The emotions*. University Press of America.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and

dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

D. Sculley. 2007. Rank aggregation for similar items. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 587–592.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1702–1723, Minneapolis, Minnesota. Association for Computational Linguistics.

Ekaterina Svikhnushina and Pearl Pu. 2021. Key qualities of conversational chatbots – the peace model. In *26th International Conference on Intelligent User Interfaces*, IUI '21, page 520–530, New York, NY, USA. Association for Computing Machinery.

Ekaterina Svikhnushina, Iuliana Voinea, Anuradha Welivita, and Pearl Pu. 2022. A taxonomy of empathetic questions in social dialogs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Chongyang Tao, Lili Mou, Dongyan Zhao, and Rui Yan. 2018. Ruber: An unsupervised method for automatic evaluation of open-domain dialog systems. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Joseph B. Walther, Tracy Loh, and Laura Granka. 2005. Let me count the ways: The interchange of verbal and nonverbal cues in computer-mediated and face-to-face affinity. *Journal of Language and Social Psychology*, 24(1):36–65.

Anuradha Welivita and Pearl Pu. 2020. A taxonomy of empathetic response intents in human social conversations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4886–4899, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '11)*, pages 143–146, New York. ACM Press.

Yubo Xie and Pearl Pu. 2021. Empathetic dialog generation with fine-grained intents. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 133–147, Online. Association for Computational Linguistics.

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. 2021. A comprehensive assessment of dialog evaluation metrics. In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 15–33, Online. Association for Computational Linguistics.

A Topic Clusters in EmpatheticDialogues

While working with the EmpatheticDialogues dataset (Rashkin et al., 2019), we noticed that many dialogs appear repetitive in terms of the situational scenarios brought up by the speakers. To examine it more closely, we used Sentence Transformers framework (Reimers and Gurevych, 2019) to compute vector embeddings of first speakers' turns in all dialogs and cluster them according to cosine-similarity. Figure 7 shows the empirical cumulative distribution function of topic cluster sizes in the train set of EmpatheticDialogues. From the figure, it can be seen that clusters with between 30 and 130 similar situation descriptions per cluster comprise almost 20% of the training data.

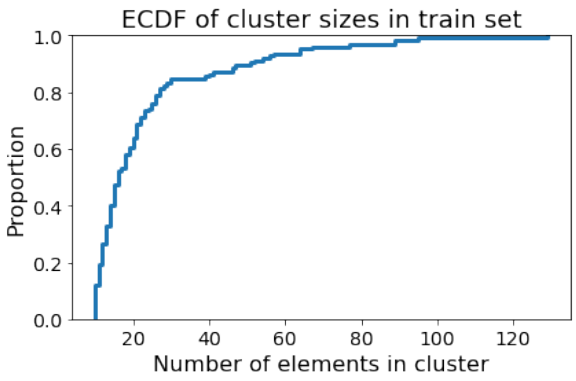


Figure 7: Empirical cumulative distribution function of topical cluster sizes in the train set of EmpatheticDialogues dataset (Rashkin et al., 2019).

	Pos: Other			Pos: Self			Pos: Both			Neg: Other			Neg: Self			Neg: Both		
	t-1	t-2	t-3	t-1	t-2	t-3	t-1	t-2	t-3	t-1	t-2	t-3	t-1	t-2	t-3	t-1	t-2	t-3
MEED	25	24	24	0	0	0	0	1	1	25	25	25	0	0	0	0	0	0
Blender	22	16	11	0	3	4	3	6	10	24	14	15	0	4	6	1	7	4
MIME	22	22	20	2	1	1	1	2	4	25	24	22	0	0	1	0	1	2
Plain	24	20	20	1	4	4	0	1	1	25	24	23	0	0	2	0	1	0

Table 2: Counts of orientation of chatbots' responses (other-, self-, or both) in 50 sampled chat logs (25 for positive and 25 for negative contexts). Prefixes "Pos" and "Neg" stand for positive and negative contexts respectively.

B Emotion Distribution in Grounding Scenarios

Figure 8 shows the distribution of original emotional labels from the EmpatheticDialogues dataset (Rashkin et al., 2019) in 480 grounding scenarios used for our benchmarking experiment. To demonstrate the even coverage of the whole emotional spectrum, we mapped 32 emotions from the dataset to 14 emotions from Plutchik's wheel (Plutchik, 1991) (8 basic and 6 intermediate emotions) and color-coded the bars in Figure 8 according to these 14 categories.

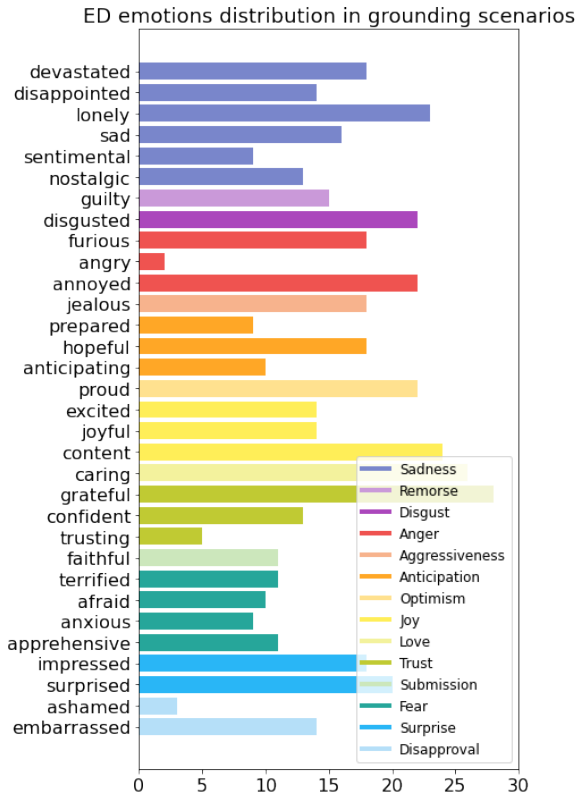


Figure 8: Distribution of emotional labels from EmpatheticDialogues dataset in grounding scenarios. The legend shows the mapping between the colors and 14 emotional categories from Plutchik's wheel (Plutchik, 1991) (8 basic and 6 intermediate emotions).

C Additional Details about Chatbots' Responses

Figure 9 depicts the average number of tokens in chatbots' responses over three dialog turns.

Table 3 shows the top-15 most frequent tokens for each of the four chatbots. As it can be noticed, question marks appear in the list of tokens of each model, pinpointing their tendency to ask questions.

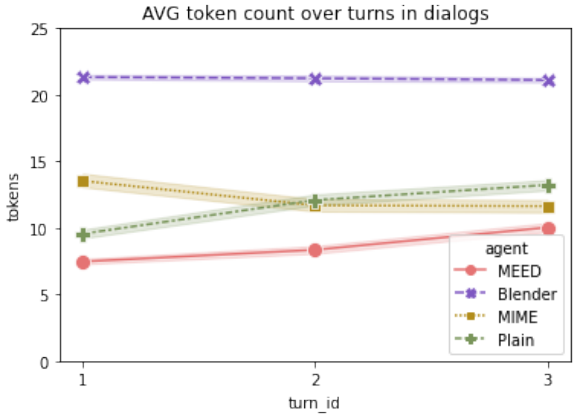


Figure 9: Counts of average number of token in chatbots' responses over three dialog turns with 95% confidence intervals.

Table 2 demonstrates the counts of orientation of chatbots' responses (other-, self-, or both) in 50 sampled chat logs (25 positive and 25 negative) over the dialog turns.

MEED	Blender	MIME	Plain
?	.	that	i
you	i	i	.
that	you	.	you
.	to	is	?
what	that	you	that
of	it	a	to
it	’s	to	!
!	a	?	sorry
a	of	am	so
i	do	!	it
’s	?	good	hear
kind	!	what	what
did	have	have	did
is	the	do	am
sounds	’m	,	of

Table 3: Top-15 most frequent tokens for each chatbot in order of decreasing frequency.

Unsupervised Domain Adaptation on Question-Answering System with Conversation Data

Amalia Istiqlali Adiba **Takeshi Homma** **Yasuhiro Sogawa**
Hitachi, Ltd.
Kokubunji, Tokyo, Japan
{amalia.adiba.dw, takeshi.homma.ps,
yasuhiro.sogawa.tp}@hitachi.com

Abstract

Machine reading comprehension (MRC) is a task for question answering that finds answers to questions from documents of knowledge. Most studies on the domain adaptation of MRC require documents describing knowledge of the target domain. However, it is sometimes difficult to prepare such documents. The goal of this study was to transfer an MRC model to another domain without documents in an unsupervised manner. Therefore, unlike previous studies, we propose a domain-adaptation framework of MRC under the assumption that the only available data in the target domain are human conversations between a *user* asking questions and an *expert* answering the questions. The framework consists of three processes: (1) training an MRC model on the source domain, (2) converting conversations into documents using document generation (DG), a task we developed for retrieving important information from several human conversations and converting it to an abstractive document text, and (3) transferring the MRC model to the target domain with unsupervised domain adaptation. To the best of our knowledge, our research is the first to use conversation data to train MRC models in an unsupervised manner. We show that the MRC model successfully obtains question-answering ability from conversations in the target domain.

1 Introduction

Conversation agents such as Siri, as well as search engines, such as Google, have been increasing the scope of user questions in which they can provide direct answers to questions that can be extracted from web pages. Providing answers directly from a structured text is often referred to as machine reading comprehension (MRC). Benefiting from deep learning technology, MRC is a question-answering (QA) task that has been extensively studied (Hermann et al., 2015; Qiu et al., 2019). MRC is used to find an answer position in a document to answer a given question. A number of large corpora have played a critical role in advancing MRC research (Rajpurkar et al., 2016; Trischler et al., 2017; Bajaj et al., 2016; Zhang et al., 2018). Many MRC studies have focused on developing new model structures by introducing a new end-to-end neural network model to obtain state-of-the-art performance (Huang et al., 2018, 2019; Shen et al., 2017; Seo et al., 2017; Xiong et al., 2017). However, these state-of-the-art models were evaluated in one domain. In fact, it has been proven that the generalization capabilities of MRC models do not perform well on different datasets (Yogatama et al., 2019).

Unsupervised domain adaptation is an approach to cope with transferring knowledge from a source domain to a different unlabeled target domain (Pan and Yang, 2010). To provide labels for a new domain dataset, question generation is commonly used to create synthetic data consisting of question-answer pairs from documents of the target domain (Rus et al., 2010), so that an MRC model can be trained with both data from the source domain and syntactic data from the target domain (Yue et al., 2021; Lee et al., 2020; Puri et al., 2020; Shakeri et al., 2020; Cao et al., 2020; Wang et al., 2019).

One critical issue in unsupervised domain adaptation for MRC is that previous studies assumed that the input documents must be available in the target domain. There are also many types of information (not limited to the document) in a real-word scenario. To apply MRC to such information, it is necessary to convert the information into documents. However, this conversion is not an easy task.

Let us take a case in customer service support. Human operators in a customer service usually refer to "manual documents" containing necessary knowledge to answer customer questions. The manual usually has limited information. Thus, when there is no information to answer questions, the operators pass the call to a supervisor, and the su-

Proceedings of the SIGdial 2022 Conference, pages 432–441
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

pervisor continues to talk with the customer to answer the question. This procedure is called "escalation". The frequency of escalation is not trivial. Moreover, the number of supervisors is usually few, thus it is necessary to reduce escalations. It is obvious that conversations between supervisors and customers have plenty of information that is not included in the document. If we add new information to the document based on supervisor-customer conversations, the MRC task can answer more varied questions.

In domain adaptation for MRC, in which the conversation between an *expert* and *user* is the only available data in the target domain, has become a new challenge. The user asks questions and the expert answers the questions. To address this challenge, we propose a framework of domain adaptation of MRC. This framework consists of three processes: (1) training an MRC model on the source domain, (2) converting conversations into documents using document generation (DG), which is a task we developed for retrieving important information from several human conversations and converting it to an abstractive document text, and (3) transferring the MRC model to the target domain with unsupervised domain adaptation, which consists of two stages; self-training and discriminative learning.

Our contributions are summarized as follows:

- We propose a framework of unsupervised domain adaptation of MRC in which the only available data are unlabeled human conversations in the target domain.
- We evaluated MRC models with four different domain data.

2 Related Work

2.1 Machine Reading Comprehension (MRC)

With the wide use of deep learning, significant progress has been achieved on many natural-language-processing tasks including MRC. Hermann et al. (2015) proposed an MRC model using bidirectional long short-term memory to capture the context of documents. The idea has become the foundation of many MRC models. It is a challenging task how to make a machine imitate a human to understand the document and be able to answer questions. A large dataset has played a critical role in progressing MRC research. Rajpurkar et al. (2016) released SQuAD (the Stanford Question Answering Dataset), which contains more than 100,000 sets of a question, answer, and document. After that, the contributions of MRC can be grouped into four categories: developing new model structures, creating new datasets, multi-task learning, and introducing a new evaluation method (Baradaran et al., 2022). Many MRC studies have focused on developing model structures by introducing an end-to-end neural network model to obtain state-of-the-art performance (Huang et al., 2018, 2019; Shen et al., 2017; Seo et al., 2017; Xiong et al., 2017). In a different direction, other papers have focused on creating new datasets (Feng et al., 2020; Choi et al., 2018; Reddy et al., 2019; Campos et al., 2020). The main trend in these papers was to create datasets considering more complex phenomena, i.e., a query is formed by multiple turns and a document has structural elements. Some papers addressed methods to evaluate whether the system acquires a "true" comprehension capability (Jia and Liang, 2017; Wang and Bansal, 2018). To test true comprehension capability, for instance, QA performance was measured when documents were made distracting by inserting adversarial noisy sentences.

2.2 Document Generation (DG)

MRC returns no answer for irrelevant questions to the documents. Self-learning of the MRC model from human conversations is a new challenge. To achieve this, converting human conversations into documents is necessary. We call this task document generation (DG). DG is a similar task to conversation summarization, which focuses on simply extracting important context from conversations (Li et al., 2019). Unlike conversation summarization, however, DG is aimed to use for a specific application, i.e., customer service. Therefore, it is necessary to extract useful information for the application from conversation contexts, e.g., the topics of customer queries and solution the operator provides. Document summarization is divided into two types of methods: extractive and abstractive. Extractive methods select key sentences from original documents (Knight and Marcu, 2000), while abstractive methods highlight key phrases and compactly rewrite them (Gehrmann et al., 2018; Maynez et al., 2020; Chen and Bansal, 2018). DG should ensure the correctness of key facts. For example, when an operator asked, "Were the plates lost or stolen?", and a customer said,

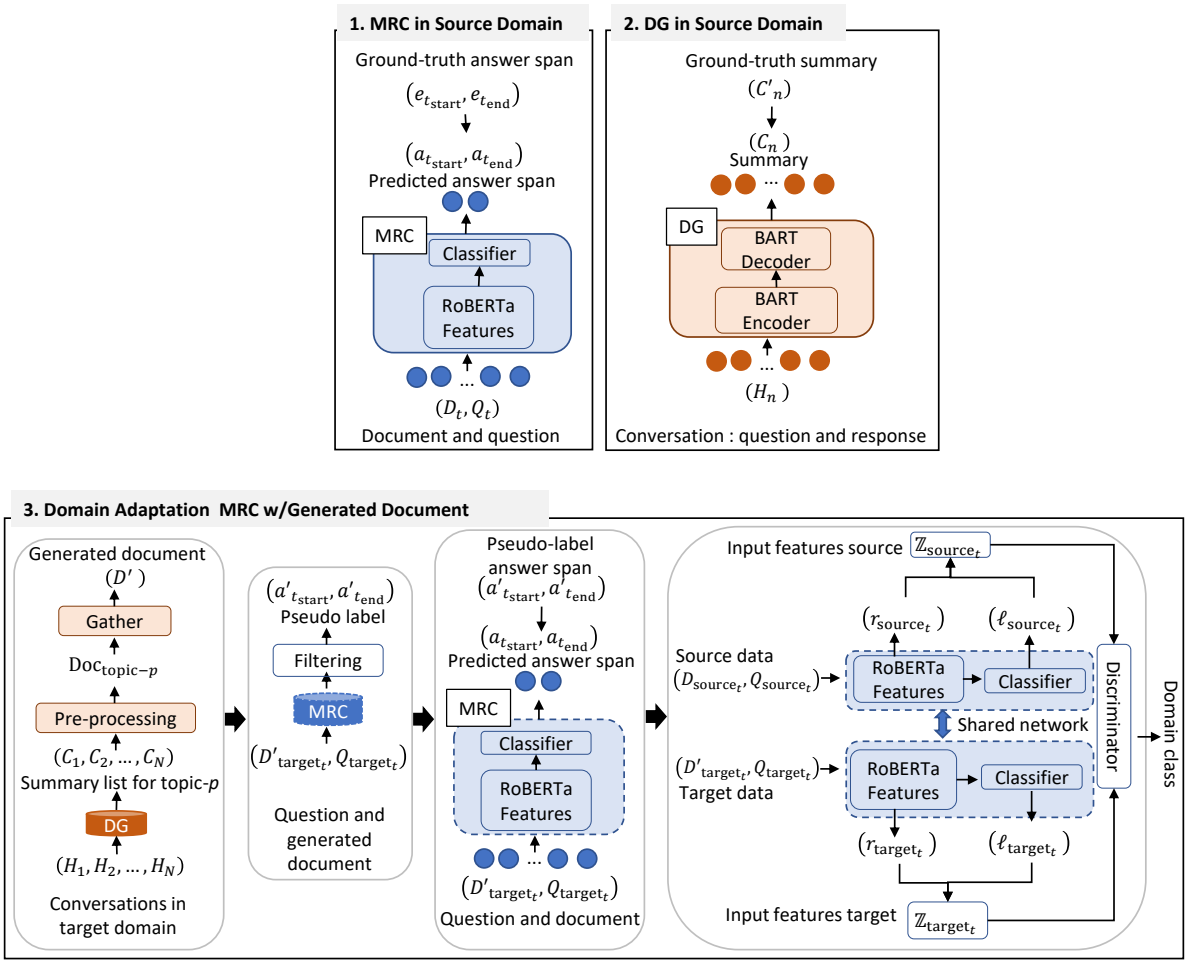


Figure 1: Proposed domain-adaptation framework that includes our developed DG for MRC. Parameters of modules in dash boxes are updated during domain adaptation.

"No", then the operator's response is "You will not be eligible for a refund". In this case, a key sentence that should be included in the document is a sentence such as "You are eligible for a refund if the plates are lost/stolen or destroyed".

2.3 Domain Adaptation

One of the hot topics in MRC is developing simultaneous learning of multiple tasks (transfer learning) (Ruder et al., 2019) and transferring the learned MRC model from one domain to another. This is a promising task for obtaining better results, especially in a data-poor setting. The task is referred to as domain adaptation, which can be divided into two types of methods; supervised, and unsupervised. With supervised methods, the model is trained, where the label is available in the target domain (Kratzwald and Feuerriegel, 2019). The aim of supervised domain adaptation in MRC is to enlarge the number of domains the learned model can cope with. With unsupervised methods, no labeled information is available in the target domain. Cao et al. proposed an unsupervised domain-adaptation method on reading comprehension (Cao et al., 2020). They first trained the MRC model in a source domain by fine-tuning a Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2019), then in the adaptation stage, the fine-tuned model is used to generate synthetic question-answer pairs in the target-domain documents, and the synthetic pairs are used in self-training. Their method worked with an assumption that questions and documents are available in the target domain. Wang et al. proposed a similar method (Wang et al., 2019). The difference is that they used a question generator to extract questions from documents in the target domain. Although their method showed promising results, current MRC cannot handle irrelevant questions that have no information in the given document.

Our proposed framework is on unsupervised domain adaptation tasks. Unlike the above-mentioned studies, we used human conversations, which are the only available data in the target domain as the input.

3 Proposed Framework

The main objective of our research is to develop an MRC technique with which the MRC model can be automatically updated on the basis of human conversations. To achieve this, we add DG to the MRC pipeline. The role of DG is to convert human conversations to documents. The generated documents are then added to the training data of the MRC model.

Let us assume that we have two different types of domain data: source domain that has conversations and corresponding documents and target domain that has only conversations. If we have an MRC model trained with source domain data, our goal is to update the model to cover target-domain questions. However, the target domain has no document related to the conversations. Thus, the MRC model should be trained with the only available conversation data in the target domain. As shown in Fig. 1, our framework consists of the following three processes.

1. Training an MRC model with answer spans for given questions and corresponding documents data in the source domain.

2. Converting conversations to documents by DG through model training with source-domain data. Given human conversation as an input, the MRC model returns a summary of the conversation.

3. Transferring the MRC model to the target domain with unsupervised domain adaptation. There are two stages; self-learning to train the MRC model with synthetic data and discriminative learning to learn the feature distribution between source and target domains. Thus, the model can provide the answers of questions from both source and target domains.

3.1 Machine Reading Comprehension in Source Domain

Let $M_{\text{source}} = (D, Q, A)$ denote an MRC dataset in the source data, where D, Q, and A represent documents, questions, and answer span for the questions, respectively. A question contains not only the *user*'s question but also the dialogue history between the *user* and *expert*. An MRC model $\mathscr{M}$ takes documents $D = (d_1, d_2, ..., d_{T_{\text{source}}})$ and questions $Q = (q_1, q_2, ..., q_{T_{\text{source}}})$ as input, where T_{source} is the amount of data in the source domain. The model is trained to predict the correct answer spans:

$$A = ([e_{1_{\text{start}}}, e_{1_{\text{end}}}], ..., [e_{T_{\text{source}_{\text{start}}}}, e_{T_{\text{source}_{\text{end}}}}]) \quad (1)$$

We use Transformer models to implement the MRC model in the source domain. The Transformer encoder is used to contextually represent the question along with the document. Question q_t and document d_t are passed to the Transformer encoder to create contextual representations of the input. To obtain the starting and ending indices of the answer, the encoder output is sent to a linear layer to be converted into logits corresponding to the probabilities of being the start index ($a_{t_{\text{start}}}$) and end index ($a_{t_{\text{end}}}$) of the answer span.

The $a_{t_{\text{start}}}$ and $a_{t_{\text{end}}}$ are optimized by minimizing the following loss function:

$$\mathcal{L} = \frac{1}{2}(CEL(e_{t_{\text{start}}}, a_{t_{\text{start}}}) + CEL(e_{t_{\text{end}}}, a_{t_{\text{end}}})), \quad (2)$$

where CEL is the cross-entropy loss function, and $e_{t_{\text{start}}}$ and $e_{t_{\text{end}}}$ are the labels at token number t for the answer start and end indices $a_{t_{\text{start}}}$ and $a_{t_{\text{end}}}$, respectively.

3.2 Document Generation

Given an input dialogue between a *user* and *expert*, the goal of DG is to produce a multi-sentence summary that captures the highlights of the dialogue. Let N be the total number of dialogues consisting of a conversation about topic-p. By giving a dialogue context H, the goal is to generate the summary C of the dialogue. The n-th dialogue has a list of utterances $H_n = (h_1, h_2, ..., h_L)$, where h_l is the l-th utterance in the dialogue and L is the number of utterances. Each utterance contains a sequence of tokens $h_l = (x_{l,\text{role}}, x_{l,1}, x_{l,2}, ..., x_{l,n_l})$, where $x_{l,j}$ is the j-th token in h_l and n_l is the number of tokens in the l-th role's utterance. At the beginning of the sequence, we add a special token $x_{l,\text{role}}$, which represents the role of the speaker, i.e., $x_{l,\text{role}} \in (\textit{user}, \textit{expert})$. The n-th output from DG is also a sequence of word tokens $C_n = (c_1, c_2, ..., c_K)$, where K is the number of tokens. Note that the highlight in C_n is just some

of the information in topic-p. To gather all information and generate a document of topic-p, we collect the highlights from all conversations, remove duplicate sentences, then put the highlights together as a completed document.

The ground truth summary C'_n is created by combining all the correct answer of the *user*'s question for corresponding input dialogue H_n, where *user*'s questions $\in Q$. Given a *user*'s question q and its answer's spans ($e_{\text{start}}, e_{\text{end}}$), the correct answer sentence w is taken from the original document. Thus, if there are y number of utterances in input dialogue H_n in which the role is *user*, the ground truth summary is a series of consecutive sentences $w_1, w_2, ..., w_y$.

We use a Transformer (Vaswani et al., 2017) model built with a seq2seq model combining an encoder with a decoder. Studies have shown that if the model is first trained on a large corpus, it will learn the distribution of that corpus vocabulary (Gururangan et al., 2020). Motivated by this, we experimented with pre-training on different out-of-domain datasets, such as the news-based CNN/Daily Mail corpus (Nallapati et al., 2016) and conversation-based SAMSum corpus (Gliwa et al., 2019), and continued to fine-tune the model on our experimental dataset in the source domain. We first trained the model for the summarization task on the large CNN/Daily Mail corpus. The reason we first train the model with this corpus is that the corpus has high quality contexts and summaries that can enable the model to learn a structured document. However, our main focus is not on summarizing an article but to generate highlights from conversation data. Thus, we continue fine-tuning the model for the summarization task with a conversation-based corpus, such as SAMSum, to obtain more auxiliary vocabulary.

The trained DG model in the procedure above will be used to generate documents in the target domain by giving conversations in the target domain.

3.3 Unsupervised Domain Adaptation in Target Domain

The unsupervised domain adaptation in our framework consists of two stages; self-learning and discriminative learning.

In the self-learning stage, we have to generate pseudo-label samples. The MRC model $\mathscr{M}$ described in Section 3.1 is used to provide pseudo-labels to unlabeled documents in the target domain generated with DG trained in Section 3.2. However, because the predicted output consists of false answers, we have to choose reliable pseudo-labels. Thus, the underlying assumption in this stage is we only take the samples having high-confidence predictions. Retraining the model using high-confidence samples will further improve its performance (Saito et al., 2017). Despite the fact that the distribution of vocabulary is different between source and target domains, both domains may have similar characteristics. Thus, some samples with high-confidence scores will be similar to or the same as correct answer spans in the target domain. To provide pseudo-labels, we first gather a set of answer spans that have the top n_{best} answer-span probabilities $\hat{p}_t$. The $\hat{p}_t$ is calculated using a softmax function applied to the sums of start index logits $a'_{t_{\text{start}}}$ and end index logits $a'_{t_{\text{end}}}$. We assign a pseudo-label to q_t if the following two conditions are satisfied. First, $\hat{p}_t$ should exceed the threshold parameter (th_{prob}), which we set in the experiment. The second requirement is that the span should not be an empty text. After the pseudo-labeled training set (M'_{target}) is composed, $a_{t_{\text{start}}}$ and $a_{t_{\text{end}}}$ are updated on the basis of the loss in Eq. (1), except we replace $e_{t_{\text{start}}}$ and $e_{t_{\text{end}}}$ with $a'_{t_{\text{start}}}$ and $a'_{t_{\text{end}}}$, respec-

Algorithm 1 Domain adaptation of MRC with DG. $\mathscr{M}$ is MRC source model, D' is generated document in target domain derived from DG model, and $iter_{\text{DA}}$ is training-epoch number for domain adaptation.

Input: $M_{\text{source}} = \{(D_t, Q_t)\}_{t=1}^{T_{\text{source}}}$, ▷ Source data
$M_{\text{target}} = \{(D'_t, Q_t)\}_{t=1}^{T_{\text{target}}}$, ▷ Target data
$\mathscr{M}$
Output: Optimal model $\mathscr{M}$ in the target domain

$M'_{\text{target}} = \emptyset$
for $j \leftarrow 1$ to $iter_{\text{DA}}$ **do**
  for $t \leftarrow 1$ to T_{target} **do** ▷ Pseudo-labeled generation
    Use $\mathscr{M}$ to predict the pseudo-labels $a'_{t_{\text{start}}}$ and $a'_{t_{\text{end}}}$ for (D'_t, Q_t) and obtain probability $\hat{p}_t$
    if $\hat{p}_t \geq th_{\text{prob}}$ **and** $D'_t(a'_{t_{\text{start}}}, a'_{t_{\text{end}}}) \neq$ empty text **then**
      if $Q_t \notin M'_{\text{target}}$ **then**
        Put $(D'_t, Q_t, a'_{t_{\text{start}}}, a'_{t_{\text{end}}})$ into M'_{target}
      end if
    end if
  end for
  for mini-batch $\mathscr{b}$ **in** M'_{target} **do** ▷ Self training
    Train $\mathscr{M}$ with $\mathscr{b}$
  end for
  for mini-batch $\mathscr{b}_{\text{target}}$ **in** M'_{target} **and** $\mathscr{b}_{\text{source}}$ **in** M_{source} **do** ▷ Discriminative learning
    Train $\mathscr{M}$ and $\mathscr{D}$ with $\mathscr{b}_{\text{target}}$, $\mathscr{b}_{\text{source}}$, and domain labels
  end for
end for

tively. In each epoch during adaptation training, pseudo-labeled samples are updated using the last model. An additional sample t will be added if the t-sample did not exist in the last pseudo-labeled samples.

The discriminative-learning stage is used for the MRC model to learn the difference in the feature distribution between source and target domains. We combine the following two representation outputs from both source and target samples: (1) the last hidden state $r \in \mathbb{R}^{\jmath \times \hbar}$, which is the output of the last layer of the MRC model, and (2) concatenation of start-logits and end-logits, which outputs ℓ with dimension $\jmath \times 2$. Note that $\jmath$ and $\hbar$ are the maximum input sequence length and hidden state dimension, respectively. We set $\jmath = 2 \times \hbar$. The input feature $\mathbb{Z}$ is calculated with the following process:

$$\mathbb{Z} = \ell \odot avg_{\text{col}}(r), \tag{3}$$

where avg_{col} means the average along columns, which returns a vector in $\in \mathbb{R}^{\hbar}$, $\odot$ is an element-wise product, $\ell \in \mathbb{R}^{\hbar}$, and $\mathbb{Z} \in \mathbb{R}^{\hbar}$. Discriminator $\mathcal{D}$ takes $\mathbb{Z}$ as input and computes the probability using a neural network consisting of three linear layers, in which the final layer outputs a one-dimensional value that shows the output probability.

The loss function is the binary cross-entropy loss,

$$\mathcal{L}_{\text{dsc}} = -\left(u \log(\hat{u}) + (1-u)\log(1-\hat{u})\right), \tag{4}$$

where $\hat{u}$ is the probability output from $\mathcal{D}$, and $u \in \{0, 1\}$ is the ground-truth label; 0 for the source domain and 1 for the target domain. The entire procedure of domain adaptation is shown in Algorithm 1.

4 Experiments

4.1 Dataset

In our experiments, we used the Doc2dial (Feng et al., 2020) dataset consisting of about 4,800 annotated conversations with an average of 14 turns per conversation. The utterances are grounded in over 480 documents from four domains of public government service websites in the U.S.: Social security administration (**ssa**), Department of Motor Vehicles (**dmv**), Federal Student Aid (**studentaid**), Veteran's Affairs (**va**).

In the training process of the DG model, as we mentioned in Section 3.2, we first trained the model on the large CNN/Daily Mail corpus (Nallapati et al., 2016). This corpus is based on the news articles taken from the CNN and Daily Mail websites. It includes various subjects such as travel and business. It also contains about 300,000 articles written by journalists at CNN and the Daily Mail. We continued to fine-tune the model in the conversation-based SAMSum corpus (Gliwa et al., 2019). The SAMSum corpus is an English dataset consisting of about 15,000 natural conversations in various scenes of real life such as chatting, meeting arrangements, and political discussion. We finally fine-tuned the model in the Doc2dial dataset.

4.2 Hyper-parameters

We implemented the QA from HuggingFace Transformers (Wolf et al., 2019) with a pre-trained model as the encoder and fine-tuned it on the Doc2dial dataset during training. We used two different pre-trained models as the MRC models in the source domain: BERT (Devlin et al., 2019), and Robustly Optimized BERT Approach (RoBERTa) (Liu et al., 2019). Since the grounded document is often longer than the maximum input sequence length for the QA model, we followed a previous study (Feng et al., 2020) to truncate the documents in windows with a stride. We set the stride to 128 tokens, the number of epochs to 5 with cross entropy as the loss function, and the learning rate to 3×10^{-5}. The batch size was set to 15, and the maximum distance between starting (a_{start}) and ending (a_{end}) indices of answers was set to 50.

In the training process of the DG model, we used BART$_{large}$, which includes 12 Transformer layers in the encoder and decoder [1]. We set the number of epochs to 5 with cross entropy as the loss function and set the learning rate to 3×10^{-5}. We set the batch size to 15 and maximum length of input sequences to 1024.

In the unsupervised domain-adaptation process, we set the learning rate to 3×10^{-5} in the self-training stage and 5×10^{-5} in the discriminative-learning stage. We set the same parameters as in MRC in source-domain training for the maximum distance between starting and ending and number of epochs ($iter_{\text{DA}}$). The batch size was set to 5. The input dimension of the first layer in the discriminator network ($\hbar$) was 1024, and the maximum sequence ($\jmath$) was 512. We used a rectified linear unit as the activation function in the first two

[1] https://huggingface.co/facebook/bart-large

layers. The threshold (th_{prob}) was set to 0.5, and n_{best} was 20.

All parameters were determined on the basis of the best ROUGE-1 score for training the DG model and F1 score for MRC models (source and domain adaptation) on the validation dataset in the experiments.

5 Results and Discussion

5.1 MRC in Source Domain

	Specific Domain				All Domains
	ssa	**dmv**	**studentaid**	**va**	
BERT	61.29	53.88	50.99	68.77	62.83
RoBERTa	**64.93**	**63.13**	**62.86**	**73.01**	**70.30**
Baseline	-	-	-	-	65.30

Table 1: The F1 scores [%] for MRC in source domain on Doc2dial validation set. The baseline score is reported in (Feng et al., 2020).

For the MRC source training, we compared the F1 score results (shown in Table 1) between two different language models: BERT and RoBERTa. We trained and evaluated the MRC model with a specific domain, and with all the domain data. With BERT, which is the same Transformer model as the baseline, we obtained an F1 score of 62.83%. This result is lower than the reported baseline (Feng et al., 2020) of 65.30%. However, with RoBERTa, we obtained a higher score of 70.30%. Therefore, we used RoBERTa to train unsupervised domain adaptation of the MRC model in the target domain.

5.2 Document Generation

Dataset	Evaluation Metrics [%]	
	ROUGE-1	ROUGE-L
Doc2dial	67.02	44.06
Doc2dial + CNN/Daily Mail	69.74	45.26
Doc2dial + CNN/Daily Mail + SAMSum	**69.94**	**45.71**

Table 2: DG results on Doc2dial validation set during further pre-training on different QA datasets. We trained with the BART model.

The performances of DG are listed in Table 2. Experiments with BART on the validation set showed that fine-tuning on different datasets is beneficial. Pre-training on more structural corpora, such as CNN/Daily Mail, is more useful than directly fine-tuning BART into the Doc2dial dataset. Furthermore, training the model using SAMSum, which contains conversational data and is more similar to Doc2dial, further improved the performance. An example of the generated document results is shown in Table 3. When we increase the conversation, new information will be added to the generated document. Current generated documents gather all information that is based on the conversation. Thus, the output results will significantly differ compared with the original document in the target domain, especially for the information structure. The information order depends on the given conversation order.

Conversation in VA [Veterans' Affairs] claim topic		
U	:	how do you check your VA claim or appeal status?
E	:	find out how to check the status of a VA claim or appeal online
U	:	can I use the tool?
E	:	do you have one of the following accounts? A Premium My HealtheVet account a Premium DS Logon account used for eBenefits and milConnect , or one you can create here on VA.gov verified ID.me?
U	:	yes
E	:	ok you just log into one of those
Generated document for VA claim topic		
Log in to start finding out how to check the status of a VA claim or appeal online. Use this tool if you have one of the following accounts: A Premium DS Logon account used for eBenefits and milConnect or Verified ID.me.		
Ground truth in original document for VA claim topic		
Check your VA claim or appeal status. Find out how to check the status of a VA claim or appeal online. To use this tool, you'll need to have one of these free accounts: A Premium My HealtheVet account or A Premium DS Logon account used for eBenefits and milConnect, or one you can create here on VA.gov verified ID.me account.		

Table 3: Given conversation between *expert* (E) and *user* (U), DG returns the generated document in the corresponding topic. We used the Doc2dial + CNN/Daily Mail + SAMSum model.

5.3 MRC with Domain Adaptation

5.3.1 With Original Target Documents

Domain (*source* to *target*)	w/o DA	with DA
studentaid to **va**	50.62	**53.27**
studentaid to **ssa**	49.38	**55.12**
studentaid to **dmv**	54.93	**60.19**

Table 4: The F1 scores [%] for MRC without DG when using **studentaid** data as source domain. DA refers to domain adaptation.

We conducted a domain-adaptation test with original target documents to verify the effective-

Domain			studentaid		va		ssa		dmv	
			ROUGE-L	F1	ROUGE-L	F1	ROUGE-L	F1	ROUGE-L	F1
studentaid	w/o DA				53.02	50.62	47.62	49.38	52.04	54.93
	with DA	w/o DSC			53.58	51.63	48.79	50.58	51.55	55.29
		with DSC			**54.26**	**52.45**	**49.70**	**51.02**	**52.17**	**55.61**
va	w/o DA		54.60	52.54			48.97	51.46	53.92	57.85
	with DA	w/o DSC	54.44	52.05			48.72	50.07	52.82	56.24
		with DSC	**55.76**	**53.68**			**49.13**	**51.64**	**53.99**	**57.90**
ssa	w/o DA		51.86	47.60	53.90	51.79			52.02	54.01
	with DA	w/o DSC	51.56	48.52	54.15	53.03			53.14	56.28
		with DSC	**52.48**	**50.12**	**55.05**	**53.12**			**53.32**	**56.95**
dmv	w/o DA		54.66	52.76	53.66	52.06	52.25	56.73		
	with DA	w/o DSC	53.71	52.35	53.71	52.22	51.64	55.99		
		with DSC	**55.68**	**55.07**	**53.77**	**53.08**	**53.33**	**57.86**		

Table 5: MRC results with the document generation (DG). DA refers to domain adaptation and DSC refers to discriminative learning.

ness of a domain-adaptation stage. We first trained an MRC model in the source domain with **studentaid** data. The next procedure was the same as that shown in Algorithm 1, except we used the original document D in the target domain. We set three domain-adaptation dataset pairs, which were **studentaid** to **va**, **studentaid** to **ssa**, and **studentaid** to **dmv**. As shown in Table 4, the F1 scores of the model trained without/with domain adaptation (DA) were 50.62/53.27, 49.38/55.12, and 54.93/60.19% for **studentaid** to **va**, **studentaid** to **ssa**, and **studentaid** to **dmv**, respectively. Thus, the model trained with DA (our framework) outperformed the model trained without DA.

5.3.2 With Generated Target Documents by DG

Finally, we conducted an experiment for our main task, in which the model is trained with unsupervised DA and with DG. The results for each domain are listed in Table 5. We tested under three conditions: the model trained without DA, model trained with DA and without the discriminative-learning stage, and model trained with both DA and discriminative-learning stage. The results indicate that for the model trained with DA, self-learning alone (without discriminative stage) was not strong enough to outperform the model trained without the DA model. We observed that the number of generated pseudo-labeled sets (M'_{target}) remained almost the same in each epoch, such as in **studentaid** to **dmv**. Consequently, the model trained with DA but without the discriminative-learning stage performed worse than the model trained without DA. For **ssa** to **dmv**, the number of generated pseudo-label sets increased during the training process. Thus, the model trained with DA but without the discriminative-learning stage outperformed the model without DA. Despite 1 or 2% improvement, as we add the discriminative stage to the DA-model training, the model trained with both DA and the discriminative-learning stage outperformed the model trained without DA in all datasets. Even with unstructured documents and without labels in the target domain, we proved that our framework can be used to adapt the model from conversation data.

6 Conclusion

We proposed a framework of unsupervised domain adaptation of MRC in which the only available data are unlabeled human conversations in the target domain. DG, which is a task in the framework, converts a given conversation into a document including conversational context. We also tackled a new challenge of conducting domain adaptation from the source domain with a structured document to a new domain with an unstructured document. We showed that only self-learning does not always improve accuracy. However, discriminative learning with self-learning successfully improved conversational-based MRC domain adaptation.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani. 2022. A survey on machine reading com-

prehension systems. *Natural Language Engineering*, pages 1–50.

Jon Ander Campos, Arantxa Otegi, Aitor Soroa, Jan Deriu, Mark Cieliebak, and Eneko Agirre. 2020. DoQA – Accessing domain-specific FAQs via conversational QA. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7302–7314.

Yu Cao, Meng Fang, Baosheng Yu, and Joey Tianyi Zhou. 2020. Unsupervised domain adaptation on reading comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7480–7487.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-up abstractive summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4098–4109.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, and Weizhu Chen. 2018. FusionNet: Fusing via fully-aware attention with application to machine comprehension. In *International Conference on Learning Representations*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031.

Kevin Knight and Daniel Marcu. 2000. Statistics-based summarization – Step one: Sentence compression. In *Proc. AAAI Conference on Artificial Intelligence*, pages 703–710.

Bernhard Kratzwald and Stefan Feuerriegel. 2019. Putting question-answering systems into practice: Transfer learning for efficient domain customization. *ACM Transactions on Management Information Systems (TMIS)*, 9(4):1–20.

Dong Bok Lee, Seanie Lee, Woo Tae Jeong, Donghwan Kim, and Sung Ju Hwang. 2020. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 208–224.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gùlçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826.

Boyu Qiu, Xu Chen, Jungang Xu, and Yingfei Sun. 2019. A survey on neural machine reading comprehension. *arXiv preprint arXiv:1906.03824*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18.

Vasile Rus, Brendan Wyse, Paul Piwek, Mihai Lintean, Svetlana Stoyanchev, and Cristian Moldovan. 2010. The first question generation shared task evaluation challenge. In *Proceedings of the 6th International Natural Language Generation Conference*.

Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 2988–2997.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Huazheng Wang, Zhe Gan, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Hongning Wang. 2019. Adversarial domain adaptation for machine reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2510–2520.

Yicheng Wang and Mohit Bansal. 2018. Robust machine comprehension models via adversarial training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 575–581.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dynamic coattention networks for question answering. In *International Conference on Learning Representations*.

Dani Yogatama, Cyprien de Masson d'Autume, Jerome T. Connor, Tomás Kociský, Mike Chrzanowski, Lingpeng Kong, Angeliki Lazaridou, Wang Ling, Lei Yu, Chris Dyer, and Phil Blunsom. 2019. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*.

Zhenrui Yue, Bernhard Kratzwald, and Stefan Feuerriegel. 2021. Contrastive domain adaptation for question answering using limited text corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9575–9593.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. ReCoRD: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

UniDU: Towards A Unified Generative Dialogue Understanding Framework

Zhi Chen[1], **Lu Chen**[1]*, **Bei Chen**[2], **Libo Qin**[2], **Yuncong Liu**[1], **Su Zhu**[3], **Jian-Guang Lou**[2] and **Kai Yu**[1]*
[1]X-LANCE Lab, Department of Computer Science and Engineering
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
State Key Lab of Media Convergence Production Technology and Systems, Beijing, China
[2]Microsoft Research Asia
[3]AISpeech Co., Ltd., Suzhou, China

Abstract

With the development of pre-trained language models, remarkable success has been witnessed in dialogue understanding (DU). However, current DU approaches usually employ independent models for each distinct DU task without considering shared knowledge across different DU tasks. In this paper, we propose a unified generative dialogue understanding framework, named *UniDU*, to achieve effective information exchange across diverse DU tasks. Here, we reformulate all DU tasks into a unified prompt-based generative model paradigm. More importantly, a novel model-agnostic multi-task training strategy (MATS) is introduced to dynamically adapt the weights of diverse tasks for best knowledge sharing during training, based on the nature and available data of each task. Experiments on ten DU datasets covering five fundamental DU tasks show that the proposed UniDU framework largely outperforms task-specific well-designed methods on all tasks. MATS also reveals the knowledge-sharing structure of these tasks. Finally, UniDU obtains promising performance in the unseen dialogue domain, showing the great potential for generalization.

1 Introduction

The development of the conversational system plays an important role in the spread of the intelligence devices, such as intelligence assistants and car play. In recent years, there has been a growing interest in neural dialogue system (Wen et al., 2017; Ultes et al., 2017; Li et al., 2017; Chen et al., 2018a, 2019, 2020b; Bao et al., 2020; Adiwardana et al., 2020; Ham et al., 2020; Peng et al., 2020; Chen et al., 2022). Dialogue understanding is a core technology and hot topic in the dialogue system, aiming to analyze a dialogue from different fine-grained angles accurately.

There are five classical dialogue understanding tasks: dialogue summary (DS) (Liu et al., 2019a), dialogue completion (DC) (Su et al., 2019; Quan et al., 2020), intent detection (ID) (Kim et al., 2016; Casanueva et al., 2020; Qin et al., 2021a), slot filling (SF) (Zhang et al., 2017; Qin et al., 2021b; Haihong et al., 2019) and dialogue state tracking (DST) (Kim et al., 2020; Chen et al., 2020a; Hosseini-Asl et al., 2020; Xu et al., 2020; Liao et al., 2021). Dialogue summary aims to generate a concise description of given dialogue content, which is normally formulated as a sequence-to-sequence generation problem (Wu et al., 2021). Dialogue completion eliminates the co-reference and information ellipsis in the latest utterance, which is also a generation task (Chen et al., 2021b). Intent detection and slot filling are two traditional spoken language understanding tasks that aim to map natural language to logical form. Intent detection is typically treated as a classification problem (Liu and Lane, 2016) and slot filling is usually formulated as a sequence labeling task (Zhang et al., 2017; Qin et al., 2019; Coope et al., 2020). The dialogue state tracking task is to extract the user's constraints on the predefined dialogue domains and slots (Budzianowski et al., 2018). The five different tasks aim to interpret a dialogue from five different perspectives. To date, these DU tasks are still learned independently due to different task formats. However, they are intuitively related. For example, the dialogue completion task should have a positive effect on the dialogue state tracking task (Han et al., 2020). On the other hand, it is usually very expensive to collect dialogue data and annotate them, which constraints the scale of annotated dialogue corpora. It is important and imperative to study how to enhance dialogue understanding capability with the existing diverse dialogue corpora.

There are two main challenges in knowledge sharing across DU tasks: *data annotation* diversity and *task nature* diversity. It is necessary to employ

*The corresponding authors are Lu Chen and Kai Yu.

Proceedings of the SIGdial 2022 Conference, pages 442–455
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

a unified DU model to allow all types of DU data to be used together. In this paper, we propose a **Uni**fined **D**ialogue **U**derstanding (UniDU) framework, in which the five fundamental DU tasks are modelled by a unified sequence-to-sequence generative model. The second challenge is related to the nature of diverse tasks. Since the output label dynamic ranges and the goals of the DU tasks are different, tasks may not be well suited to be trained together with straightforward multi-task learning. It is then a nontrivial problem to effectively weight diverse tasks for the unified model with different dialogue corpora. In this paper, we propose a novel adaptive weighting approach and compare it with other different training strategies under the UniDU framework.

The main contributions of this paper are summarized below:

- To the best of our knowledge, we are the first to formulate different dialogue understanding tasks as a unified generation task spanned five DU tasks. The proposed UniDU outperforms well-designed models on five well-studied dialogue understanding benchmarks.
- We propose a model-agnostic adaptive weighting approach for multitask learning to address the task nature diversity problem. We find that the intuitive multitask mixture training method makes the unified model bias convergence to more complex tasks. The proposed model-agnostic training method can efficiently relieve this problem.
- Experimental results show that the proposed UniDU method has excellent generalization ability, which achieves advanced performance both on few-shot and zero-shot setups.

2 Dialogue Understanding Tasks

We denote dialogue context as $C = (H_n, U_n)$, where $H_n = (U_1, U_2, \ldots, U_{n-1})$ represents the dialogue history containing the first $n-1$ turns of utterances. U_n is n-th turn utterance, which may consist of multiple sentences stated by one speaker. For the task-oriented dialogue, the domain scope is restricted by the dialogue ontology, which the dialogue expert designs. The ontology O is composed of dialogue domains $D = \{d\}$ (like *hotel*), domain slots (like *price*) $S = \{s\}$ and user intent candidates $I = \{i\}$ (like *find_hotel*). There are five fundamental tasks to interpret a dialogue from different perspectives.

Dialogue Summary (DS) aims to extract important information of the dialogue. It is a typical generation problem, which takes the whole dialogue context C as input and generates the summary description. DS requires the model to focus on the whole dialogue flow and the important concepts.

Dialogue Completion (DC) purposes to relieve the co-reference and information ellipsis problems, which frequently occur in the dialogue context. It is also a typical generation task, which inputs the dialogue history H_n and the current utterance U_n and then infers the semantic-completed statement of the current utterance U_n. DC requires the model to focus on the connection between current utterance and dialogue history.

Slot Filling (SF) is to extract the slot types S of the entities mentioned by the user. It is a word tagging problem where the utterance is labeled in the IOB (Inside, Outside, and Beginning) format. The input is only the current utterance U_n.

Intent Detection (ID) is to recognize the intent from predefined abstracted intent expresses I. It is normally formulated as a classification problem. The input is the current utterance U_n, and the output is the possible distribution of all the intent candidates I.

Dialogue State Tracking (DST) aims to record the user's constraints, which consists of the triple set of domain-slot-value. For example, hotel-price-cheap means the user wants a cheap hotel. The input of DST at the n-th turn is the first n turns $(U_1, \ldots, U_n)$.

3 UniDU

In this section, we first introduce the unified sequence-to-sequence data format for the five DU tasks. Then we introduce the formulation of each task in detail, especially how to reformulate the intent detection, slot filling and dialogue state tracking as the generation task.

There are three components in the input of UniDU: task identification, dialogue content, and task query. The task identification represents with a special token, e.g., dialogue summary identified by "[DS]". The dialogue content means the task-dependent input, such as dialogue history for dialogue summary. The task query can be regarded as the task-specific prompt, which includes the task definition and domain-related information. There

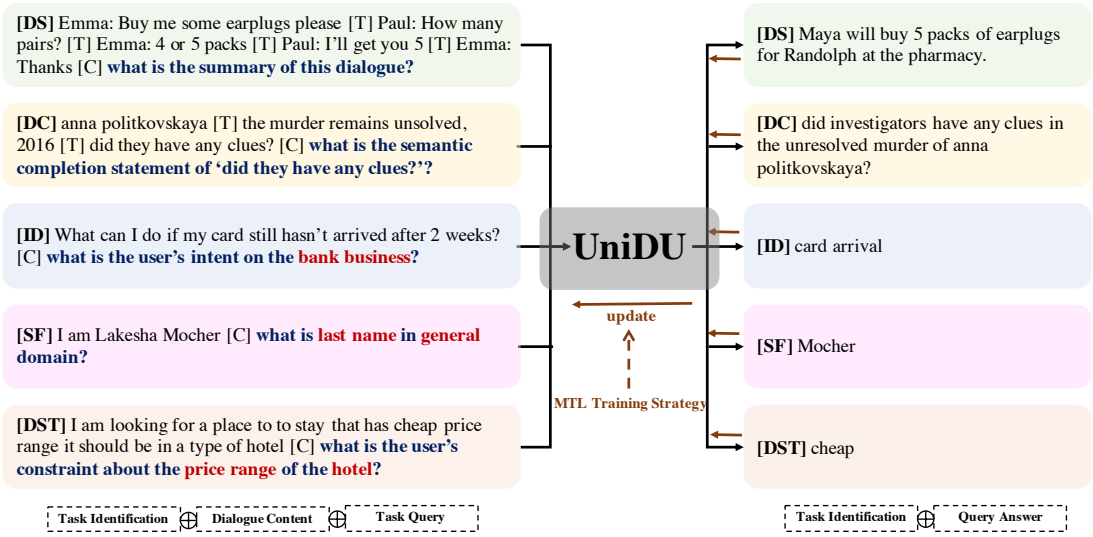


Figure 1: Overview of UniDU. Under UniDU framework, the input consists of three parts: task identification, dialogue content and task query, where ⊕ means concatenation. The output has two components: task identification and query answer. We train the UniDU model with different multitask learning strategies.

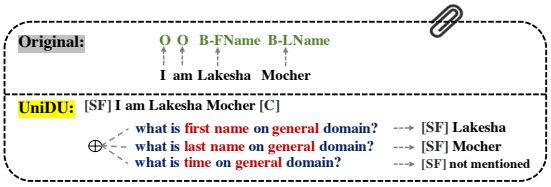
are two elements in the output of UniDU: task identification and query answer. The query answer is the understanding result of the task query given by the dialogue content. The unified input and output can be formalized as:

INPUT: [TI] dialogue content [C] task query
OUTPUT: [TI] query answer

where "[C]" is separate character and "[TI]" is task identification (replaced by "[DS]", "[DC]", "[SF]","[ID]" and "[DST]", which correspond to dialogue summary, dialogue completion, slot filling, intent detection and dialogue state tracking respectively). At inference time, the UniDU model must first predict the task identification.

Dialogue summary and dialogue completion are originally generative tasks. The dialogue contents in the input are the whole dialogue context C and multi-turn utterances H_n respectively. Since these two tasks are independent of the dialogue domain, there is no domain information in the task query. For dialogue summary, the task query is "*what is the summary of this dialogue?*". For dialogue completion, the query is "*what is the semantic completion statement of* U_n*?*", where U_n is the t-th utterance. Their understanding answers are annotated dialogue summaries and rewritten utterances in the output.

The original slot filling task demands the model to extract all the mentioned slot values and their slot types in an utterance U_n. In this paper, the UniDU model predicts the value slot by slot, which is an iterated generation process on the slot candidate list. Two different slot filling formats are shown below:

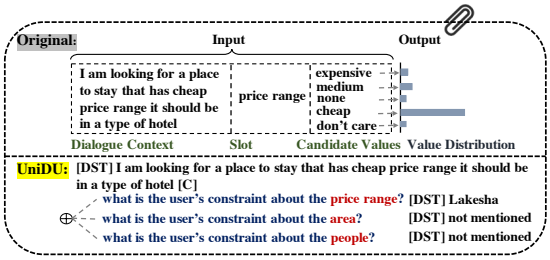


To be clear, we do not list all the candidate slots here. In general, for each sample, it can be formalized as:

INPUT: [SF] U_n [C] what is s of d?
OUTPUT: [SF] slot value

where s and d are predefined slots and domains. If s has no value in U_n, slot value will be "not mentioned". If s has multiple values, they will be separated by a comma in the slot value. When the value is "not mentioned", we call it a negative sample. Otherwise, it is a positive sample. To balance the ratio of negative and positive samples during the training process, we set the ratio to 2:1. If the number of negative samples exceeds the threshold, we randomly sample twice as many negative instances as positive ones.

For dialogue state tracking tasks, the classification methods always achieve better performance than generative methods. However, under the UniDU framework, we also formulate DST as a slot-wise value generation task similar to the slot filling task. The DST task formats are shown below:



where the output of the original DST model is the distribution of all the candidate values of the slot. The input and output of the DST task under UniDU can be formalized as follows:

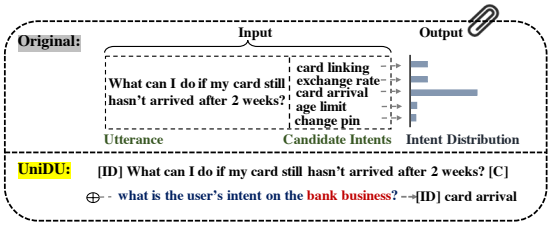
INPUT: [DST] (H_n, U_n) [C] what is the user's constraint about s of d?
OUTPUT: [DST] slot value

where (H_n, U_n) is dialogue context. If slot s of the domain d is not in the dialogue state, its value is "not mentioned", which is a negative sample. Note that different utterances are separated by the special token "[T]" in the input. During the training process, the ratio of negative and positive samples is also set below 2:1.

For the intent detection task, the original methods formulate it as the intent classification problem and output the distribution of all the candidate intents. The UniDU model directly generates the intent name of the current utterance, which can be formalized as:

INPUT: [ID] U_n [C] what is the user's intent on domain d?
OUTPUT: [ID] intent name

where domain d is normally known in advance. The specific examples of original and UniDU formats are shown below:



where we do not list all the intents. To integrate the generalization capability into the UniDU model, we also construct negative samples for the intent detection task. The intent name of the negative sample is "not defined", where the input utterances U_n are sampled from out-of-domain dialogues. The ratio of negative and positive samples is set to 2:1.

Until now, all the five dialogue understanding tasks have been formulated as the unified sequence-to-sequence generation task. The specific examples are shown in Figure 1.

4 Multitask Training Strategies

Although the five DU tasks can be formulated as a unified generative task, straightforward multitask training may not work due to the different natures of these tasks. In this section, we discuss multitask training strategies and propose a novel model-agnostic adaptive weighting strategy.

4.1 Multitask Learning Classification

The existing multitask training strategies can be classified into three categories: average sum method, manual scheduled method, and learnable weight method.

Average Sum method distributes all the samples with the same weight. In other words, the losses from different samples are directly averaged, formulated as $\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T}\mathcal{L}_t$, where T is the number of the tasks and $\mathcal{L}_t$ is the loss of the t-th task.

Manual Schedule method designs a heuristic training schedule for planning the learning process of different tasks. For example, curriculum learning (Bengio et al., 2009) is a kind of typical manual scheduled method, which first trains the easier samples and then adds the more complicated cases. The manual scheduled method can be formulated as $\mathcal{L} = \frac{1}{\sum \mathbb{I}(t)}\sum_{t=1}^{T}\mathbb{I}(t)\cdot\mathcal{L}_t$, where $\mathbb{I}(t)$ is indicator function, whose value is 0 or 1.

Learnable Weight method aims to parameterize the loss weights of different tasks. The target of the parameterized weights is to balance the effects of task instances, which prevents the model from slanting to one or several tasks and achieves global optimization. There are two classical learnable weight algorithms: homoscedastic uncertainty weighting (**HUW**) (Kendall et al., 2018) and gradient normalization (**GradNorm**) (Chen et al., 2018b). For the tasks, the loss function is formulated as $\mathcal{L} = \sum_{t=1}^{T} W_t \cdot \mathcal{L}_t$, where W_t is learnable weights and greater than 0. In the HUW algorithm, the weights update as the following loss function:

$$\mathcal{L}_{\text{HUW}} = \sum_{t=1}^{T}(\mathcal{L}_t \cdot W_t - \log(W_t)), \qquad (1)$$

where $\log(W_t)$ is to regularize weights, which is adaptive to regression tasks and classification tasks. The motivation of the GradNorm method is to slow

down the learning scale of the task that has a larger gradient magnitude and faster convergence rate.

4.2 Model-Agnostic Training Strategy

In Equation 1, the learnable weight W_t is only dependent on the corresponding task. Thus, we can regard the weight as the function of task $W_\phi(t)$, where ϕ are parameters shared among five tasks. Under the UniDU framework, five tasks share the same encoder-decoder model, which is a constant in weight function $W_\phi(t)$. The task format depends on task attributes, such as input, output, and data scale. To extract the characters of five tasks, we manually design a vector as the task feature to represent a task. Each dimension in the task feature has its physical meaning related to the model-agnostic setting. In this paper, we design 14 dimensional vector $\mathbf{f}_t$ for each task introduced in detail in Appendix B. Since the model-agnostic training strategy (MATS) formulates the weight as the task-related function and may share the function parameters among different tasks, the weights are no longer independent as in the original learnable weight method. The MATS improved from Equation 1 is formalized as:

$$\mathcal{L}_{\text{MATS}} = \sum_{t=1}^{T}(\mathcal{L}_t \cdot W_\phi(\mathbf{f}_t) - \log(W_\phi(\mathbf{f}_t))). \quad (2)$$

5 Experiments

We conduct the experiments on ten dialogue understanding corpora. Each task has two corpora. We evaluate the UniDU framework with eight different training strategies. Compared with well-designed models, our proposed UniDU can get better performance in five benchmarks. Then we deeply analyze different factors affecting the UniDU model's performance, including DU tasks, unified format, and pre-trained language models. Last but not least, we conduct few-shot experiments to validate the generalization ability of UniDU.

5.1 Corpora&Metrics

There are ten dialogue understanding corpora in total spanned five tasks: dialogue summary (DS), dialogue completion (DC), slot filling (SF), intent detection (ID), and dialogue state tracking (DST). We choose two well-studied corpora for each task: one is the evaluation corpus, and the other is the auxiliary corpus. The dataset statistics are shown in Appendix A.

Dialogue Summary: We choose SAMSUM (Gliwa et al., 2019) and DIALOGSUM (Chen et al., 2021a) datasets. The common metrics for the summary task are ROUGE scores, which measure the overlap of n-grams in the generated summary against the reference summary.

Dialogue Completion: TASK (Quan et al., 2019) and CANARD (Elgohary et al., 2019) are used. The metrics are BLEU score and exact match (EM) accuracy. BLEU measures how similar the rewritten sentences are to golden ones. Exact match means the rate of the generated totally equaled to the golden.

Intent Detection: We conduct the experiments on BANKING77 (Casanueva et al., 2020) and HWU64 (Liu et al., 2019c), where 77 and 64 means the number of predefined intents. The evaluation metric is detection accuracy (ACC.).

Slot Filling: We choose to conduct the experiments on RESTAURANTS8K (Coope et al., 2020) and SNIPS (Coucke et al., 2018). We report F_1 scores for extracting the correct span per user utterance. Note that the correct predictions on negative samples are not calculated in the F_1 score, which is comparable with traditional methods.

Dialogue State Tracking: WOZ2.0 (Wen et al., 2017) and MULTIWOZ2.2 (Zang et al., 2020) are used. The metric is joint goal accuracy (JGA), which measures the percentage of success in all dialogue turns, where a turn is considered a success if and only if all the slot values are correctly predicted. Note that we only use "hotel" domain data of MULTIWOZ2.2 in the training phase.

5.2 Eight Training Strategies

As introduced in Section 4, the multitask training strategies can be divided into three categories: average sum, manual schedule, and learnable weight. Before introducing MTL training methods, there is an intuitive baseline trained on its own data named single training (**ST**). In ST, the sequence-to-sequence models are only trained on five evaluated datasets, respectively. In average sum method, there are two types of training strategies: task transfer learning (**TT**) (Torrey and Shavlik, 2010; Ruder et al., 2019) and mixture learning (**MIX**) (Wei et al., 2021). The task transfer learning aims to enhance the performance using external data from the auxiliary corpus that has the same task setup. This is the main reason that we select two corpora for each task. The mixture learning directly mixes up all

Methods	DS(SAMSUM)		DC(TASK)		ID(BANKING77)	SF(RESTAURANTS8K)	DST(WOZ2.0)
	R-1	R-L	EM	BLEU	ACC.	F_1	JGA
Baselines	49.67* (Wu et al., 2021)	48.95*	74.2 (Chen et al., 2021b)	89.4	93.44 (Mehri et al., 2020)	96.00 (Coope et al., 2020)	91.4 (Tian et al., 2021)
Eight Training Strategies under UniDU Framework							
ST	49.74	47.10	76.4	89.0	91.49	95.76	89.8
TT	51.24	48.59	76.1	89.2	91.94	95.12	91.0
MIX	50.98	48.13	76.2	90.8	91.91	96.43	90.8
G2S	51.13	48.75	76.3	90.1	90.12	94.81	86.8
CL	51.04	48.36	77.2	89.8	92.17	96.02	90.8
GradNorm	51.33	48.69	77.4	90.4	92.07	96.69	90.5
HUW	50.31	47.69	76.2	90.4	93.14	97.43	91.9
MATS	50.53	47.97	76.6	90.6	**93.60**	**97.61**	**92.3**
Finetune	51.93	**49.01**	76.1	**91.0**	93.54	97.19	92.1

Table 1: The results on five DU tasks trained with eight learning strategies. **Finetune** means that the best model (according to underlined metric values) of each task continues to be fine-tuned on separate task corpus. * means that we run their released code with BART-base instead of BART-large to fairly compare with our model.

the training samples from ten corpora together. In these two methods, the learning weight for each sample is equally distributed. In the manual schedule method, we test two training routes according to the curriculum learning method. From the input perspective, five tasks can be divided into three classes: utterance-level input on intent detection and slot filling, turn-level input on dialogue completion, and dialogue state tracking and dialogue-level input on dialogue summary. The inputs gradually become more complex in order: utterance-level, turn-level, and dialogue-level. Thus, the intuitive method (named **CL**) trains five tasks in this order. Note that the previous data are kept in the next training phase. From the task setup perspective, dialogue summary and dialogue completion belong to domain-independent tasks. The other three tasks are domain-dependent tasks. There is another training route (**G2S**): from general tasks to domain-specific tasks. In learnable weight method, we evaluate three methods introduced in Section 4: **GradNorm**, **HUW** and our proposed **MATS**.

5.3 Experimental Setup

In this paper, we set BART-base as the backbone of the unified encoder-decoder model. The BART model is implemented with HuggingFace library (Wolf et al., 2019). We conduct all the experiments on the 2080TI GPU with 11G memory. we run every experiment for 60 epochs spent 72 hours. The batch size is 32 with the gradient accumulation strategy (updated per 8 steps). The learning rates of the unified model and learnable weights are 1e-5 and 1e-4, respectively. In the MATS method, the weight function consists of two linear layers with the ReLU activation function, whose hidden sizes are 64.

5.4 Results

In Table 1, we report the best evaluation performance on five tasks with eight training strategies. The well-designed models as baselines are introduced in Section 1. The experimental results show that different training strategies greatly affect the performance of five tasks under the UniDU framework. Our proposed MATS achieves the best or near best performance except on dialogue summary. On the atypical generation tasks (intent detection, slot filling, and dialogue state tracking), the UniDU with MATS methods can achieve promising improvement compared to well-designed models. The simple task transfer learning method (TT) can not largely increase the performance compared with single training. The mixture operation leads to consistent performance improvement on five tasks. However, compared with TT, the improvement is still limited except for dialogue completion. Compared with our proposed MATS, MIX biases convergence to more complex DU tasks (dialogue summary and dialogue completion). Two manual schedule methods (G2S and CL) do not have any distinct advantages. In learnable weight methods, GradNorm only achieves excellent performance on dialogue summary. HUW achieves performance gain on intent detection, slot filling, and dialogue state tracking. We continue fine-tuning the best

Methods	DS (R-L)	DC (BLEU)	ID (ACC.)	SF (F_1)	DST (JGA)	Overall
MIX	**48.04**	90.40	91.9	96.43	90.1	83.23
HUW	47.63	89.95	93.0	97.43	91.8	83.97
MATS	47.57	**90.43**	**93.5**	**97.46**	**91.9**	**84.16**

Table 2: The best overall performance of MIX, HUW and MATS methods.

Method	DS (R-L)	DC (BLEU)	ID (ACC.)	SF (F_1)	DST (JGA)
MATS	47.97	90.6	93.60	97.61	92.3
- DS	-	90.2$_{\blacktriangledown 0.4}$	93.20$_{\blacktriangledown 0.4}$	97.35$_{\blacktriangledown 0.26}$	92.8$_{\blacktriangle 0.5}$
- DC	47.77$_{\blacktriangledown 0.20}$	-	93.41$_{\blacktriangledown 0.19}$	97.39$_{\blacktriangledown 0.22}$	91.8$_{\blacktriangledown 0.5}$
- ID	47.81$_{\blacktriangledown 0.16}$	90.5$_{\blacktriangledown 0.1}$	-	97.45$_{\blacktriangledown 0.16}$	92.3$_{<0.0}$
- SF	47.77$_{\blacktriangledown 0.20}$	90.5$_{\blacktriangledown 0.1}$	93.60$_{<0.0}$	-	92.0$_{\blacktriangledown 0.3}$
- DST	47.85$_{\blacktriangledown 0.12}$	90.6$_{<0.0}$	93.47$_{\blacktriangledown 0.13}$	97.58$_{\blacktriangledown 0.03}$	-

Table 3: Ablation study on effects of each task corpora.

UniDU models (signed with underline) on the corresponding corpus. We find that only the dialogue summary and dialogue completion have obvious performance gain, which reflects the necessity of the UniDU framework for simpler generative tasks.

In Table 1, we report the task-specific performance of the UniDU model, whose checkpoints are selected by the task-specific metric. Table 2 shows unified performance on five tasks with MIX, HUW, and MATS methods. We evaluate the single checkpoint of UniDU model, which has the highest evaluated overall score, on the five tasks. The overall score is the average value of the five main metrics shown in Table 2. We can see that our proposed MATS gets the highest overall performance and the best performance on four DU tasks.

5.5 Analysis

In this subsection, we analyze factors to affect the performance of UniDU model including DU tasks, unified format and pre-trained language models.

5.5.1 Effects of DU Tasks

To validate the effects of the dialogue understanding tasks, we directly remove one of five DU corpora and train the UniDU model with the MATS method shown in Table 3. In general, the five DU tasks benefit each other, except that dialogue summary has negative effects on the dialogue state tracking task. We guess the general dialogue summary task summarizes a dialogue into a sentence, ignoring the domain-specific information. On the other hand, we find that the dialogue completion task has the most significant effect on the other four DU tasks. It indicates that the co-reference and information ellipsis are still the main factors to impact the dialogue understanding ability. The phenomenon can facilitate the dialogue understanding community to pay more attention to dialogue completion. For example, when pre-training a scaling dialogue model, the pre-trained tasks should be close to the dialogue completion task.

Backbone	DS (R-L)	DC (BLEU)	ID (ACC.)	SF (F_1)	DST (JGA)
Trans.-B	34.84	74.2	86.36	83.01	72.5
BART-B	**47.97**	**90.6**	**93.60**	**97.61**	**92.3**
T5-S	41.63	85.9	87.04	96.94	89.9
Trans.-L	34.10	67.4	86.46	71.65	71.0
BART-L	**48.89**	88.6	93.44	97.12	**92.6**
T5-B	**48.89**	**90.7**	**93.90**	**98.14**	**92.6**

Table 4: Ablation study on effects of different pre-trained language models with encoder-decoder architecture.

5.5.2 Effects of Unified Format

As introduced in Section 3, we formulate dialogue understanding tasks in QA format. There is an intuitive alternative: prefix format, where the task query is concatenated on the decoder side. At inference time, the decoder is directly fed with task query and then generates the answer. As shown in Figure 2, the QA format achieves a performance boost on four of five DU tasks (except for dialogue summary) compared to the prefix format.

5.5.3 Effects of PLMs

To validate the effects of the different pre-trained backbones, we initialize the encoder-decoder of UniDU model with random mechanism, BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). The **Trans.-B** and **Trans.-L** in Table 4 mean the random-initialized Transformer trained from scratch, which has the same parameters with BART-base model (**BART-B**) and BART-large model (**BART-L**). **T5-S** and **T5-B** mean T5-small and T5-base respectively. We can see that the pre-trained language models get absolute performance gain compared to random-initialized models. BART-B can get better performance than T5-S. When the parameter scale increases, T5-base achieves the best performance than other models. The results show that the large PLMs can improve the complex dialogue summary by a large margin.

Unseen Dialogue Content	UniDUMATS
[DS] USER : I'd like a taxi to **take me to ruskin gallery** [T] SYSTEM : Sure! What is your departure site? [T] USER : I will depart from saffron brasserie **at 7:15**. What is the car type and contact number so I know who and where you will pick me up? [T] SYSTEM : Booking completed! **A grey ford** will be picking you up. The contact number is 07689877132. [T] USER : That is all I needed, thank you. [C] what's the summary of this dialogue?	[DS] a grey ford will take USER to ruskin gallery at 7:15.
[DC] USER : Please reserve for me a taxi that will **pick me up at cambridge arts theatre** after 09:30 [T] SYSTEM : And where will you be going? [T] USER : I'm going to restaurant one seven. [T] SYSTEM : Your booking is complete, **a black audi will be picking you up**. [T] USER : Thank you. **I need the contact number**, as well. [C] what is the semantic completion statement of "Thank you. I need the contact number, as well."?	[DC] I need the contact number of a black audi to pick me up at cambridge arts theatre
[ID] help me **get a taxi to the cambridge museum of technology** please. [C] what is the user's intent on the taxi?	[ID] transport taxi
[SF] I need a taxi to pick me up at Ashley Hotel to **leave after 10:45**. [C] what is leaving time of taxi?	[SF] 10:45
[DST] USER : I need a taxi. I am **going to avalon** and I need to leave after 16:15 [C] what is the user's constraint about the destination of the taxi?	[DST] avalon

Table 5: Case study of the zero-shot performance of the best unified model trained with MATS method. The input dialogue contents are sampled from unseen "Taxi" domain.

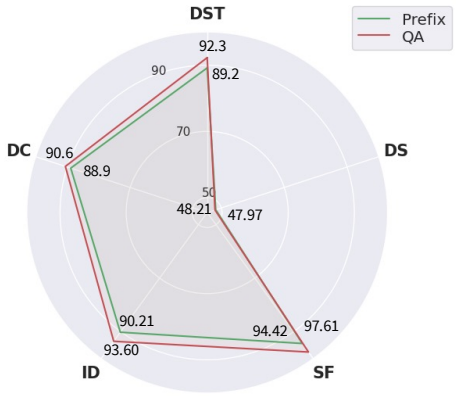


Figure 2: Ablation study of different unified understanding format.

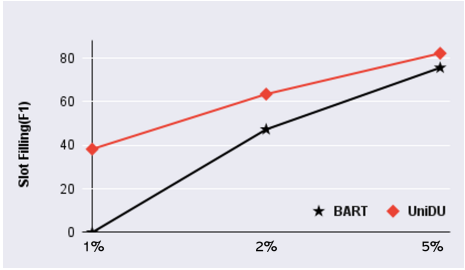


Figure 3: Few-shot learning results on slot filling fine-tuned on BART and UniDU. 1%, 2% and 5% are the percents of the training data on unseen "Bus" domain.

5.6 Generalization Ability

To further evaluate the generalization ability of the UniDU model, we first conduct few-shot learning experiments on the domain-dependent slot filling task. We test the zero-shot capability of UniDU on unseen dialogue data.

Few-shot Learning: We select the UniDU model that gets the best evaluation of overall performance on five tasks learned with the MATS method. For the slot filling task, we extend another dialogue corpus DSTC8 (Rastogi et al., 2020). We choose the "Bus" domain data in DSTC8, which is unseen in the training process of UniDU. Compared with vanilla BART, UniDU has obvious advantages, especially in the extremely resource-limited situation. When there is only 1% training data, the vanilla BART is disabled to learn, as shown in Figure 3. The few-shot experiment on the DST task is shown in Appendix C.

Zero-shot Performance: We validate UniDU model trained with MATS method on unseen "Taxi" domain dialogue data collected from MULTIWOZ2.2 corpus. UniDU model can get 18.24% accuracy on ID, 39.69% F1 score on SF and 1.6% JGA on DST.

6 Case Study

We directly validate the UniDU model trained with the MATS method on unseen "Taxi" domain dialogue data collected from MULTIWOZ2.2 corpus. As shown in Table 5, we find that the UniDU model can generate reasonable dialogue summary and completion. Note that the UniDU model did not see any task-oriented dialogue in these two tasks. For domain-specific tasks, the UniDU model can still generate accurate query answers in some cases. It indicates that our proposed generative UniDU model has excellent generalization ability, which not only can adapt to unseen dialogue and also directly generate reasonable answers on five DU tasks in the zero-shot setting.

To further explore the relations among five tasks, we plot the reduced-dimension map of the task embeddings of five tasks with the t-SNE algorithm shown in Figure 4. The task embeddings are the final decoder layer representation of the task identi-

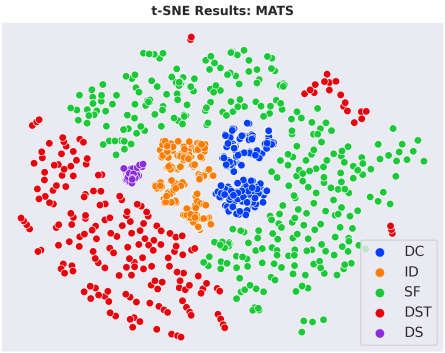


Figure 4: The reduce-dimension map of task embeddings collected from UniDU model trained by MDTS. The task embedding is the final decoder representation of the task identification token.

fication token, whose model is trained with MDTS. The dialogue data is from the above unseen "Taxi" domain to eliminate the impacts of the dialogue context. We find that the embeddings of dialogue summary, dialogue completion, and intent detection cluster together. These three tasks under the UniDU framework are more general than slot filling and dialogue state tracking, whose task queries are slot-wise. The task formats between slot filling and dialogue state tracking are close. However, the UniDU model can still have good performance to distinguish between these two tasks.

7 Related Work

Our work relates to several broad research areas, including prompting, dialogue modeling, and multitask learning. Due to the content limitation, here we describe one subarea: multitask learning in NLP applications that relate most closely to our work. Luong et al. (2016) apply a sequence-to-sequence model on three general NLP tasks and study different parameter-sharing strategies. Kumar et al. (2016); McCann et al. (2018) try to cast NLP tasks as QA over a context. The main topics in this work are how to design an efficient model to integrate the knowledge between question and context. Liu et al. (2019b) combine four natural language understanding tasks, which utilize BERT as the shared representation model. The model corresponding to each task still has a well-designed part of solving the intrinsic problem. It hampers the analysis of the interaction among the different tasks.

Recently, Wei et al. (2021) formulated the NLP tasks as the generation task by directly mixing scaling annotated data up. They only focus on zero-shot and few-shot ability on the NLP tasks and ignore the impacts of the different multitask training strategies, which can not achieve better performance on general NLP tasks compared to supervised learning methods on well-designed models. In task-oriented dialogue (TOD) modelling, Peng et al. (2020); Su et al. (2021) reformulate the pipeline TOD model as the sequential end-to-end generation problem. The end-to-end model needs to generate dialogue state, dialogue action, and response at the same time, which is not scalable when the number of tasks increases. The sequential format needs all the annotations of the same context, which is unavailable in the DU area. Most recently, PPTOD (Su et al., 2021) unifies the TOD task as multiple generation tasks, including intent detection, DST, and response generation. However, they focus on the response generation ability and ignore the effects of different tasks. In this paper, we deep dive into analyzing the effects of five DU tasks.

8 Conclusion&Future Work

In this paper, we propose a unified generative dialogue understanding framework (UniDU) to share the knowledge across five typical dialogue understanding tasks. We introduce a model-agnostic adaptive weight learning method for multitask training to alleviate the biased generation problem. Our proposed UniDU method achieves better performance compared to well-designed models on a total of five DU tasks. We further deep dive into studying the affected factors. Finally, experimental results indicate that our proposed UniDU model can also get excellent performance under few-shot and zero-shot settings. In the future, we will increase the scale of the DU corpora and integrate the unsupervised dialogue pre-training tasks. We will further examine the task-level transferability of the UniDU model.

Acknowledgements

We sincerely thank the anonymous reviewers for their valuable comments. We thank the SIGDIAL mentors Stefan Ultes and Ondrej Dusek to help us prepare our final submission. This work has been supported by the China NSFC Projects (No.62120106006 and No. 62106142), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), CCF-Tencent Open Fund and Startup Fund for Youngman Research at SJTU (SFYR at SJTU).

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. Plato: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 85–96.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Inigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. *ACL 2020*, page 38.

Lu Chen, Cheng Chang, Zhi Chen, Bowen Tan, Milica Gašić, and Kai Yu. 2018a. Policy adaptation for deep reinforcement learning-based dialogue management. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 6074–6078. IEEE.

Lu Chen, Zhi Chen, Bowen Tan, Sishan Long, Milica Gašić, and Kai Yu. 2019. Agentgraph: Toward universal dialogue management with structured deep reinforcement learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(9):1378–1391.

Lu Chen, Boer Lv, Chi Wang, Su Zhu, Bowen Tan, and Kai Yu. 2020a. Schema-guided multi-domain dialogue state tracking with graph attention neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7521–7528.

Yulong Chen, Yang Liu, and Yue Zhang. 2021a. DialogSum challenge: Summarizing real-life scenario dialogues. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 308–313, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018b. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR.

Zhi Chen, Jijia Bao, Lu Chen, Yuncong Liu, Da Ma, Bei Chen, Mengyue Wu, Su Zhu, Jian-Guang Lou, and Kai Yu. 2022. Dialogzoo: Large-scale dialog-oriented task learning. *arXiv preprint arXiv:2205.12662*.

Zhi Chen, Lu Chen, Hanqi Li, Ruisheng Cao, Da Ma, Mengyue Wu, and Kai Yu. 2021b. Decoupled dialogue modeling and semantic parsing for multi-turn text-to-sql. In *Findings of ACL 2021*.

Zhi Chen, Lu Chen, Xiaoyuan Liu, and Kai Yu. 2020b. Distributed structured actor-critic reinforcement learning for universal dialogue management. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2400–2411.

Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-convert: Few-shot span extraction for dialog with pretrained conversational representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 107–121.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *EMNLP-IJCNLP 2019*, page 70.

E Haihong, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5467–5471.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592.

Ting Han, Ximing Liu, Ryuichi Takanobu, Yixin Lian, Chongxuan Huang, Dazhen Wan, Wei Peng, and Minlie Huang. 2020. Multiwoz 2.3: A multi-domain task-oriented dialogue dataset enhanced with annotation corrections and co-reference annotation. *arXiv preprint arXiv:2010.05594*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Joo-Kyung Kim, Gokhan Tur, Asli Celikyilmaz, Bin Cao, and Ye-Yi Wang. 2016. Intent detection using semantically enriched word embeddings. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 414–419. IEEE.

Sungdong Kim, Sohee Yang, Gyuwan Kim, and Sang-Woo Lee. 2020. Efficient dialogue state tracking by selectively overwriting memory. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 567–582.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. PMLR.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. 2017. End-to-end task-completion neural dialogue systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 733–743.

Lizi Liao, Le Hong Long, Yunshan Ma, Wenqiang Lei, and Tat-Seng Chua. 2021. Dialogue state tracking with incremental reasoning. *Transactions of the Association for Computational Linguistics*, 9:557–569.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019a. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1957–1965.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019c. Benchmarking natural language understanding services for building conversational agents. In *10th International Workshop on Spoken Dialogue Systems Technology 2019*.

Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *International Conference on Learning Representations*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *arXiv preprint arXiv:2009.13570*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Building task bots at scale with transfer learning and machine teaching. *arXiv preprint arXiv:2005.05298*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021a. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197. IEEE.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021b. A survey on spoken language understanding: Recent advances and new frontiers. *arXiv preprint arXiv:2103.03095*.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. 2019. Gecor: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4547–4557.

Jun Quan, Shian Zhang, Qian Cao, Zizhong Li, and Deyi Xiong. 2020. Risawoz: A large-scale multi-domain wizard-of-oz dataset with rich semantic annotations for task-oriented dialogue modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 930–940.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Schema-guided dialogue state tracking task at dstc8. *arXiv preprint arXiv:2002.01359*.

Sebastian Ruder, Matthew Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing tutorial. *NAACL HTL 2019*, page 15.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance rewriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *arXiv preprint arXiv:2109.14739*.

Xin Tian, Liankai Huang, Yingzhan Lin, Siqi Bao, Huang He, Yunyi Yang, Hua Wu, Fan Wang, and Shuqi Sun. 2021. Amendable generation for dialogue state tracking. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 80–92.

Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Stefan Ultes, Lina M Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Inigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, et al. 2017. Pydial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Chien-Sheng Wu, Linqing Liu, Wenhao Liu, Pontus Stenetorp, and Caiming Xiong. 2021. Controllable abstractive dialogue summarization with sketch supervision. *arXiv preprint arXiv:2105.14064*.

Zihan Xu, Zhi Chen, Lu Chen, Su Zhu, and Kai Yu. 2020. Memory attention neural network for multi-domain dialogue state tracking. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 41–52. Springer.

Xiaoxue Zang, Abhinav Rastogi, Srinivas Sunkara, Raghav Gupta, Jianguo Zhang, and Jindong Chen. 2020. Multiwoz 2.2: A dialogue dataset with additional annotation corrections and state tracking baselines. *ACL 2020*, page 109.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Appendix

A Dialogue Understanding Corpora

Corpora	#Sample	$\mathbf{I}_{(\text{Token})}$	$\mathbf{I}_{(\text{Turn})}$	$\mathbf{O}_{(\text{Token})}$	Task
SAMSum	14732	104.95	11.16	20.31	DS
DialogSum	12460	140.48	9.49	22.86	DS
Task	2205	34.92	2.75	10.84	DC
Canard	31526	102.67	9.80	11.55	DC
Banking77	12081	21.64	1	3.14	ID
Hwu64	25715	17.69	1	2.05	ID
Restaurants8k	15270	14.44	1	3.38	SF
Snips	35748	15.31	1	1.77	SF
Woz2.0	7608	78.96	4.63	1.30	DST
MultiWoz2.2	35119	115.80	5.99	1.45	DST

Table 6: The ten DU corpora trained on UniDU model. $\mathbf{I}_{(\text{Token})}$ and $\mathbf{I}_{(\text{Turn})}$ mean the average length of the split tokens and the average turns of the input dialogue content. $\mathbf{O}_{(\text{Token})}$ means the average length of the split tokens of the task-specific output.

In this paper, we train our proposed unified generative model on ten dialogue understanding corpora, as shown in Table 6. For each DU tasks, we select two well-studied datasets. The first one is used to evaluate and the second one is an auxiliary corpus. The main reason to select two datasets for each task is to compare the multitask learning with the task transfer learning. We aim to know whether the knowledge sharing between different dialogue understanding data is only happening in the same DU task rather than all the DU tasks. The experimental results show that the annotated data from the other DU tasks are also important to enhance the performance, which indicates that it is an efficient way to transfer the knowledge among all the DU tasks. Note that the selected DU data are from different corpora, which means that the distribution of the input dialogue content is totally different. As shown in Table 6, the inputs and the outputs of the five DU tasks are greatly different from each other. The longest average input reaches to 140.48 and the shortest is only 14.44. The longest output is 22.86 from dialogue summary and the shortest is 1.30 from dialogue state tracking. These characters lead a big challenge to train all the dialogue understanding data in multitask learning way. The experimental results show that the intuitive mixture learning method makes UniDU model bias convergence to the more complex tasks like dialogue summary and dialogue completion. In this paper, we compare eight multitask training strategies. Our proposed MATS method can achieve the best overall performance on the five tasks under UniDU framework.

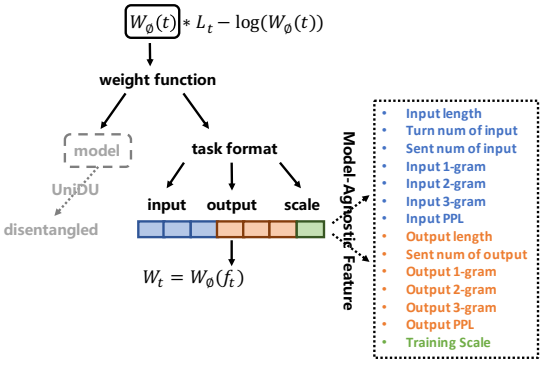


Figure 5: Overview of model-agnostic training strategy.

B Model-Agnostic Training Strategy

In traditional HWU algorithm, the learnable weight W_t is only dependent on the corresponding task. Thus, we can regard the weight function of task $W_\phi(t)$, where ϕ are parameters shared among five tasks. Generally, the task is associated with two factors: its corresponding model and task format. Under UniDU framework, five tasks share the same encoder-decoder model, which can be regarded as a constant in weight function $W_\phi(t)$. The task format dependents on model-agnostic task setting, such as input, output and data scale. To distinguish the five tasks under UniDU framework, we manually design a vector as the task feature to represent a task. Each dimension in the task feature has its physical meaning related to model-agnostic setting. In this paper, we design 14 dimensional vector $\mathbf{f}_t$, as shown in Figure 5. For input and output, we add the average length of token, the average sentence number, the n-grams and the perplexity (PPL) as the attributes of the DU tasks. Especially for input, the average turn number is also an important character. The last attribute is training scale for each task. Since the model-agnostic training strategy (MATS) formulates the weight as the task-related function and may share the function parameters among different tasks, the weights are not longer independent to each other as in original learnable weight method.

C Few-shot Learning

We select UniDU model that gets the best evaluation overall performance on five tasks learned with

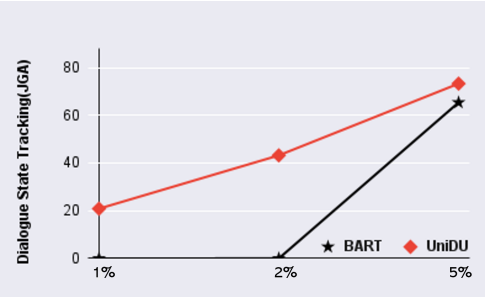


Figure 6: Few-shot learning results on DST fine-tuned on BART and UniDU. 1%, 2% and 5% are the percents of the training data on unseen "Taxi" domain.

MATS method. For dialogue state tracking, we utilize the "Train" domain data in MULTIWOZ2.2, which is unseen in MTL training phase. Compared with vanilla BART, UniDU has obvious advantages, especially on extremely resource-limited situation. When there is only 1% and 2% training data, the vanilla BART is disable to learn. UniDU model warmed up by MATS method can quickly adapt the model on the unseen domain.

Advancing Semi-Supervised Task Oriented Dialog Systems by JSA Learning of Discrete Latent Variable Models

Yucheng Cai, Hong Liu, Zhijian Ou*
Speech Processing and Machine Intelligence Lab
Tsinghua University, Beijing, China
cyc22@mails.tsinghua.edu.cn
liuhong21@mails.tsinghua.edu.cn
ozj@tsinghua.edu.cn

Yi Huang, Junlan Feng
China Mobile Research Institute
Beijing, China
{huangyi,fengjunlan}
@chinamobile.com

Abstract

Developing semi-supervised task-oriented dialog (TOD) systems by leveraging unlabeled dialog data has attracted increasing interests. For semi-supervised learning of latent state TOD models, variational learning is often used, but suffers from the annoying high-variance of the gradients propagated through discrete latent variables and the drawback of indirectly optimizing the target log-likelihood. Recently, an alternative algorithm, called joint stochastic approximation (JSA), has emerged for learning discrete latent variable models with impressive performances. In this paper, we propose to apply JSA to semi-supervised learning of the latent state TOD models, which is referred to as JSA-TOD. To our knowledge, JSA-TOD represents the first work in developing JSA based semi-supervised learning of discrete latent variable conditional models for such long sequential generation problems like in TOD systems. Extensive experiments show that JSA-TOD significantly outperforms its variational learning counterpart. Remarkably, semi-supervised JSA-TOD using 20% labels performs close to the full-supervised baseline on MultiWOZ2.1.

1 Introduction

Task-oriented dialog (TOD) systems are designed to help users to achieve their goals through multiple turns of natural language interaction. The system needs to parse user utterances, track dialog states, query a task-related database (DB), decide actions and generate responses, and to do these iteratively across turns. The information flow in a task-oriented dialog is illustrated in Figure 1.

Recent studies recast such information flow in a TOD system as conditional generation of tokens and base on pretrained language models (PLMs) such as GPT2 (Radford et al., 2019) and T5 (Raffel et al., 2020) as the model backbone. Fine-tuning a

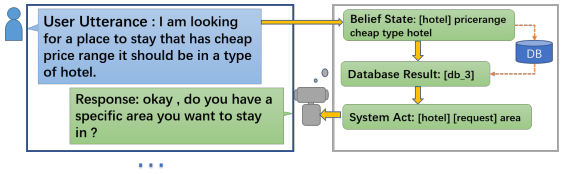


Figure 1: The information flow in a task-oriented dialog. Square brackets denote special tokens in GPT2.

PLM over annotated dialog datasets such as MultiWOZ (Budzianowski et al., 2018) via supervised learning has shown promising results (Hosseini-Asl et al., 2020; Peng et al., 2020; Yang et al., 2021; Liu et al., 2022), but requires manually labeled dialog states and system acts (if used).

Notably, there are often easily-available unlabeled dialog data such as in customer-service logs and online forums. This has motivated the development of semi-supervised leaning (SSL) for TOD systems, which aims to leverage both labeled and unlabeled dialog data. A broad class of SSL methods builds a latent variable model (LVM) of observations and labels and blends unsupervised and supervised learning. Unsupervised learning with a LVM usually maximizes the marginal log-likelihood, which is often intractable to compute. Variational learning (Kingma and Welling, 2014) introduces an auxiliary inference model and, instead, maximizes the evidence lower bound (ELBO) of the marginal log-likelihood. This approach of variational learning of LVMs has been studied for semi-supervised TOD systems such as in Jin et al. (2018); Zhang et al. (2020b); Liu et al. (2021); Li et al. (2021). Particularly, discrete latent variables are mostly used, since dialog states and system acts are often modeled as taking discrete values.

However, for variational learning of discrete latent variable models, the Monte-Carlo gradient estimator for the inference model parameter is known to have high-variance. Most previous studies use the Gumbel-Softmax trick (Jang et al., 2017) or the

*Corresponding author.

Proceedings of the SIGdial 2022 Conference, pages 456–467
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Straight-Through trick (Bengio et al., 2013) empirically, which in fact are biased estimators. Another drawback of variational learning is that it indirectly optimizes the lower bound of the target marginal log-likelihood, which leaves an uncontrolled gap between the target and the bound, depending on the expressiveness of the inference model.

Recently, an alternative algorithm, called joint stochastic approximation (JSA) (Xu and Ou, 2016; Ou and Song, 2020), has emerged for learning discrete latent variable models with impressive performances. JSA directly optimizes the marginal likelihood and completely avoids gradient propagation through discrete latent variables. In this paper, we propose to apply JSA to semi-supervised learning of the latent state TOD models, which is referred to as JSA-TOD. We develop recursive turn-level Metropolis Independence Sampling (MIS) to enable the successful application of JSA, which needs posterior sampling of the latent states from the whole dialog session. To our knowledge, JSA-TOD represents the first work in developing JSA based semi-supervised learning of discrete latent variable conditional models for such long sequential generation problems like in TOD systems.

Extensive experiments show that JSA-TOD significantly outperforms its variational learning counterpart in semi-supervised learning. Remarkably, semi-supervised JSA-TOD using 20% labels performs close to the supervised-only baseline using 100% labels on MultiWOZ2.1. The code and data are released at https://github.com/cycrab/JSA-TOD.

2 Related Work

2.1 Semi-Supervised TOD Systems

There are increasing interests in developing SSL methods for TOD systems, which aims to leverage both labeled and unlabeled data. Roughly speaking, there are two broad classes of SSL methods - the pretraining-and-finetuning approach and the latent variable modeling approach. With the development of pretrained language models such as GPT2 (Radford et al., 2019) and T5 (Raffel et al., 2020), the pretraining-and-finetuning approach based on backbones of PLMs has shown excellent performance for TOD systems (Hosseini-Asl et al., 2020; Yang et al., 2021; Lee, 2021).

Discrete latent variable models have been used for semi-supervised TOD systems (Jin et al., 2018; Zhang et al., 2020b)[1], initially based on LSTM architectures. Recently, discrete latent variable models based on PLMs have been studied in Liu et al. (2021), combining the strengths of PLMs and LVMs for semi-supervised TOD systems. However, previous studies all resort to variational methods for learning latent variable models, which suffers from the high-variance of the gradients propagated through discrete latent variables and the drawback of indirectly optimizing the target log-likelihood.

2.2 Joint Stochastic Approximation for Learning Latent Variable Models

Traditionally, variational methods minimize the "exclusive Kullback-Leibler (KL) divergence" $KL[p||q] \triangleq \int q \log \left(\frac{q}{p}\right)$, where p and q are shorthands for the true posterior (of the latent variable given the observation) and its approximation (also called the inference model) respectively, in learning a latent variable model. Recently, the JSA algorithm has been developed (Xu and Ou, 2016; Ou and Song, 2020), which proposes to minimize the "inclusive KL" $KL[p||q] \triangleq \int p \log \left(\frac{p}{q}\right)$, which has good statistical properties that makes it more appropriate for certain inference and learning problems, particularly for those using discrete latent variables. Similar idea has been studied in a concurrent and independent work (Naesseth et al., 2020). More investigations and extensions along this direction have been examined (Kim et al., 2020, 2022).

In Song and Ou (2020), JSA is applied to semi-supervised sequence-to-sequence learning, which consistently outperforms variational learning on two semantic parsing benchmark datasets. However, both generative model and inference model in (Song and Ou, 2020) are LSTM-based and much simpler than the ones in this work; its model complexity is similar to a single turn in a TOD system. Another difference is that this paper represents the first application of JSA in its conditional sequential version, since the latent state TOD model is a conditional sequential generative model.

[1]There are other previous studies of using discrete latent variable models in TOD systems, for example, Wen et al. (2017); Zhao et al. (2019); Bao et al. (2020). But most of them are mainly designed to improve response generation and diversity, instead of towards semi-supervised learning. See Zhang et al. (2020b); Liu et al. (2021) for more review of related work in latent variable models for dialogs.

3 Preliminary: Joint Stochastic Approximation (JSA)

Stochastic approximation (SA) refers to an important family of iterative stochastic optimization algorithms for stochastically solving a root finding problem, which has the form of expectations being equal to zeros (Robbins and Monro, 1951). Within the SA framework, the joint stochastic approximation (JSA) algorithm is recently developed (Xu and Ou, 2016; Ou and Song, 2020) for learning a broad class of latent variable models, particularly for learning models with discrete latent variables. Interestingly, JSA amounts to coupling an SA version of Expectation-Maximization (SAEM) (Delyon et al., 1999; Kuhn and Lavielle, 2004) with an adaptive Markov Chain Monte Carlo (MCMC) procedure. Based on JSA, the annoying difficulty of propagating gradients through discrete latent variables and the drawback of indirectly optimizing the target log-likelihood can be gracefully addressed.

Consider a latent variable generative model $p_\theta(z, x)$ for observation x and latent variable z, with parameter θ. Like in variational methods, JSA also jointly trains the target model $p_\theta(z, x)$ together with an auxiliary amortized inference model $q_\phi(z|x)$. The difference is that JSA directly maximizes w.r.t. θ the marginal log-likelihood and simultaneously minimizes w.r.t. ϕ the inclusive KL divergence $KL(p_\theta(z|x)||q_\phi(z|x))$ between the posterior and the inference model, pooled over the training dataset:

$$\begin{cases} \min_\theta \frac{1}{n}\sum_{i=1}^n \log p_\theta(x^{(i)}) \\ \min_\phi \frac{1}{n}\sum_{i=1}^n KL[p_\theta(z^{(i)}|x^{(i)})||q_\phi(z^{(i)}|x^{(i)})] \end{cases} \tag{1}$$

where the training dataset consists of n independent and identically distributed (IID) data-points $\{x^{(1)}, \cdots, x^{(n)}\}$.

The optimization problem Eq. (1) can be solved by setting the gradients to zeros and applying the SA algorithm to find the root for the resulting simultaneous equations, which has the exact form of expectations equal to zeros:

$$\begin{cases} \frac{1}{n}\sum_{i=1}^n E_{p_\theta(z^{(i)}|x^{(i)})}\left[\nabla_\theta \log p_\theta(x^{(i)}, z^{(i)})\right] = 0 \\ \frac{1}{n}\sum_{i=1}^n E_{p_\theta(z^{(i)}|x^{(i)})}\left[\nabla_\phi \log q_\phi(z^{(i)} \mid x^{(i)})\right] = 0 \end{cases} \tag{2}$$

The resulting JSA algorithm, as summarized in Algorithm 1, iterates Monte Carlo sampling and parameter updating. In each iteration, we draw a training observation $x^{(\kappa)}$ and then sample $z^{(\kappa)}$ through Metropolis Independence Sampling (MIS), with $p_\theta(z^{(\kappa)}|x^{(\kappa)})$ as the target distribution and $q_\phi(z|x^{(\kappa)})$ as the proposal:

1) Propose $z \sim q_\phi(z|x^{(\kappa)})$;

2) Accept $z^{(\kappa)} = z$ with probability

$$min\left\{1, \frac{w(z)}{w(\bar{z}^{(\kappa)})}\right\}$$

where $w(z) = \frac{p_\theta(z|x^{(\kappa)})}{q_\phi(z|x^{(\kappa)})} \propto \frac{p_\theta(z,x^{(\kappa)})}{q_\phi(z|x^{(\kappa)})}$ is the usual importance ratio between the target and the proposal distribution and $\bar{z}^{(\kappa)}$ denotes the cached latent state for observation $x^{(\kappa)}$.

Algorithm 1 The JSA algorithm

repeat

Monte Carlo sampling:

Draw κ over $1, \cdots, n$, pick the data-point $x^{(\kappa)}$ along with the cached $\bar{z}^{(\kappa)}$, and use MIS to draw $z^{(\kappa)}$;

Parameter updating:

Update θ by ascending: $\nabla_\theta \log p_\theta(z^{(\kappa)}, x^{(\kappa)})$;

Update ϕ by ascending: $\nabla_\phi \log q_\phi(z^{(\kappa)}|x^{(\kappa)})$;

until convergence

The JSA algorithm can be intuitively understood as a stochastic extension of the well-known EM algorithm (Dempster et al., 1977). Since the latent variable $z^{(\kappa)}$ is unknown for data-point $x^{(\kappa)}$, the Monte Carlo sampling step in JSA fills the missing value for $z^{(\kappa)}$ through sampling $p_\theta(z^{(\kappa)}|x^{(\kappa)})$, which is analogous to the E-step in EM. Then in the parameter updating step, $z^{(\kappa)}$ is treated as if being known, and used to optimize over θ and ϕ by performing gradient ascent using $\nabla_\theta \log p_\theta(z^{(\kappa)}, x^{(\kappa)})$ and $\nabla_\phi \log q_\phi(z^{(\kappa)}|x^{(\kappa)})$ respectively. This is analogous to the M-step in EM, but with the proposal q_ϕ being adapted as well. In summary, we could refer to the underlying mechanism of JSA as Propose, Accept/Reject, and Optimize (or, for short, the PARO mechanism), which establishes JSA as a simple, solid and effective approach to learning discrete latent variable models.

4 Method

4.1 Definition of Discrete Latent Variables in TOD systems

In a TOD system, let u_t denote the user utterance, b_t the dialog state, db_t the DB result, a_t the system

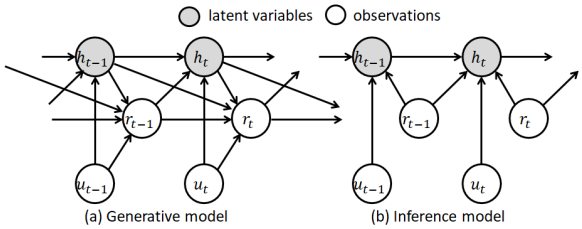


Figure 2: The probabilistic graphical model of Markov latent state generative model (a) and inference model (b) for TOD systems. u_t and r_t are user utterance and system response respectively. The latent variables $h_t = \{b_t, a_t\}$ are the concatenation of dialog state and system act, which specifically are represented by token sequences in our experiments.

act and r_t the delexicalized response, respectively, at turn t. In this work, all these variables are converted to token sequences, like in DAMD (Zhang et al., 2020a). As shown in Figure 1, the workflow for a TOD system is, for each dialog turn t, to generate b_t, a_t and r_t, given u_t and dialog history $u_1, r_1, \cdots, u_{t-1}, r_{t-1}$. The database result db_t is deterministically obtained by querying database using the predicted b_t, and thus could be omitted in the following probabilistic modeling of a TOD system for simplicity.

Let $h_t = \{b_t, a_t\}$ denote the concatenation of dialog state and system act. Specifically, dialog state b_t and system act a_t are represented by sequences of labels, for example, $[train]$ day $monday$ $[hotel]$ $pricerange$ $cheap$ and $[train]$ $[inform]$ $choice$ $departure$ $[request]$ $destination$, respectively. Notably, h_t's are observed in labeled dialogs, but they become latent variables in unlabeled dialogs in training and need to be generated in testing. With this definition of h_t's, latent variable models can be developed for TOD systems, which will be described shortly in the next subsection.

Remarkably, the above definition of latent variables as sequences of labels in this paper is similar to Zhang et al. (2020b); Liu et al. (2021). An important feature of such latent variables is that they are sensible and interpretable, which correspond to meaningful annotations according to the task knowledge. It is only in unlabeled dialogs that they become unobservable. This is different in nature from some other previous studies of using latent variables in TOD models (Wen et al., 2017; Zhao et al., 2019; Bao et al., 2020), where the latent variables are just assumed to be K-way categorical variables and learned in a purely data driven way.

4.2 A Probabilistic Latent State TOD Model

With the above introduction of latent variables and motivated by recent studies (Zhang et al., 2020b; Liu et al., 2022), the workflow of a TOD system could be described by a conditional sequential generative model with latent variables h_t's as follows for T turns, with parameter θ:

$$p_\theta(h_{1:T}, r_{1:T}|u_{1:T}) = \prod_{t=1}^{T} p_\theta(h_t, r_t|u_1, h_1, r_1, \cdots, u_{t-1}, h_{t-1}, r_{t-1}, u_t) \quad (3)$$

$$= \prod_{t=1}^{T} p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t) \text{(by Markov assumption)} \quad (4)$$

Here Eq. (3) and Eq. (4) could be collectively referred to as *latent state TOD models*, being non-Markov and Markov respectively. Eq. (3) represents non-Markov latent state models, which, with different further instantiations, are used in recent PLM-based TOD systems such as in Hosseini-Asl et al. (2020); Yang et al. (2021); Liu et al. (2021). In contrast, Eq. (4) makes the Markov assumption that the conditional generation of current h_t and r_t (when given u_t) depends on the dialog history only through h_{t-1} and r_{t-1} at the immediately preceding turn. Markov models have been employed in LSTM-based TOD systems such as in Lei et al. (2018); Zhang et al. (2020a); Zhang et al. (2020b). A recent study in Liu et al. (2022) revisits Markovian generative architectures (MGAs) for PLM backbones (GPT2 and T5) and shows their efficiency advantages in memory, computation and learning over non-Markov models.

4.3 Model Instantiation and Supervised Learning

In our experiments, we mainly consider MGA based latent state TOD systems (Liu et al., 2022), which are illustrated in Figure 2 as directed probabilistic graphical models. The conditional distribution $p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t)$ is instantiated as

$$p_\theta(b_t, a_t, r_t|b_{t-1}, r_{t-1}, u_t) \quad (5)$$

which is realized based on a GPT2 backbone in our experiments. The concatenation $b_{t-1} \oplus r_{t-1} \oplus u_t$ is used as the conditioning input, and the output $b_t \oplus a_t \oplus r_t$ is generated token-by-token in an autoregressive manner, where $\oplus$ denotes the concatenation of token sequences.

In order to perform unsupervised learning over unlabeled dialogs (to be detailed below), we introduce an inference model $q_\phi(h_{1:T}|u_{1:T}, r_{1:T})$ as follows to approximate the true posterior $p_\theta(h_{1:T}|u_{1:T}, r_{1:T})$:

$$q_\phi(h_{1:T}|u_{1:T}, r_{1:T}) = \prod_{t=1}^{T} q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t) \tag{6}$$

The conditional $q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t)$ is instantiated as

$$q_\phi(b_t, a_t|b_{t-1}, r_{t-1}, u_t, r_t) \tag{7}$$

which is realized based on a GPT2 backbone as well in our experiments.

When labeled dialog data are available, the supervised training of the latent state generative model p_θ in and inference model q_ϕ can be decomposed into turn-level teacher-forcing, since the latent states h_t's are known (labeled) for all turns and the model likelihoods decomposes over turns, as shown in Eq. (4) and Eq. (6).

4.4 JSA Learning over Unlabeled Dialogs

Suppose that we have n unlabeled dialogs $\left\{(u^{(i)}_{1:T_i}, r^{(i)}_{1:T_i})|i = 1, \cdots, n\right\}$, i.e., user utterances and system responses are available for each dialog, but without any annotations of the latent states. The training instances are indexed by the superscripts, and T_i denote the number of turns in the i-th training instance. The unsupervised learning of the latent state TOD model over such unlabeled data can be realized by applying the JSA algorithm, and more specifically its conditional version, to maximize the conditional marginal log-likelihood $\log p_\theta(r_{1:T}|u_{1:T})$.

The objective functions in JSA learning can be developed as follows, similar to Eq. (1):

$$\begin{cases} \min\limits_{\theta} \dfrac{1}{n}\sum\limits_{i=1}^{n} \log p_\theta(r^{(i)}_{1:T_i}|u^{(i)}_{1:T_i}) \\ \min\limits_{\phi} \dfrac{1}{n}\sum\limits_{i=1}^{n} KL[p_\theta(h^{(i)}_{1:T_i}|u^{(i)}_{1:T_i}, r^{(i)}_{1:T_i}) \\ \qquad\qquad ||q_\phi(h^{(i)}_{1:T_i}|u^{(i)}_{1:T_i}, r^{(i)}_{1:T_i})] \end{cases}$$

where we substitute observation x by $r_{1:T}$ and latent variable z by $h_{1:T}$, all conditioned on $u_{1:T}$.

Basically, JSA learning iterates Monte Carlo sampling and parameter updating, as outlined in Algorithm 1. In each iteration, we randomly pick a training instance $(u_{1:T}, r_{1:T})$ along with the cached latent state $\bar{h}_{1:T}$, and we need to draw a posterior sample $h_{1:T} \sim p_\theta(h_{1:T}|u_{1:T}, r_{1:T})$. Remarkably, it can be shown in Appendix A that for the posterior $p_\theta(h_{1:t}|u_{1:t}, r_{1:t})$ induced from the joint distribution in Eq. (4), the following recursion holds:

$$\begin{aligned} &p_\theta(h_{1:t}|u_{1:t}, r_{1:t}) \\ \propto &p_\theta(h_{1:t-1}|u_{1:t-1}, r_{1:t-1}) p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t) \end{aligned} \tag{8}$$

Based on such recursion, we can develop a recursive turn-level MIS sampler, as shown in Algorithm 2, which recursively runs MIS sampler turn-by-turn and finally obtains a valid posterior sample for the whole dialog session, i.e., $h_{1:T} \sim p_\theta(h_{1:T}|u_{1:T}, r_{1:T})$.

Suppose that we have obtained a sample for the previous t-1 turns, i.e., $h_{1:t-1} \sim p_\theta(h_{1:t-1}|u_{1:t-1}, r_{1:t-1})$. Then, we perform MIS sampling as follows, with $p_\theta(h_{1:t}|u_{1:t}, r_{1:t})$ as the target distribution and

$$p_\theta(h_{1:t-1}|u_{1:t-1}, r_{1:t-1}) q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t) \tag{9}$$

as the proposal distribution:

1) Propose $h'_t \sim q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t)$. Thus, $(h_{1:t-1}, h'_t)$ is a valid sample proposed from the proposal distribution as shown in Eq. (9);

2) Simulate $\xi \sim Uniform[0, 1]$ and let

$$h_t = \begin{cases} h'_t, & \text{if } \xi \leq min\left\{1, \dfrac{w(h_{1:t-1}, h'_t)}{w(h_{1:t-1}, \bar{h}_t)}\right\} \\ \bar{h}_t, & \text{otherwise} \end{cases} \tag{10}$$

where the importance ratio between the target and the proposal distribution

$$\begin{aligned} &w(h_{1:t-1}, h_t) \\ =&\frac{p_\theta(h_{1:t}|u_{1:t}, r_{1:t})}{p_\theta(h_{1:t-1}|u_{1:t-1}, r_{1:t-1}) q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t)} \\ \propto&\frac{p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t)}{q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t)} \end{aligned} \tag{11}$$

After we obtain the sampled latent state $h_{1:T}$ from Algorithm 2, we perform parameter updating, as outlined in Algorithm 1. The sampled latent state $h_{1:T}$ is treated as if being known, and we can calculate the gradients of $\log p_\theta(h_{1:T}, r_{1:T}|u_{1:T})$ and $\log q_\phi(h_{1:T}|u_{1:T}, r_{1:T})$ w.r.t. θ and ϕ according to Eq. (4) and Eq. (6) respectively, as if we calculate gradients in supervised training. Thanks

Algorithm 2 Recursive turn-level MIS sampler

Input: A T-turn dialog $(u_{1:T}, r_{1:T})$ with cached latent state $\bar{h}_{1:T}$, generative model p_θ in Eq. (4), inference model q_ϕ in Eq. (6).

for $t = 1$ to T **do**

Propose $h'_t \sim q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t)$;

Accept h'_t as h_t, or reject h'_t and keep $\bar{h}_t$ as h_t, according to Eq. (10);

end for

Return: $h_{1:T}$, as a posterior sample from $p_\theta(h_{1:T}|u_{1:T}, r_{1:T})$ and used as the new cached latent state.

to the PARO mechanism of JSA, we have such a conceptual simplicity for learning seemly complex conditional sequential latent variable model.

4.5 Semi-Supervised TOD Systems via JSA

Now we have introduced the method of building latent state TOD systems (Eq. (4) and Eq. (6)) with JSA learning (Algorithm 1 and Algorithm 2), which is referred to JSA-TOD. Semi-supervised learning over a mix of labeled and unlabeled data could be readily realized in JSA-TOD by maximizing the weighted sum of $\log p_\theta(h_{1:T}, r_{1:T}|u_{1:T})$ (the conditional joint log-likelihood) over labeled data and $\log p_\theta(r_{1:T}|u_{1:T})$ (the conditional marginal log-likelihood) over unlabeled data.

The semi-supervised training procedure of JSA-TOD is summarized in Algorithm 3. Specifically, we first conduct supervised pre-training of both the generative model p_θ and the inference model q_ϕ on labeled data in JSA-TOD. Then we randomly draw supervised and unsupervised mini-batches from labeled and unlabeled data. For labeled dialogs, the latent states h_t's are given (labeled). For unlabeled dialogs, we apply the recursive turn-level MIS sampler (Algorithm 2) to sample the latent states h_t's[2] and treat them as if being given. The gradients calculation and parameter updating are then the same for labeled and unlabeled dialogs. Such simplicity in application is an appealing property of JSA, apart from its superior performance, as we show later in experiments.

[2]Sampling is empirically implemented via greedy decoding in our experiments.

Algorithm 3 Semi-supervised training in JSA-TOD

Input: A mix of labeled and unlabeled dialogs.

Run supervised pre-training of θ and ϕ on labeled dialogs;

repeat

Draw a dialog $(u_{1:T}, r_{1:T})$;

if $(u_{1:T}, r_{1:T})$ is not labeled **then**

Generate $h_{1:T}$ by applying the recursive turn-level MIS sampler (Algorithm 2);

end if

$J_\theta = 0, J_\phi = 0$;

for $i = 1, \cdots, T$ **do**

$J_\theta += \log p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t)$;

$J_\phi += \log q_\phi(h_t|h_{t-1}, r_{t-1}, u_t, r_t)$;

end for

Update θ by ascending: $\nabla_\theta J_\theta$;

Update ϕ by ascending: $\nabla_\phi J_\phi$;

until convergence

return θ and ϕ

5 Experiments

5.1 Experiment settings

Experiments are conducted on MultiWOZ2.1 (Eric et al., 2020), which is an English multi-domain dialogue dataset of human-human conversations, collected in a Wizard-of-Oz setup with 10.4k dialogs over 7 domains. The dataset was officially randomly split into a train, test and development set, which consist of 8434, 1000 and 1000 dialog samples, respectively. The dialogs in the dataset are all labeled with dialog states and system acts at every turn. Compared to MultiWOZ2.0, MultiWOZ2.1 removed some noisy state values. Following (Liu et al., 2022), some inappropriate state values and spelling errors are further corrected. Dialog responses are delexicalized to reduce surface language variability. We implement domain-adaptive pre-processing like in DAMD (Zhang et al., 2020a). More implementation details for our experiments are available in Appendix B.

For evaluation in MultiWOZ2.1, there are mainly four metrics for corpus based evaluation (Mehri et al., 2019). *Inform Rate* measures how often the entities provided by the system are correct; *Success Rate* refers to how often the system is able to answer all the requested attributes by user; *BLEU Score* is used to measure the fluency of the generated responses by analyzing the amount of n-gram overlap between the real responses and the gener-

Table 1: Main results on MultiWOZ2.1 for comparison between supervised-only, variational, and JSA methods. Results are reported as the mean and standard deviation from 3 runs with different random seeds.

Proportion	Method	Inform	Success	BLEU	Combined
100%	Sup-only	84.50±0.29	72.77±0.50	18.96±0.36	97.59±0.54
20%	Sup-only	75.70±1.87	61.07±2.21	16.66±0.29	85.05±2.16
	Variational	81.83±1.55	67.67±0.50	17.88±0.95	92.63±0.30
	JSA	83.25± 0.65	71.40±1.20	18.72±0.07	**96.04±0.85**
15%	Sup-only	80.00±0.43	55.57±1.22	16.20±0.24	79.00±1.51
	Variational	80.85±0.65	67.67±0.88	17.68±0.29	91.86±0.19
	JSA	83.23±0.53	71.97±1.27	18.59±0.19	**95.47±0.73**
10%	Sup-only	67.57±0.39	50.03±1.09	15.31±0.28	74.11±0.59
	Variational	80.67±1.33	66.97±1.23	17.34±0.56	91.15±1.80
	JSA	81.97±0.79	70.40±0.99	18.09±0.38	**94.27±1.23**
5%	Sup-only	49.73±2.45	33.67±1.79	14.07±0.13	55.77±1.94
	Variational	74.17±0.53	59.93±0.34	16.06±0.69	83.11±1.04
	JSA	72.37±1.19	59.73±0.92	18.57±0.54	**84.62±0.43**

ated responses; *Combined Score* is computed as (BLEU + 0.5 * (Inform + Success)). To avoid any inconsistencies in evaluation, we use the evaluation scripts in Nekvinda and Dušek (2021), which are now also the standardized scripts adopted in the MultiWOZ website.

5.2 Main Results

In the semi-supervised experiments, we randomly draw some proportions (5%, 10%, 15% and 20%) of the labeled dialogs from the MultiWOZ2.1 training set, with the rest dialogs in the training set treated as unlabeled, and conduct semi-supervised experiments. Specifically, the number of dialogs kept as labeled under these proportions are 1686, 1265, 843, and 421, respectively, while the rest dialogs are used as unlabeled (i.e., the original labels of dialog states and system acts at all turns are removed for those dialogs in the training set).

The main results are shown in Table 1. For model instantiations, we use the GPT2 based Markov generative model and inference model, as introduced in Liu et al. (2022). It has been shown in Liu et al. (2022) that using Markovian generative architecture achieves better results than non-Markov models in the low-resource setting for both supervised-only learning and semi-supervised variational learning, which makes it a strong baseline to compare. We first train the generative model and inference model on only the labeled data, which is referred to as "Supervised-only" (Sup-only for short). Then, we perform semi-supervised training on both labeled and unlabeled data. Using the variational method in (Liu et al., 2021, 2022), we get the baseline results of "Variational", where the Straight-Through trick is used to propagate the gradients through discrete latent variables. Using the JSA method proposed in Algorithm 3, we get the results of "JSA". We conduct the experiments with 3 random seeds and report the mean and standard deviation in Table 1.

From Table 1, we can see that both the Variational and the JSA methods outperform the Supervised-only method substantially across all label proportions. This clearly demonstrate the advantage of semi-supervised TOD systems. Remarkably, semi-supervised JSA-TOD using 20% labels performs close to the supervised-only baseline using 100% labels on MultiWOZ2.1.

When comparing the two semi-supervised methods, JSA performs better than Variational significantly across almost all label proportions in terms of all four metrics (Inform Rate, Success Rate, BLEU, and Combined Score). Exceptionally, in the case of 5% labels, the Inform Rate of JSA is worse than that of Variational, the Success Rates are close; Nevertheless, the Combined Score of JSA is significantly better. Presumably, this is because we use the Combined Scores to monitor the training, apply early stopping and select the model with the best Combined Score on the validation set. Such model selection put more priority on the overall performance in terms of Combined Scores.

Further, the results in Table 1 are pooled over all label proportions and all random seeds, and the matched-pairs significance tests (Gillick and Cox, 1989) are conducted to compare JSA and Variational for Inform, Success and BLEU respectively. The p-values are 9.27×10^{-2}, 2.576×10^{-14}, and 2.939×10^{-39} respectively, which show that JSA significantly outperforms Variational.

5.3 Ablation study and analysis

Notably, the JSA and the variational methods in our experiments use the same model instantiations for p_θ and q_ϕ. The only difference lies in the learning methods they used. In the following, we provide ablation study to illustrate the superiority of JSA over variational in learning latent state TOD models.

The importance of Metropolis Independence Sampling in JSA. In JSA, we need to use Monte Carlo sampling, particularly the Metropolis Independence Sampling (MIS) to decide whether or not to update the cached latent states h_t's. A naive method is to always accept the labels proposed by the inference model, which is somewhat like self-training (Rosenberg et al., 2005). Another simple method is to run session-level MIS, with the whole

Table 2: Ablation results for using different methods to update latent states h_t's (label proportion: 10%, random seed:11)

Method	Inform	Success	BLEU	Combined
Without MIS	71.10	59.80	18.71	84.16
Session-level MIS	78.70	65.50	17.20	89.30
Recursive turn-level MIS	82.80	71.80	18.56	**95.86**

Eq. (4) as the target distribution and the whole Eq. (6) as the proposal distribution. The whole $h_{1:T}$ is proposed via ancestral sampling and then get accepted/rejected. The results from one run with each different method are shown in Table 2. Both MIS based methods significantly improves the results, which clearly reveals the importance of using MIS in JSA. By the accept/reject mechanism, we accept latent states which have higher importance ratios and exploit them to update both generative model and inference model, and at the same time, we also explore the state space by randomly accepting latent states which have lower importance ratios. Exploitation and exploration of the latent states seems to be well balanced in JSA, which may explain its good performance. Our proposed recursive turn-level MIS in Algorithm 2 clearly outperforms the session-level MIS, since it samples in a much lower dimensional state space.

The latent state prediction performance of inference model. In both variational and JSA learning, the inference model q_ϕ, which is introduced to approximate the true posterior, plays an important role. The latent states inferred from q_ϕ are used, either directly as in variational learning or after accepted/rejected as in JSA learning, to optimize the generative model p_θ. We measure the quality of the latent states predicted from q_ϕ by label precision/recall/F1, compared to oracle b_t and a_t (excluding db_t) . We compare different q_ϕ obtained from three training methods - Supervised-only, Variational, and JSA. Note that at the end of running any particular training method, we obtain not only p_θ but also q_ϕ. The performances of p_θ over the test set are shown in Table 1. The testing performances of q_ϕ obtained from one run of each different method are shown in Table 3. It can be seen that semi-supervised variational learning does not improve the prediction ability of the inference model, compared to the inference model trained only on the labeled data. In contrast, the prediction performance of the inference model is increased significantly by semi-supervised JSA learning, which is in line with the superior results of JSA's generative model as shown in Table 1.

Table 3: Performance comparison of inference models from different methods, measured by latent state prediction precision/recall/F1 over the test set.

Label Proportion	Method	Precision	Recall	F1
20%	Supervised-only	0.928	0.908	0.918
	Variational	0.924	0.900	0.912
	JSA	**0.936**	**0.925**	**0.931**
15%	Supervised-only	0.924	0.891	0.907
	Variational	0.917	0.872	0.894
	JSA	**0.934**	**0.910**	**0.922**
10%	Supervised-only	0.916	0.868	0.891
	Variational	0.887	0.880	0.883
	JSA	**0.930**	**0.898**	**0.914**
5%	Supervised-only	0.894	0.804	0.847
	Variational	0.891	0.838	0.864
	JSA	**0.904**	**0.863**	**0.883**

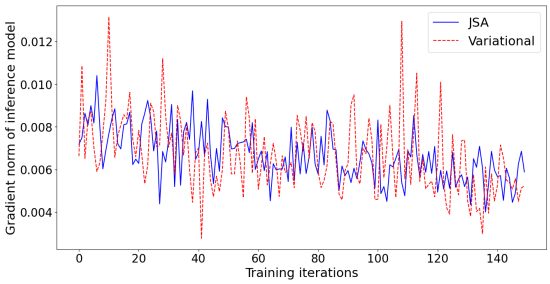


Figure 3: Comparison of the gradient norms from the inference models during training, using variational and JSA methods respectively (label proportion: 10%).

The variance of the gradients from inference model. The gradients for the inference model parameters in variational learning are known to have high-variance, due to gradient propagation through discrete latent variables, while JSA avoids such drawback. From one run of semi-supervised learning under 10% labels, we plot the gradient norms for the inference model parameters, from using the variational and the JSA methods respectively, which are shown in Figure 3. For clarity of comparison, we normalize the sum of the gradient norms over all iterations to be one. It can be clearly seen from Figure 3 that the gradients during variational training are more noisy than those in JSA tranining. Specifically, the variances of the time-series of the gradient norms in Figure 3 are 3.097×10^{-6} and 1.527×10^{-6} for the variational and the JSA methods respectively.

Pretrained models on external dialog corpora can be further improved by JSA learning for semi-supervised TOD systems. Pretraining and LVM based learning are two broad classes of semi-supervised methods. Recently, pretraining on external dialog copora has also shown to be promising

Table 4: MultiWOZ2.1 testing results for different methods (label proportion: 3%). "+Pretrained model" means that the method is initialized from the pretrained models over four external dialog corpra.

Method	Inform	Success	BLEU	Combined
Supervised-only	38.70	25.60	16.42	48.57
+ Pretrained model	57.70	39.30	14.15	62.65
Semi-supervised JSA	55.50	44.80	16.93	67.08
+ Pretrained model	**73.00**	**58.60**	**18.61**	**84.41**

for building TOD systems for low-resource scenarios (Peng et al., 2020; Su et al., 2021). In this section, we show that JSA learning can be used to further improve over such pretrained models. We use four dialog corpora - MSRE2E (Li et al., 2018), Frames (El Asri et al., 2017), TaskMaster (Byrne et al., 2019) and SchemaGuided (Rastogi et al., 2020), which consist of 16545 dialogs with human annotations on belief states and dialog acts, and we follow the preprocessing in (Su et al., 2021). Generative model and inference model, initialized from GPT2, are pretrained separately on those four corpora, the same as that in supervised-pretraining. Then, we conduct semi-supervised training with only 3% labels in MultiWOZ2.1 (i.e., 240 labeled dialogs with the rest being unlabeled). The results in Table 4 show that semi-supervisded JSA on top of pretrained models obtains the best result. This is an encouraging result from using 3% labels, which is close to the naive supervised-only method using 20% labels as shown in Table 1.

6 Conclusion and Future Work

This paper represents a progress towards building semi-supervised TOD systems by learning latent state TOD models. Traditionally, variational learning is often used; notably, the recently emerged JSA method has been shown to surpass variational learning, particularly in learning of discrete latent variable models. This paper represents the first application of JSA in its conditional sequential version, particularly for such long sequential generational problems like in TOD systems. Extensive experiments clearly show the superiority of JSA-TOD over its variational learning counterpart, not only in benchmark metrics for semi-supervised TOD systems but also from the latent state prediction performances and the variances of the gradients of the inference model. Since discrete latent variable models are widely used in many natural language procession tasks, we hope the results presented in this paper will encourage the community to further explore the applications of JSA and improve upon current approaches.

7 Acknowledgements

This research was funded by Joint Institute of Tsinghua University - China Mobile Communications Group Co., Ltd., Beijing, China.

References

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525.

Bernard Delyon, Marc Lavielle, and Eric Moulines. 1999. Convergence of a stochastic approximation version of the EM algorithm. *Annals of statistics*, pages 94–128.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39.

Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. 2017. Frames: A corpus for adding memory to goal-oriented dialogue systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Kumar Goyal, Peter Ku, and Dilek Hakkani-Tür. 2020. Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *LREC*.

Laurence Gillick and Stephen J Cox. 1989. Some statistical issues in the comparison of speech recognition algorithms. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 532–535. IEEE.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*.

Xisen Jin, Wenqiang Lei, Zhaochun Ren, Hongshen Chen, Shangsong Liang, Yihong Zhao, and Dawei Yin. 2018. Explicit state tracking with semi-supervision for neural dialogue generation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*.

Hyunsu Kim, Juho Lee, and Hongseok Yang. 2020. Adaptive strategy for resetting a non-stationary markov chain during learning via joint stochastic approximation. In *Third Symposium on Advances in Approximate Bayesian Inference*.

Kyurae Kim, Jisu Oh, and Hongseok Kim. 2022. Markov-chain monte carlo score estimators for variational inference with score climbing.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations (ICLR)*.

Estelle Kuhn and Marc Lavielle. 2004. Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM: Probability and Statistics*, 8:115–131.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *56th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dongdong Li, Zhaochun Ren, Pengjie Ren, Zhumin Chen, Miao Fan, Jun Ma, and Maarten de Rijke. 2021. Semi-supervised variational reasoning for medical dialogue generation. In *ACM SIGIR*, pages 544–554.

Xiujun Li, Yu Wang, Siqi Sun, Sarah Panda, Jingjing Liu, and Jianfeng Gao. 2018. Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv preprint arXiv:1807.11125*.

Hong Liu, Yucheng Cai, Zhenru Lin, Zhijian Ou, Yi Huang, and Junlan Feng. 2021. Variational latent-state GPT for semi-supervised task-oriented dialog systems. *arXiv preprint arXiv:2109.04314*.

Hong Liu, Yucheng Cai, Zhijian Ou, Yi Huang, and Junlan Feng. 2022. Revisiting Markovian generative architectures for efficient task-oriented dialog systems. *arXiv preprint arXiv:2204.06452*.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*.

Christian Naesseth, Fredrik Lindsten, and David Blei. 2020. Markovian score climbing: Variational inference with KL(p||q). *Advances in Neural Information Processing Systems*, 33:15499–15510.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of bleu, flavours of success: The case of multiwoz. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 34–46.

Zhijian Ou and Yunfu Song. 2020. Joint stochastic approximation and its application to learning discrete latent variable models. In *Conference on Uncertainty in Artificial Intelligence*, pages 929–938. PMLR.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics (TACL), 2021*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8689–8696.

Herbert Robbins and Sutton Monro. 1951. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models. *WACV/MOTION*.

Yunfu Song and Zhijian Ou. 2020. Semi-supervised seq2seq joint-stochastic-approximation autoencoders with applications to semantic parsing. *IEEE Signal Processing Letters*, 27:31–35.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. *CoRR*, abs/2109.14739.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve J. Young. 2017. Latent intention dialogue models. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Haotian Xu and Zhijian Ou. 2016. Joint stochastic approximation learning of Helmholtz machines. In *ICLR Workshop Track*.

Yunyi Yang, Yunhao Li, and Xiaojun Quan. 2021. UBAR: Towards fully end-to-end task-oriented dialog system with GPT-2. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Yichi Zhang, Zhijian Ou, Min Hu, and Junlan Feng. 2020b. A probabilistic end-to-end task-oriented dialog model with latent belief states towards semi-supervised learning. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020a. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*.

A Proof of Eq. (8)

First, we have

$$
\begin{aligned}
&p_\theta(h_{1:t}, r_{1:t}|u_{1:t}) \\
=&p_\theta(h_{1:t-1}, r_{1:t-1}|u_{1:t-1}, u_t) \\
&\times p_\theta(h_t, r_t|u_{1:t-1}, u_t, h_{1:t-1}, r_{1:t-1}) \\
=&p_\theta(h_{1:t-1}, r_{1:t-1}|u_{1:t-1}, \cancel{u_t}) \\
&\times p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t, \cancel{h_{1:t-2}, r_{1:t-2}, u_{1:t-1}}) \\
=&p_\theta(h_{1:t-1}, r_{1:t-1}|u_{1:t-1}) p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t)
\end{aligned}
$$

It can be seen that in simplifying the above equations, those conditional independence properties hold for our generative model Eq. (4). Then,

$$
\begin{aligned}
&p_\theta(h_{1:t}|u_{1:t}, r_{1:t}) = \frac{p_\theta(h_{1:t}, r_{1:t}|u_{1:t})}{p_\theta(r_{1:t}|u_{1:t})} \\
=&\frac{p_\theta(h_{1:t-1}, r_{1:t-1}|u_{1:t-1}) p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t)}{p_\theta(r_{1:t}|u_{1:t})} \\
=&p_\theta(h_{1:t-1}|u_{1:t-1}, r_{1:t-1}) p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t) \\
&\times \frac{p_\theta(r_{1:t-1}|u_{1:t-1})}{p_\theta(r_{1:t}|u_{1:t})} \\
\propto&p_\theta(h_{1:t-1}|u_{1:t-1}, r_{1:t-1}) p_\theta(h_t, r_t|h_{t-1}, r_{t-1}, u_t)
\end{aligned}
$$

B Implementation Details

We implement the models with Huggingface Transformers repository of version 4.8.2. We initialize both the generative model and the inference model with DistilGPT-2, a distilled version of GPT2. For all of supervised pre-training, variational learning and JSA learning, we use the AdamW optimizer and a linear scheduler with 20% warm up steps and maximum learning rate 10^{-4}. The minibatch base size is set to be 8 with gradient accumulation steps of 4. The 3 random seeds for the results in Table 1 are 9, 10 and 11. The total epochs for supervised pre-training are 50, and those for both variational learning and JSA learning are 40. We monitor the performance on the validation set and apply early stopping (stop when the current best model is not exceeded by models in the following 4 epochs). We select the best model on the validation set, then evaluate it on test set. All our experiments are performed on a single 32GB Tesla-V100 GPU.

Redwood: Using Collision Detection to Grow a Large-Scale Intent Classification Dataset

Stefan Larson and **Kevin Leach**
Vanderbilt University
{firstname.lastname}@vanderbilt.edu

Abstract

Dialog systems must be capable of incorporating new skills via updates over time in order to reflect new use cases or deployment scenarios. Similarly, developers of such ML-driven systems need to be able to add new training data to an already-existing dataset to support these new skills. In intent classification systems, problems can arise if training data for a new skill's intent overlaps semantically with an already-existing intent. We call such cases *collisions*. This paper introduces the task of intent collision detection between multiple datasets for the purposes of growing a system's skillset. We introduce several methods for detecting collisions, and evaluate our methods on real datasets that exhibit collisions. To highlight the need for intent collision detection, we show that model performance suffers if new data is added in such a way that does not arbitrate colliding intents. Finally, we use collision detection to construct and benchmark a new dataset, *Redwood*, which is composed of 451 intent categories from 13 original intent classification datasets, making it the largest publicly available intent classification benchmark.

1 Introduction

As task-oriented dialog systems like Alexa and Siri have become more and more pervasive, tools enabling developers to build custom dialog systems have followed suit. Such tools—like Microsoft's Luis[1], Twilio's Autopilot[2], Rasa[3], and Google's DialogFlow[4]—enable engineers and dialog designers to craft dialog systems composed of *intents*, or core categories of competencies or skills in which the system is knowledgeable and to which the system can respond intelligently. New intents may be added periodically to the dialog system as part of its development and maintenance cycle, or dialog system models may be combined together (e.g., Clarke et al. (2022)).

[1] luis.ai [2] twilio.com/autopilot [3] rasa.com
[4] google.com/dialogflow

These phenomena may occur especially in real-world deployments, where datasets for dialog models may be developed, grown, and modified by large (and even disparate) teams over the span of a project's lifetime. Furthermore, dialog system models and their corresponding training datasets are sometimes offered as-a-service or "off-the-shelf" to dialog system builders who might not be fully familiar with the breadth or scope of the pre-existing dataset or model. If the builder adds a new intent to the dataset that overlaps with an existing intent, then the re-trained model's performance can suffer. As such, there is a need for tools and algorithms to help detect when a new intent overlaps—that is, *collides*—with an already-existing intent category.

In this paper, we introduce the challenge of *intent collision detection*, and develop several algorithms for determining whether a candidate intent category collides with another intent category. To do so, we curate and release a meta-dataset of 722 intents from 13 existing datasets. This graph-like meta-dataset consists of annotations indicating tuples of colliding intent pairs (examples of colliding intents can be seen in Table 1). We then introduce several collision detection algorithms and evaluate them on this meta-dataset.

We also use intent collision detection to build *Redwood*, a new intent classification dataset of 451 intent categories. *Redwood* is built by combining 13 smaller datasets. As a comparison, we also build *Redwood-naïve*, which is constructed by naïvely joining together all 13 datasets without arbitrating colliding intents. We find that classifier performance on *Redwood-naïve* to be substantially worse than *Redwood*, showcasing the negative effect of not addressing intent collisions in data.

Upon official release, *Redwood* will by far be the largest openly available intent classification dataset in terms of breadth of intent categories. Our hope is that the new *Redwood* dataset serves as a showcase for intent collision detec-

Proceedings of the SIGdial 2022 Conference, pages 468–477
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Dataset	Samples		
Snips	*how cold is it in princeton junction*	*will it be chilly in fiji at ten pm*	*is it foggy in shelter island*
Clinc-150	*give me the 7 day forecast*	*what's the temperature like in tampa*	*will it rain today*
MTOP	*what is the weather in new york today*	*how much is it going to rain tomorrow*	*give me the weather for march 13th*
Slurp	*set alarm tomorrow at 6 am*	*make an alarm for 4pm*	*set a wake up call for 10 am*
MTOP	*can you set a warning alarm for 7pm*	*set an alarm for monday at 5pm*	*make an alarm for the 5th*
Clinc-150	*wake me up at noon tomorrow*	*set my alarm for getting up*	*i need you to set alarm for me*
HWU	*how much is 1gbp in usd*	*what's the exchange rates*	*how much is $50 in pounds*
Clinc-150	*tell me five dollars in yen and rubles*	*how many pesos in one dollar us*	*usd to yen is what right now*
Banking-77	*do you know the rate of exchange*	*how is the exchange rate doing*	*what are the current exchange rates*
Clinc-150	*please start calling me mandy*	*I want you to call me this new name*	*the name you should call me is janet*
ACID	*how do i change my name*	*need my name to be updated*	*I need to fix my name in your system*
Banking-77	*where can I find how to change my name*	*details need to be modified*	*after I got married I need to change my name*
Snips	*play magic sam from the thirties*	*play music by blowfly from the seventies*	*play jeff pilson on youtube*
DSTC-8	*I want to hear the song high*	*I would like to listen to touch it on tv*	*I'd like to listen to the way I talk*
HWU	*please play yesterday from beattles*	*I'd like to hear queen's barcelona*	*play daft punk*
MetalWOz	*help me find restaurants in miami fl*	*I need help finding a place to eat*	*I need to find an italian restaurant in denver*
DSTC-8	*can you help find a place to eat*	*I'm looking for a filipino place to eat*	*I want to find a restaurant in albany*
HWU	*find me a nice restaurant for dinner*	*where can I get shawarma in this area*	*what's the best chicken place near me*
Outlier	*what is my balance*	*update me on my account balance*	*let me know how much money I have*
Clinc-150	*what's my current checking balance*	*what is the total of my bank accounts*	*how much total cash do I have in the bank*
DSTC-8	*I want to know my checking account balance*	*I'd like to check my balance*	*man how much money do I have in the bank*

Table 1: Examples of data that will trigger collisions. Each row of the table displays three samples from a single intent in a particular dataset. Among these three samples, each line collides with an intent category from the other two datasets.

tion as well as a new, publicly-available, large-scale challenge dataset for intent classification models for dialog systems. Both the collision meta-dataset and *Redwood* are publicly available at `github.com/gxlarson/redwood`.

2 Related Work

The Collision Detection Task. We discuss three areas of related work related to our proposed intent collision detection task: generalized zero-shot learning, open set classification, and out-of-domain (or out-of-scope) sample identification.

In generalized zero-shot learning (e.g., Zhang et al. (2022)), a model is trained with data from a set of "seen" label classes (e.g., intents) and, during inference, must identify test samples as belonging to either a "seen" label class or an "unseen" class for which the model has limited auxiliary knowledge (e.g., descriptions of unseen classes, but no concrete training examples).

Both open set classification and out-of-domain sample identification refer to the modeling task of classifying inference samples among label classes seen during training or to identify if the sample belongs to an unknown or undefined label class (e.g., Larson et al. (2019b); Zhang et al. (2021)). Slot-filling models that are trained on B/I/O tags naturally predict the unknown class label as O tags, but for intent classifiers the task is much more challenging since it requires curating viable training data for an out-of-domain category (i.e., it is challenging to know in advance what types of out-of-domain inputs a system might encounter).

Our proposed task of intent collision detection differs from the aforementioned tasks because "inference" samples need not be considered one at a time, but can instead be grouped together into entire candidate intent categories. This enables considering entirely different modeling tasks like those discussed in Section 3.3. Nevertheless, both our meta-dataset of intent collisions and *Redwood* allow for the evaluation of both zero-shot and generalized zero-shot learning models, and the *Redwood* intent classification dataset includes a substantial number of out-of-domain samples for evaluating open set classification and out-of-domain sample detection.

Intent Classification Corpora. There are several smaller corpora for evaluating intent classification models, some spanning broad domains (e.g., Liu et al. (2019), Larson et al. (2019b), Li et al. (2021)) and others focusing fine-grained evaluation of individual domains (e.g., the Banking-77 corpus (Casanueva et al., 2020) with respect to the personal banking domain). While most datasets are constructed via crowdsourcing, our new Redwood dataset is constructed from both (1) already existing datasets and (2) newly crowdsourced intents.

Dataset Derivation and Combination. Datasets are sometimes formed from other datasets, either

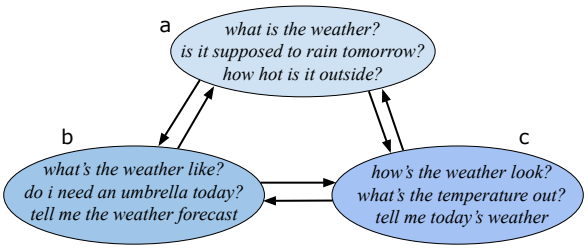


Figure 1: Transitive collisions.

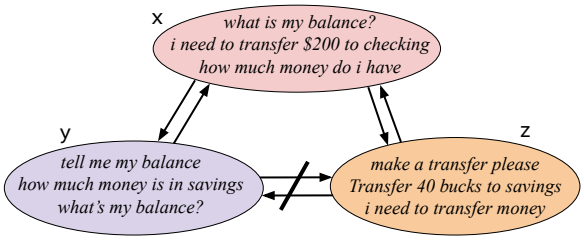


Figure 2: Non-transitive collisions.

by deriving a new dataset from an existing one, or by combining datasets together. The former category include translations of dialog datasets (e.g., (Upadhyay et al., 2018; Xu et al., 2020)) as well as re-formulations of existing datasets into new tasks (e.g., converting a semantic role labeling (SRL) dataset to open information extraction (OIE) data as done in Solawetz and Larson (2021)).

Dataset combination has been used in other fields beyond dialog systems and conversational AI. For instance, Song et al. (2020) combined several speech recognition datasets together to form their *SpeechStew* dataset. As there are no target labels analogous to intents in automatic speech recognition, the creators of *SpeechStew* did not have to consider collisions among intent categories. In this paper, our focus is primarily on dataset combination, but we also derive intent classification data from several turn-based dialog corpora (*MetalWOz* and *DSTC-8*, discussed in Section 3.4).

3 Detecting Collisions

In this section we discuss our proposed challenge, intent collision detection. We begin with a motivating example showing why detecting collisions is important, as well as a formal problem statement. Then, we introduce and evaluate several collision detection baselines on our meta-dataset.

3.1 Motivating Example

As a motivating example, suppose our intent classification system has been trained on the *Clinc-150* dataset (Larson et al., 2019b), an intent classification dataset consisting of 150 intents.[5] The *Clinc-150* dataset includes an intent called `weather`, which is meant to handle weather-related queries such as "*what's the weather like today*" and "*tell me the weather in New York*." Suppose further that a new developer or a new team attempts to update the intent classifier with new data that contains a new intent category, such as the `get_weather` intent from the *HWU* dataset[6] (Liu et al., 2019). In such a scenario, there are now training data samples that overlap substantially, but that are labeled with different intents (`weather` vs. `get_weather` in this example). Thus, upon updating the model by training on *HWU*'s `get_weather` data, the predictive performance on any weather-related inference queries might be split between these two intents. This disparity can also cause unintended consequences downstream in production models, such as calls to database systems that are triggered based on the user's intent.

Indeed, when we train a BERT classifier on the original *Clinc-150* training set, the accuracy on the `weather` test set is 100%. When we add a *HWU*'s `get_weather` intent to *Clinc-150* to create a new 151^{st} intent and re-train the BERT classifier, we observe an accuracy score of 60% on the `weather` test set. This performance drop is a symptom of having added an intent category that collides with another intent category. Such a model—which was trained on colliding intents—could cause unexpected behavior on downstream events, especially if the `weather` and `get_weather` intents trigger different business logic workflows or system responses. We note that, while in this example, the colliding `weather` and `get_weather` intent names are quite similar, other colliding pairs like Snips' `search_screening_event` and MetalWOz's `movie_listings` do not have lexically similar intent names, precluding straightforward string matching of intent names.

3.2 Problem Statement

In this subsection, we formally define our collision detection problem. We first consider a scenario in which we have two intent classification datasets, $\mathcal{A}$ and $\mathcal{B}$, where $A_i \in \mathcal{A}$ and $B_j \in \mathcal{B}$ refer to specific intent categories in each. We say that intent categories A_i and B_j *collide* if there exist a suf-

[5] In this paper, dataset names are in *italics* and intent names are in `teletype font`. Example queries are in *italics* and in quotes if they appear in-line.

[6] Recall from Section 1 that such updates from new teams or new developers may be from routine perfective maintenance during a model's lifetime.

ficient number of queries in A_i that *semantically overlap* with a sufficient number of queries in B_j. This *semantic overlap* can occur when a developer attempts to add new intent categories to a starting training dataset—when an intent classification model trained on the combined dataset $\mathcal{A} \cup \mathcal{B}$ will cause queries belonging to A_i to be classified in B_j (and vice versa).

As an example, suppose we have an intent classifier built from a starting dataset such as *Clinc-150*, which, among other things, contains a `weather` intent category for weather-related inquiries (cf. Section 3.1). Suppose further that we seek to grow this starting dataset by adding datapoints from a candidate dataset such as *HWU* (see Section 3.1, which contains a `get_weather` intent category). If we naïvely combine these two datasets together, a resulting intent classifier will result in some queries from the original `weather` category to be classified to the newly-added `get_weather` category because these two categories are semantically similar. Table 1 illustrates several example colliding intents and associated queries. Our approach addresses these collisions by detecting their prevalence and quantifying their impact automatically, aiding developers in improving the quality of their datasets and scope of their dialogue systems.

Because the notion of semantic overlap can differ from category to category and dataset to dataset, we observe several classes of relationships among colliding intent categories in practice. In particular, intent collisions can be *simple-pairwise*, *transitive*, or *hierarchical*. In the simple-pairwise case, two intents collide with each other *only*, and not with any other intent in either dataset. However, we also observe transitivity within intent classes. Figure 1 illustrates example utterances within intent classes `a`, `b`, and `c`, where all intent classes are transitively related to one another in a cycle.

Lastly, we observe non-transitive hierarchies among colliding intents. In this case, a broad intent category from one dataset can collide with two or more intent categories that do not relate to each other. Figure 2 shows a hypothetical intent class `x` consisting of general banking queries, including balance inquiries and transfer requests, and classes `y` and `z` consist solely of balance inquiry and transfer requests, respectively. Here, because class `x` is more broad than `y` and `z`, each of `y` and `z` collide with `x`, but `y` and `z` do not collide with each other. Our approach can help developers reveal such cases when managing datasets, and we consider these collision relationships in the creation of our Redwood dataset.

Dataset	# Intents	# Collisions
ACID	175	36
Clinc-150	150	158
MTOP	113	60
Banking-77	77	25
HWU	64	103
New	58	5
MetalWOz	51	80
DSTC-8	34	67
ATIS	26	7
Outlier	10	9
Snips	7	20
Jobs640	1	0
Talk2Car	1	0
Total	767	570

Table 2: Number of intents with collisions. A total of 570 intents have at least one collision.

3.3 Approaches

We introduce two approaches for detecting collisions: *Classifier Confusion* and *Data Coverage*.

Classifier Confusion. A column of a confusion matrix charts the distribution of predictions of a classifier for data in a particular category. We call such a distribution the *classification distribution*. We adapt this notion for our first collision detection approach, which identifies a candidate intent A to collide with $B \in \mathcal{C}$ if a classifier model trained on dataset $\mathcal{C}$ produces a *classification distribution* d such that $\frac{max(d)}{sum(d)} > \tau$, where τ is a threshold set by the developer. We call this ratio the *classifier collision score*.

Data Coverage. We define the *coverage* of one intent B over another intent A as

$$\text{Coverage}(A, B) = \frac{1}{|B|} \sum_{b \in B} \max_{a \in A} sim(a, b).$$

Here, $sim(a, b)$ computes the similarity between two phrases a and b (for instance, $sim(a, b)$ could be the cosine similarity between two phrase embeddings or the Jaccard similarity between n-*gram* sets). The coverage metric can be used to detect if two intents collide using a threshold rule. In other words, A and B collide if Coverage(A,B) $> \kappa$, where κ is a threshold chosen by the developer. We call the coverage metric the *coverage score*.

```
[
    {
        "source": "clinc150",
        "intent": "weather"
        "name": "clinc150__weather"
        "collisions": [
            "snips__get_weather",
            "hwu__get_weather",
            "mtop__get_weather",
            "metalwoz__weather_check",
            "dstc8__GetWeather"
        ]
    },
    {
        "source": "hwu",
        "intent": "general_praise",
        "name": "hwu__general_praise",
        "collisions": [
            "acid__st_thank_you",
            "clinc150__thank_you"
        ]
    },
    ...
]
```


Figure 3: Example entries in the graph-like collision meta-dataset, showing collisions for *Clinc-150*'s `weather` intent and *HWU*'s `general_praise` intent.

3.4 Datasets

We evaluate the effectiveness of our intent collision approaches using several indicative datasets. These datasets can be roughly grouped into three categories: (1) intent classification datasets like *Clinc-150* (Larson et al., 2019b), *Banking-77* (Casanueva et al., 2020), *ACID* (Acharya and Fung, 2020), *Outlier* (Larson et al., 2019a), and *New* (this work; a corpus that was crowdsourced in a manner similar to Larson et al. (2019b) and Larson et al. (2019a)); (2) joint slot-filling and intent classification or semantic parsing datasets like *ATIS* (Hemphill et al., 1990; Hirschman et al., 1992, 1993; Dahl et al., 1994), *Snips* (Coucke et al., 2018), *HWU* (Liu et al., 2019), and *MTOP* (Li et al., 2021); and (3) turn-based dialog datasets like *DSTC-8* (Kim et al., 2019) and *MetalWOz* (Lee et al., 2019). We only consider the initial queries in the turn-based *DSTC-8* and *MetalWOz*, and discard all subsequent dialog turns.

Queries in these datasets span a wide range of topic domains, including banking and personal finance (*Banking-77* and *Outlier*) and insurance (*ACID*); other datasets cover a wide array of topic domains, such as *Clinc-150* and *HWU*, which cover smart home, automotive, travel, banking, cooking, and others. Since we are concerned with detecting colliding intents, we do not consider any slot annotations, and we use only the first turns from the multi-turn dialog datasets. In addition, we also use the *Jobs640* (Califf and Mooney, 1997) and *Talk2Car* (Deruyttere et al., 2019) datasets, which, although not originally designed for intent classification tasks, are categorized in a way that admit consideration as single-intent classification for our purposes. Table 2 summarizes these datasets.

The Collision Meta-Dataset We constructed a graph-like dataset that indicates the collision relationships between intents. To build this dataset, we reviewed all intents from all of the datasets listed in Table 2 to check for collisions between other intents. We developed a ground truth set of tuples indicating whether two intents collide among these datasets. Figure 3 shows the structure of the intent collision meta-dataset, and Table 2 displays the number of collisions that occur relative to each individual dataset. The meta-dataset includes the three types of collisions defined in Section 3.2.

3.5 Experimental Evaluation

Implementation Details. We evaluate our intent collision detection methods on our newly-created collision meta-dataset. For evaluating the classifier confusion approach, we train a multi-class intent classifier on each individual dataset (except the single-intent datasets) and then run inference on all other intents from the other datasets. We compute and report the classifier confusion score for each run. In our experiments, we use a linear SVM classifier with bag-of-words feature representations.

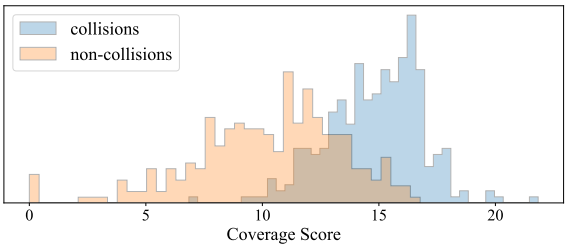
For evaluating the data coverage approach, we first sample[7] a nearly equal number of colliding and non-colliding intent pairs from the collision meta-dataset. We then compute the coverage scores for the selected pairs using several sentence representation and similarity metrics. We use the SBERT library's SBERT-NLI and SBERT-miniLM sentence embedders (Reimers and Gurevych, 2019) along with cosine similarity. Additionally, we also use n-gram-based similarity, defined as

$$sim(a,b) = \frac{1}{N}\sum_{n=1}^{N}\frac{|n\text{-grams}_a \cap n\text{-grams}_b|}{|n\text{-grams}_a \cup n\text{-grams}_b|}$$

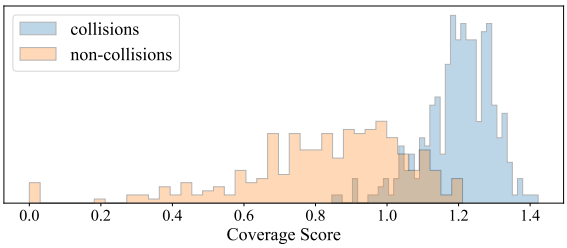
where a and b are queries from two intents, and $N = 3$ in our experiments.

For both the data coverage and classifier confusion experiments, we only consider intents that

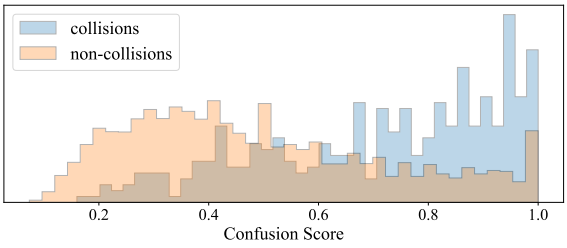
[7] Sampling avoids combinatorial explosion of possible intent pairs.



(a) SBERT-NLI Coverage Score



(b) Mini-LM Coverage Score



(c) SVM classifier confusion.

Figure 4: Data coverage and classifier confusion score distributions for various intent collision detection approaches.

have at least 10 queries. For the collision detection experiments, we used all 285 collision pairs and sampled 300 non-colliding pairs since there are substantially more non-colliding pairs. The classifier confusion approach does not compare intents in a pairwise manner, and instead compares a dataset (i.e., a classifier trained on a dataset) against a single intent at a time. We run a classifier on all multi-intent datasets, which yielded a total of 400 collision pairs and 6,802 non-collision pairs for the classifier confusion experiments.

Metrics. While in actual application settings, a user may wish to use thresholds for τ and κ (defined earlier in Section 3.3) to determine whether intents collide, we evaluate both classifier confusion and coverage methods in a threshold-free manner using the AUC score. (In practice, values for τ and κ could be set by the practitioner via cross-validation or by using the meta-dataset provided in this work to set optimal thresholds for their application.) The AUC score allows us to judge each method's ability to distinguish collisions versus non-collisions; an AUC score of 1.0 means perfect separability between collisions and non-collisions, while an AUC score of 0.5 means a method is unable to distinguish between colliding and non-colliding intents.

Approach	Coverage AUC	Confusion AUC
SBERT-NLI	0.898	—
token	0.931	—
SBERT-miniLM	0.963	—
SVM-based	—	0.756

Table 3: AUC metrics for each intent collision detection approach.

3.6 Results

Data Coverage. Figure 4 charts coverage scores and confusion scores for various approaches. In Figure 4 (a) and (b), the coverage approaches tend to return higher coverage scores for non-collisions and lower coverage scores for collisions, which aligns with our expectations given our definition of the coverage metric and assuming the similarity metric used in the coverage computation is effective. The AUC scores allow us to quantitatively judge the performance of the various coverage-based approaches: in Table 3, the SBERT-miniLM embedding method yields the highest AUC score, and interestingly the n-gram-based coverage method performs second best, with the SBERT-NLI embedding method in third.

Classifier Confusion. Figure 4 (c) charts classifier confusion scores for the SVM-based classifier confusion approach. Our results demonstrate that actual intent collisions typically yield high classifier confusion scores, while non-collisions yield lower confusion scores. Visually, however, Figure 4 (c) seems to indicate that that the classifier confusion approach is less effective than the coverage-based approaches. This is made more apparent by the AUC score in Table 3. We note that the data coverage and classifier confusion AUC scores are not directly comparable as they use different evaluation settings. Nonetheless, the difference in performance scores does lead us to conclude that the data coverage approach is more effective.

In sum, these experimental results demonstrate that the two intent collision detection approaches introduced here are effective in detecting collisions

Original Dataset	Original Intent	Sample
HWU	alarm_remove	*remove the alarm set for 10pm*
Clinc-150	reminder_update	*set a reminder for me to take my meds*
MTOP	get_weather	*should i wear a raincoat tuesday*
Jobs-640	—	*what systems analyst jobs are there in austin*
Talk2Car	—	*switch to right lane and park on right behind parked black car*
Jobs640	Jobs640	*what systems analyst jobs are there in austin*
Snips	add_to_playlist	*add paulinho da viola to my radio rock song list*
Outlier	hours	*tell me the hours of operation for my bank*
New	balance	*do I have holiday time saved*
DSTC-8	LookupMusic	*I like metal songs can you find me some*
ATIS	ground_service	*i'll need to rent a car in washington dc*
MetalWOz	name_suggester	*I need to find a name for my new cat*
Clinc-150	find_phone	*can you help me find my cell*
ACID	info_amt_due	*what is the current amount due on my account*
Banking-77	terminate_account	*how do I deactivate my account*
Clinc-150	measurement_conversion	*what amount of millimeters are in 50 kilometers*
ACID	info_name_change	*i need to fix my name*
MTOP	play_music	*find me the latest linkin park album*
HWU	audio_volume_up	*just increase the volume a little*
Outlier	balance	*how much oney do i have available*
Vertanen (2017)	—	*why on earth is there cereal in the fridge*
Vertanen (2017)	—	*who are you going to vote for in november*
Vertanen (2017)	—	*do you know where i put my glasses*
Clinc-150	out-of-scope	*what size wipers does this car take*
Clinc-150	out-of-scope	*how long is winter*
Clinc-150	out-of-scope	*are any earning reports due*

Table 4: Sample intents and queries from our *Redwood* dataset, along with the corresponding original dataset and intent (where applicable). Samples are grouped into *in-scope* (top) and *out-of-scope* (bottom).

among real datasets, with the data coverage approach being the stronger of the two.

4 Building the *Redwood* Dataset

With tools addressing the problem of intent collision detection in hand, we now turn our attention to combining the individual datasets from Table 2 together to form a single large-scale intent classification dataset, *Redwood*. This section discusses the construction of *Redwood* and a companion *out-of-scope* evaluation set, and then evaluates several benchmark intent classifiers on the dataset. These datasets and associated evaluations demonstrate the consequences of leaving colliding intents unaddressed, providing a valuable resource for the community to improving intent classification models.

4.1 Data

***In-Scope* Data.** After creating the collision meta-dataset, a natural extension was to combine each dataset together to form *Redwood*. We used the collision meta-dataset to help inform us of which intents could combined, and which intents could stand alone in *Redwood*. In some cases, we removed intents that caused *hierarchical* collisions, as sometimes joining together intents from a hierarchical collision produced an intent that was too broad. We included only those intents that have at least 50 queries, and the resulting *Redwood* consists of 451 total intents and 62,216 queries. Following the terminology used in Larson et al. (2019b), we call these 451 intents *in-scope*.

Dataset	N. Samples
Vertanen (2017)	2067
Clinc-150	1200
Total	3267

Table 5: Sources of *out-of-scope* data and number of samples used in *Redwood*'s out-of-scope test set.

By way of comparison, we also produced a "naïve" version of *Redwood*, called *Redwood-naïve*, where all the intents from the datasets listed in Table 3.4 were joined together *without* using collision detection or any other method of arbitrating or correcting colliding intents. Like the original *Redwood*, we included only intents that have at least 50 queries, and capped each intent at a maximum of 150 queries so as to avoid drastic class imbalances. *Redwood-naïve* consists of 619 intents and 85,746 total queries.

All versions of *Redwood* were split into train and test splits per intent: 85% training, 15% testing.

***Out-of-Scope* Data.** In contrast to *in-scope*, *out-of-scope* queries are those that do not belong to any of the *in-scope* intents. Considering *out-of-scope* queries in an evaluation of intent classification models is important because such queries occur in production settings, where end users cannot be expected to know the full range of intents when interacting with a conversational AI system. We include a collection of 3,267 *out-of-scope* queries in addition to the *Redwood* corpus. *Redwood*'s *out-of-scope* data originates from the following sources: *Clinc-150* dataset, which itself includes a set of *out-of-scope* queries; and Vertanen (2017), a crowd-sourced dialog dataset from which we use the first dialog turns. We reviewed all candidate *out-of-scope* queries, removing those that were actually *in-scope*. Examples of queries from the *Redwood* dataset are shown in Table 4.

4.2 Benchmark Evaluation

Models. We benchmark intent classification performance using the MobileBERT model (Sun et al., 2020) using the HuggingFace library (Wolf et al., 2020). The MobileBERT implementation uses a softmax function to compute logits to a probability vector $\boldsymbol{p}$, from which we can obtain confidence scores for each intent. These confidence scores can be used to predict whether a query is in- or out-of-scope, according to a decision threshold t given by

$$\text{decision rule} = \begin{cases} \text{in-scope}, & \text{if } \max(\boldsymbol{p}) \geq t \\ \text{out-of-scope}, & \text{if } \max(\boldsymbol{p}) < t. \end{cases}$$

Such decision rules were used in Hendrycks and Gimpel (2017) and Larson et al. (2019b).

Metrics and Experiments. We measure intent classifier accuracy on in-scope data without considering out-of-scope inputs. We also measure each model's ability to distinguish in-scope and out-of-scope queries by computing the AUC between in- and out-of-scope confidence scores. In this way, we use AUC to measure how separable in- and out-of-scope queries based on their confidence scores without having to select an confidence threshold t. An AUC score of 0.5 (the minimum AUC score) implies the model cannot distinguish in- versus out-of-scope inputs. An AUC of 1.0 indicates the model can perfectly separate inputs.

Training Dataset	In-Scope Accuracy	Clinc OOS AUC	Vertanen OOS AUC
Redwood	0.913	0.921	0.928
Redwood-naïve	0.861	0.909	0.925

Table 6: Model performance of the MobileBERT classifier on *Redwood* and *Redwood-naïve*.

Collisions	0	1	2	3	4	5	6	14
Mean Acc.	0.91	0.80	0.81	0.79	0.81	0.80	0.89	0.57
Size	322	74	42	51	15	11	13	8

Table 7: Accuracy scores on *Redwood-naïve* intents per number of collisions.

4.3 Results

Model performance on *Redwood-naïve* and *Redwood* is shown in Table 6. First, we notice that the intent classifiers perform reasonably well on the in-scope classification task, with MobileBERT classifying queries with 91% accuracy. The models also perform well on the out-of-scope task, and discriminate between in- and out-of-scope queries with AUC scores of 0.921 and 0.928 on the *Clinc-150* and Vertanen (2017) out-of-scope data.

The bottom half of Table 6 presents model performance when trained and tested on *Redwood-naïve*. In this case, model performance is substantially worse than models trained on the carefully-crafted *Redwood* dataset, confirming our hypothesis from Section 3.1 that model performance suffers if trained on data with colliding intents.

We drill deeper into the impact of intent collisions on models trained on *Redwood-naïve* in Table 7 which charts per-intent accuracy based on the number of other intents that collide with that intent. This table groups intents based on the number of collisions, and we see that on average, intents with no collisions exhibit higher accuracy than intents with collisions. In general, colliding intents lead to degraded accuracy: intents with one or more collisions have accuracy of around 10 or more points lower than the no-collision group, with the exception of the 6-collision group. The average accuracy of the 6-collision group on *Redwood-naïve* is indeed surprising, and we posit that the MobileBERT model—a high-capacity transformer model—can learn the nuances of each individual intent, even if they do semantically collide.

5 Conclusion and Future Work

This paper introduces the task of intent collision detection when constructing or updating an intent

classification model's dataset to incorporate additional intents. Using 13 individual datasets, we constructed a meta-dataset to track intent collisions between the datasets, and then introduced and evaluated two intent collision detection techniques and found that both perform effectively at the collision detection task. To help measure and address this problem, we constructed *Redwood*, a large-scale intent classification dataset consisting of 451 intents and over 60,000 queries. We used *Redwood* to benchmark several intent classification models on the task of in-scope query prediction and out-of-scope detection, The new *Redwood* dataset is the largest publicly available intent classification benchmark, in terms of number of intents, and will be made publicly available. Future work will include annotating slots to extend Redwood to joint intent classification and slot-filling, and it is likely that new tools will have to be developed for doing so. Additionally, using the collision detection methods introduced in this paper, *Redwood* can be periodically updated with new intents whenever other new intent classification datasets are published.

Acknowledgements

We thank the anonymous reviewers for their detailed and thoughtful feedback, and Jacob Solawetz for his feedback on early iterations of the *Redwood* concept.

References

Shailesh Acharya and Glenn Fung. 2020. Using optimal embeddings to learn new intents with few examples: An application in the insurance domain. In *Proceedings of the KDD 2020 Workshop on Conversational Systems Towards Mainstream Adoption (KDD Converse 2020)*.

Mary Elaine Califf and Raymond J. Mooney. 1997. Relational learning of pattern-match rules for information extraction. In *CoNLL97: Computational Natural Language Learning*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*.

Christopher Clarke, Joseph Peper, Karthik Krishnamurthy, Walter J. Talamonti, Kevin Leach, Walter S. Lasecki, Yiping Kang, Lingjia Tang, and Jason Mars. 2022. One agent to rule them all: Towards multi-agent conversational ai. *ArXiv*, abs/2203.07665.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *CoRR*, abs/1805.10190.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Thierry Deruyttere, Simon Vandenhende, Dusan Grujicic, Luc Van Gool, and Marie-Francine Moens. 2019. Talk2Car: Taking control of your self-driving car. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations (ICLR)*.

L. Hirschman, M. Bates, D. Dahl, W. Fisher, J. Garofolo, D. Pallett, K. Hunicke-Smith, P. Price, A. Rudnicky, and E. Tzoukermann. 1993. Multi-site data collection and evaluation in spoken language understanding. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Lynette Hirschman, Madeleine Bates, Deborah Dahl, WIlliam Fisher, John Garofolo, Kate Hunicke-Smith, David Pallett, Christine Pao, Patti Price, and Alexander Rudnicky. 1992. Multi-site data collection for a spoken language corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, Minlie Huang, Luis Lastras, Jonathan K. Kummerfeld, Walter S. Lasecki, Chiori Hori, Anoop Cherian, Tim K. Marks, Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, and Raghav Gupta. 2019. The eighth dialog system technology challenge. In *NeurIPS Workshop: Conversational AI: Today's Practice and Tomorrow's Potential*.

Stefan Larson, Anish Mahendran, Andrew Lee, Jonathan K. Kummerfeld, Parker Hill, Michael A. Laurenzano, Johann Hauswald, Lingjia Tang, and Jason Mars. 2019a. Outlier detection for improved data

quality and diversity in dialog systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019b. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Sungjin Lee, Hannes Schulz, Adam Atkinson, Jianfeng Gao, Kaheer Suleman, Layla El Asri, Mahmoud Adada, Minlie Huang, Shikhar Sharma, Wendy Tay, and Xiujun Li. 2019. Multi-domain task-completion dialog challenge. In *Dialog System Technology Challenges 8*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL)*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. In *Proceedings of the Tenth International Workshop on Spoken Dialogue Systems Technology (IWSDS)*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Jacob Solawetz and Stefan Larson. 2021. LSOIE: A large-scale dataset for supervised open information extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume (EACL)*.

Congzheng Song, Alexander Rush, and Vitaly Shmatikov. 2020. Adversarial semantic collisions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *Proceedings of the IEEE ICASSP*.

Keith Vertanen. 2017. Towards improving predictive aac using crowdsourced dialogues and partner context. In *ASSETS '17: Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Hanlei Zhang, Xiaoteng Li, Hua Xu, Panpan Zhang, Kang Zhao, and Kai Gao. 2021. TEXTOIR: An integrated and visualized platform for text open intent recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*.

Yiwen Zhang, Caixia Yuan, Xiaojie Wang, Ziwei Bai, and Yongbin Liu. 2022. Learn to adapt for generalized zero-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Dialogue Evaluation with Offline Reinforcement Learning

Nurul Lubis, Christian Geishauser, Hsien-Chin Lin,
Carel van Niekerk, Michael Heck, Shutong Feng, Milica Gašić
Heinrich Heine University Düsseldorf, Germany
{lubis, geishaus, linh, niekerk, heckmi, fengs, gasic}@hhu.de

Abstract

Task-oriented dialogue systems aim to fulfill user goals through natural language interactions. They are ideally evaluated with human users, which however is unattainable to do at every iteration of the development phase. Simulated users could be an alternative, however their development is nontrivial. Therefore, researchers resort to offline metrics on existing human-human corpora, which are more practical and easily reproducible. They are unfortunately limited in reflecting real performance of dialogue systems. BLEU for instance is poorly correlated with human judgment, and existing corpus-based metrics such as success rate overlook dialogue context mismatches. There is still a need for a reliable metric for task-oriented systems with good generalization and strong correlation with human judgements. In this paper, we propose the use of offline reinforcement learning for dialogue evaluation based on a static corpus. Such an evaluator is typically called a critic and utilized for policy optimization. We go one step further and show that offline RL critics can be trained the static corpus for any dialogue system as external evaluators, allowing dialogue performance comparisons across various types of systems. This approach has the benefit of being corpus- and model-independent, while attaining strong correlation with human judgements, which we confirm via an interactive user trial.

1 Introduction

With the rise of personal assistants, task-oriented dialogue systems have received a surge in popularity and acceptance. Task-oriented dialogue systems are characterized by a user goal which motivates the interaction, e.g., booking a hotel, searching for a restaurant or calling a taxi. The dialogue agent is considered successful if it is able to fulfill the user goal by the end of the interaction.

Ideally, success rates are obtained via interaction with a real user in-the-wild. Unfortunately, with a handful of exceptions, e.g., LetsGO (Lee et al., 2018) and Alexa Challenge (Gabriel et al., 2020), that is often out of reach. The closest approximation is human trials with paid users such as Amazon Mechanical Turk workers, which has also been adopted as final evaluation in recent incarnations of the Dialogue State Tracking Challenge (DSTC) (Gunasekara et al., 2020). However, such evaluations are highly time- and cost-intensive, making them impractical for optimization during an iterative development. The third alternative is to use a user simulator to conduct online dialogue simulation, however the result is subject to the quality of the user simulator itself. Furthermore, developing such simulators is far from straightforward and requires significant amounts of handcrafting (Schatzmann, 2008). Only recently we have seen data-driven user simulators that can compete with hand-coded ones (Lin et al., 2021).

While there has been considerable progress towards more meaningful automatic evaluation metrics for dialogues, there remains a number of limitations as highlighted by the recent NSF report (Mehri et al., 2022): the metrics 1) measure only a limited set of dialogue qualities, which mostly focus on subjective aspects such as fluency and coherence, 2) lack generalization across datasets and models, and 3) are not yet *strongly* correlated with human judgements. These limitations hinder a more widespread use of newly proposed metrics for benchmarking and comparison, especially with prior works. Further, in particular for task-oriented dialogue systems, the need for reliable automatic evaluation of dialogue success is still unanswered.

Being able to automatically evaluate the success rate of any policy using static data offers a number of benefits in terms of required resources, generalizability, and reproducibility. Furthermore, it is not only suitable for the final evaluation of a dialogue policy, but can also be utilized as an objective for iterative optimization. The corpus-based success

Proceedings of the SIGdial 2022 Conference, pages 478–489
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

rate is one such method, which has become the standard metric for state-of-the-art comparisons of policy optimization approaches today (Budzianowski et al., 2018). Unfortunately, this metric is computed based on pseudo-dialogues that may contain context mismatch. Therefore, we believe it should be treated more as an approximation: it is insufficient at best, and misleading at worst, in reflecting real performance of dialogue systems. In addition, the rules used to check the goal completion need to be handcrafted based on the ontology, making this method data- or ontology-dependent.

In this paper, we propose to use offline reinforcement learning (RL) to train a policy evaluator, also known as a critic, based on a static collection of dialogue data[1]. We show that an offline critic addresses the limitations of current automatic metrics: 1) it can be trained to evaluate any dialogue system architecture after-the-fact, allowing comparisons across various types of systems from prior works, 2) it can be utilized in the iterative development phase to optimize a dialogue policy, 3) it is theoretically grounded, solving the problems that standard corpus-based success rate has due to context mismatch, and 4) it strongly correlates with the performance of the system when interacting with human users, which we confirm via a user trial.

2 Related Work

For a long time, the research in dialogue policy has focused on user-centered criteria such as user satisfaction (Walker et al., 1997; Lee and Eskénazi, 2012; Ultes et al., 2017). The most reliable way to obtain these scores is to have users interact directly with the system and let them subjectively rate the system afterwards. Due to the time and resource requirements to carry out such evaluations, human trials are usually done only as the final evaluation after the system development is finished.

As the line between policy and natural language generation (NLG) tasks becomes blurred, we see the introduction of metrics such as BLEU (Papineni et al., 2002) and perplexity. However, these have been labeled early on to be potentially misleading, as they correlate poorly with human judgement (Stent et al., 2005; Liu et al., 2016). This circumstance motivates automatic metrics that are highly correlated with human ratings (Dziri et al., 2019; Mehri and Eskenazi, 2020a,b). However, these metrics are designed to measure subjective quality of a dialogue response, making them more suitable for evaluating chat-based systems.

Despite the availability of toolkits that facilitate user simulation (US) evaluation (Zhu et al., 2020), corpus-based match and success rates are the default benchmark for works in task-oriented dialogue systems today (Budzianowski et al., 2018; Nekvinda and Dušek, 2021). These metrics are practical to compute, reproducible, and scalable. Current standard corpus-based metrics are computed on a pseudo-dialogue constructed using user utterances from data and responses generated by the system. A set of rules then checks whether the system provides all information requested by the user. Unfortunately, they do not take into account context mismatches that may originate from the pseudo-dialogue construction and therefore does not reflect other aspects of dialogue quality as the resulting dialogue flow is completely overlooked.

There has been few applications of offline RL to dialogue systems. Jaques et al. (2019) explores various language-based criteria, e.g., sentiment and semantic similarity, as reward signals for open-domain dialogue, paired with a Kullback-Leibler (KL) control for exploration within the support of the data. Verma et al. (2022) proposed using fine-tuned language models to utilize unlabeled data for learning the critic function. The method is however only demonstrated on a very small state and action space, and it is therefore unclear whether it generalizes to more complex set ups. Ramachandran et al. (2021) applied offline RL with a pair-wise reward learning model based on preference learning, however it still utilizes the corpus-based success rate for choosing the preferred rollout. To the best of our knowledge, offline RL has not previously been deployed for dialogue evaluation.

3 Preliminaries

3.1 Offline RL

Dialogue can be formulated as a reinforcement learning problem with a Markov decision process (MDP) $\mathcal{M} = \{\mathcal{S}, \mathcal{A}, r, p, p_0, \gamma\}$. In this MDP, $\mathcal{S}$, $\mathcal{A}$, and r denote the state and action spaces, and the reward function, respectively. $p(s_{t+1}|s_t, a_t)$ denotes the probability of transitioning to state s_{t+1} from s_t after executing a_t, and $p_0(s)$ is the probability of starting in state s. $\gamma \in [0, 1]$ is the discount factor that weighs the importance of immediate and future rewards. At each time step t, the agent ob-

[1] https://gitlab.cs.uni-duesseldorf.de/general/dsml/lava-plas-public

serves a state s_t, executes its policy π by selecting an action a_t according to $\pi(a_t|s_t)$, transitions to a new state s_{t+1} and receives a reward r_t. The goal of the policy is to maximize the cumulative discounted rewards, i.e., the return $R_t = \sum_{i \geq 0} \gamma^i r_{t+i}$.

Instead of interacting with the MDP to learn a policy, offline RL aims to learn a policy exclusively from previously collected data containing state transitions $\mathcal{D} = \{(s_i, a_i, s_{i+1}, r_i)\}_i$ under an unknown behavior policy π_β. This set-up is especially useful in cases where deploying the agent in the real environment is too costly, as is the case with real user interaction for dialogue systems. As the agent can not interact with the environment, the performance of the trained policy π needs to be evaluated also based on the data $\mathcal{D}$. The Q-value $Q_\pi(s_t, a_t)$ denotes the expected return when executing a_t in s_t and following policy π thereafter. Q-learning algorithms estimate the Q-function Q_π by iteratively applying the Bellman operator

$$\mathcal{T}Q(s_t, a_t) = \mathbb{E}_{s_{t+1}}[r_t + \gamma Q(s_{t+1}, a_{t+1})]. \quad (1)$$

Value-based RL methods optimize the policy by maximizing the Q-values for every state-action pair $(s_t, a_t) \in \mathcal{S} \times \mathcal{A}$. With discrete actions, and for given state s, the actor can then simply select $\text{argmax}_a Q(s, a)$ in a greedy fashion.

Alternatively, with an actor-critic method, an actor is trained which optimizes its parameters to maximize the expected return of the starting states, for example via the deterministic policy gradient method (Silver et al., 2014; Lillicrap et al., 2016):

$$\nabla_\theta J(\theta) = \mathbb{E}_{s \sim \mathcal{S}}[\nabla_\theta \pi_\theta(s) \nabla_a Q_\pi(s, a)|_{a=\pi(s)}]. \quad (2)$$

The challenge in performing offline RL comes from the fact that $\mathcal{D}$ is static and has limited coverage of $\mathcal{S}$ and $\mathcal{A}$. While an out-of-distribution state is not a problem during training as the state is always sampled from $\mathcal{D}$, the policy may select an out-of-distribution action that is not contained in $\mathcal{D}$. This tends to lead to arbitrarily high estimates which further encourages the policy to take out-of-distribution actions. There are two main methods to counteract this: 1) constraining the policy to stay within the support of the dataset (Wu et al., 2019; Jaques et al., 2019; Fujimoto et al., 2019; Zhou et al., 2020), and 2) modifying the critic to better handle out-of-distribution actions (Kumar et al., 2019, 2020). In this work, we focus on the former.

3.2 Dialogue Policy in the Latent Action Space

RL can be applied to a dialogue system policy at different levels of abstraction. Semantic actions, i.e., tuples containing intent, slot and values, such as `inform(area=centre)`, are widely used for a compact and well-defined action space (Geishauser et al., 2021; Tseng et al., 2021). Predefining the actions and labeling the dialogue data however requires considerable labor. In addition, the final policy needs to be evaluated dependent on an NLG module. On the opposite end, natural language actions view each word of the entire system vocabulary as an action in a sequential decision making process (Mehri et al., 2019; Jaques et al., 2019). This blows up the action space size and the trajectory length, hindering effective learning and optimal convergence.

Zhao et al. (2019) proposed instead an automatically inferred latent space to serve as action space of the dialogue policy, where a latent action is a real-valued vector containing latent meaning. This decouples action selection and language generation, as well as shorten the dialogue trajectory. Lubis et al. (2020) followed up this work by proposing the use of variational auto-encoding (VAE) for a latent-space that is action characterized. In both of these works, the latent space is trained via supervised learning (SL) on the response generation task, and then followed with policy gradient RL using the corpus-based success as the reward signal, i.e.,

$$\nabla_\theta J(\theta) = \mathbb{E}_\theta[\sum_{t=0}^{T} R_t \nabla_\theta \log p_\theta(z_t|c_t)]. \quad (3)$$

3.3 Offline RL for Policy in the Latent Action Space (PLAS)

A latent action space also lends itself well to offline RL with a policy-constraint technique. Zhou et al. (2020) proposed to use a conditional VAE (CVAE) to model the behavior policy $\pi_\beta(a|s)$ to reconstruct actions conditioned on states. The benefit of learning in the latent space is that the latent policy has the flexibility of choosing the shape of the distribution via the prior. By constraining the latent policy to output latent actions with high probability under the prior, the decoder will output an action that is likely under the behavior policy in expectation. By choosing a simple prior such as a normal Gaussian distribution, constraint to the latent policy becomes simple to enforce, for example by defining $z = \pi(s)$ such that $z_i \in [-\sigma, \sigma]$ for each dimension i of the latent space for some hyperparameter

σ. PLAS defines a deterministic policy with continuous latent action that is optimized using the deterministic policy gradient method (Silver et al., 2014). Dual critics are used that are optimized with soft clipped double Q-learning. The PLAS algorithm has been applied to real robot experiment as well as locomotive simulations tasks. In this environment, the latent actions and action space are continuous. This differs quite considerably from dialogue systems, where the latent action needs to be translated to word-level actions which are discrete.

4 Offline Critic for Dialogue Policy Evaluation and Optimization

The architecture of our proposed critic is depicted in Figure 1(b). We utilize recurrency to let the critic take dialogue context into account. We encode the word-level user utterance with an RNN and concatenate it with the binary belief state to obtain s_t. On the other hand, the critic has the flexibility of taking any form of action. With latent actions, the action can be used as input directly by concatenating it with the state. When word-level or semantic actions are considered, a separate encoder can be used before concatenating it with the state.

In addition, to leverage the available data as much as possible, we incorporate the user goal for estimating the return. The MDP then becomes the dynamic parameter MDP (DP-MDP) as described by Xie et al. (2020), where a set of task parameters $g \in \mathcal{G}$ governs the state dynamics $p(s_{t+1}|s_t, a_t; g)$ and reward function $r(s_t, a_t; g)$. It is safe to incorporate the user goal for learning, because the critic is only used for policy evaluation and not needed to run the policy. If the user goal is not given in the data, it can be automatically derived from the dialogue state. To maintain the correctness of the dialogue context, when predicting $Q(s_t, a_t)$, all actions $a_{<t}$ are taken from the corpus. Only a_t is taken from the output of the policy. This is in contrast to the existing corpus-based success rate computation, where all $a_{\leq t}$ are taken from the policy and thus create context mismatches.

To keep the critic pessimistic in the face of uncertainty, we implement a dropout layer and do K forward passes for each state-action pair and the lowest value is then taken as the final prediction, i.e., $Q(s_t, a_t) = \min_{k=1}^{K} Q_k$. In this way, prediction with high variance, i.e., high uncertainty, is punished by taking the lower bound. This mechanism replaces the use of double critic in PLAS.

4.1 Offline Critic for Optimization: LAVA+PLAS

We combine LAVA (Lubis et al., 2020) and PLAS (Zhou et al., 2020) approaches in order to train a dialogue policy with latent action via offline RL. We use the multi-task LAVA approach, i.e., LAVA_mt, depicted in Figure 1(a), using continuous latent variables modeled via Gaussian distributions, as the normal distribution prior works best with the PLAS approach. In the original LAVA_mt, the model utilizes response generation (RG) and response VAE objectives for optimization with a 10:1 ratio, i.e., the VAE objective is optimized once every 10th RG epoch. In other words, the VAE is only used as an auxiliary task to ground the latent space from time to time. In this work, we modify the model training to preserve both RG and VAE abilities equally, as we will need the VAE to retrieve the latent action from the dataset $\mathcal{D}$.

With θ as state encoder parameters, ϕ action encoder, and ω decoder, for each training pass, both tasks are performed and the model uses their joint loss to update its parameters, i.e.,

$$\begin{aligned}&\mathcal{L}_{\text{LAVA\_mt}}(\omega, \theta, \phi) = \\ &\mathbb{E}_{p_\theta(z|s)}[\log p_\omega(x|z)] - \alpha \mathrm{D}_{\mathrm{KL}}[p_\theta(z|s)||p(z)] \\ &+ \mathbb{E}_{q_\phi(z|r)}[\log p_\omega(x|z)] - \beta \mathrm{D}_{\mathrm{KL}}[q_\phi(z|r)||p(z)].\end{aligned} \tag{4}$$

While the original LAVA uses policy gradient RL with the corpus-based success rate, in this work we follow the SL with PLAS algorithm. Parts of the LAVA_mt model are used to initialize the actor and critic networks: parameters θ are used for the actor, ϕ to retrieve the latent action z given a word-level response a, and the decoder ω to map latent actions produced by the actor into word-level responses. Prior to PLAS training, we warm-up the LAVA_mt model with only the VAE objective to further improve the latent action reconstruction capability:

$$\begin{aligned}&\mathcal{L}^{\text{VAE}}_{\text{LAVA\_mt}}(\omega, \phi) = \mathbb{E}_{q_\phi(z|r)}[\log p_\omega(x|z)] - \\ &\beta \mathrm{D}_{\mathrm{KL}}[q_\phi(z|r)||p(z)].\end{aligned} \tag{5}$$

PLAS training is depicted in Figure 1(c). It consists of two interleaved training loops. For each pass, an episode is sampled from the static dataset $\mathcal{D}$. In the actor training loop, the actor parameter is optimized using deterministic policy gradient (Silver et al., 2014) to maximize the critic estimate. Due to the deterministic nature of the policy,

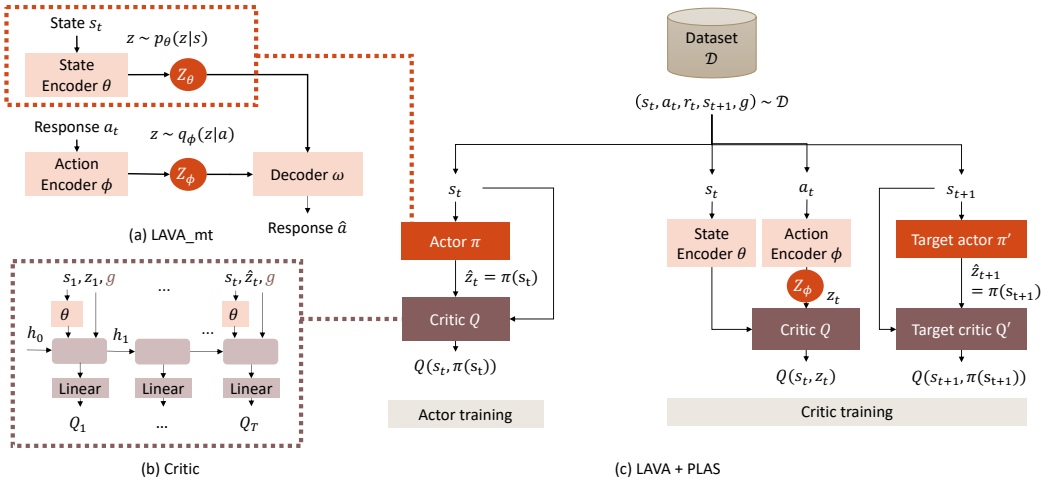


Figure 1: Overview of LAVA_mt, critic network and offline RL with PLAS. First, (a) we pre-train LAVA_mt with modified shared objective. The state encoder and latent space of the resulting model is used to initialize the actor for PLAS. The critic (b) is an RNN-based model that takes state, action and user goal to estimate the return. PLAS samples the transition from the static dataset and uses it to train actor and critic in an alternating fashion. To compute the target Q-value $Q(s_{t+1}, \pi(s_{t+1}))$, target actor and critic networks are used with soft update to improve stability.

the actor no longer samples from the distribution, but instead takes the distribution mean as the action. To encourage the policy to stay close to the behavior policy, as an additional loss, we add a mean-squared error (MSE) term between the chosen action $\hat{z}_t = \pi(s)$ and the reconstructed action from the corpus z_t. The actor loss is defined as

$$\mathcal{L}_{\text{actor}} = Q(s, \pi(s)) + \text{MSE}(\hat{z}_t, z_t). \quad (6)$$

On the other hand, the critic is trained to minimize the error of the Bellman equation. In addition, we penalize the critic with a weighted KL loss term as a means of regularization when the target actor chooses an action that is far from the behavior policy. The critic loss is defined as

$$\mathcal{L}_{\text{critic}} = (Q(s_t, a_t) - (r_t + \gamma Q'(s_{t+1}, \pi'(s_{t+1})))^2 - \lambda D_{KL}(q_\phi(z_{t+1}|a_{t+1}) || \pi'(s_{t+1})). \quad (7)$$

As is common practice, we use the target critic and actor networks for computing the target Q-value. The actor, critic, and their corresponding target networks are initialized the same way, but the target networks are updated with a soft update to promote stability in training.

4.2 Offline Critic for Evaluation

In this paper, we utilize offline RL critic in a new way, as a data- and model-independent evaluator for task-oriented dialogue systems. Following the critic training loop in Figure 1(c), we replace the target actor with the fixed policy π_e, i.e. the one to be evaluated, and perform the critic loop training with Equation 7 as the loss function, setting $\lambda = 0$ for systems with word-level action.

Note that with this approach, the dataset consisting of N dialogues $\mathcal{D} = \{\{(s_i, a_i, s_{i+1}, r_i)\}_{i=1}^{T^n}\}_{n=1}^{N}$ for evaluation can take any form as long as the states s_i and actions a_i are compatible with the dialogue system input and output, allowing comparisons across various types of dialogues systems. For instance, the states s_i can be represented as sequences of utterances or binary vectors and actions a_i as word-level, latent, semantic, or binary actions. In terms of rewards, those can be sparse (i.e. intermediate rewards are set to 0, $r_i = 0$, $i < T_n$, $n = 1, \ldots, N$) and in case that the corpus represents the desireable behaviour, a maximum reward can be assumed as a final reward for every dialogue in the corpus (i.e. set to 1, $r_{T_n} = 1$, $n = 1, \ldots, N$). Of course more accurate reward labels would result in an even more precise evaluator. As a consequence, dialogue systems can be evaluated on static corpora that differ from the training corpus and also not necessarily generated by interacting with the system.

A possible use case scenario would be a human-human corpus annotated with states and sparse rewards and a number of different dialogue systems being evaluated on this corpus. This is the case we consider in our evaluation below, whereby we use word-level and latent actions, and thus do not require explicit action labels.

5 Experimental Set-up

5.1 Data

We use MultiWOZ 2.1 (Budzianowski et al., 2018; Eric et al., 2019) to conduct our experiments, one of the most challenging and largest corpora of its kind. MultiWOZ is a collection of conversations between humans in a Wizard-of-Oz fashion, where one person plays the role of a dialogue system and the other one a user. The user is tasked to find entities, e.g., a restaurant or a hotel, that fit certain criteria by interacting with the dialogue system. The corpus simulates a multi-domain task-oriented dialogue system interaction.We use the training, validation and test set partitions provided in the corpus, amounting to 8438 dialogues for training and 1000 each for validation and testing.

5.2 Policy and Critic Training

For the LAVA_mt pre-training, we use simple recurrent models as encoder and decoder and follow the hyperparameters as set in the original work (Lubis et al., 2020) with a few exceptions, i.e. we use 200-dimensional continuous latent variables with a normal Gaussian as the prior and we lower the learning rate to $5e{-}4$. As depicted in Figure 1, parts of the LAVA_mt model are then used by the actor, critic, and different parts of PLAS training. For the critic, we set the hidden size to be 500 and the linear layer to use the sigmoid activation function. During PLAS, we use a learning rate of 0.01 for the critic and 0.005 for the actor. The critic dropout rate and λ are set to 0.3 and 0.1, respectively. The policy is trained with a maximum of 10K sampled episodes from the corpus, and the best checkpoint is chosen according to the corpus-based success rate. We set the hyper-parameters of the critic as an offline evaluator the same way, except that it uses 100K sampled episodes for training without early stopping.

5.3 Dialogue Systems

To show the generalization ability of our proposed offline evaluation, we evaluate various dialogue systems that differ in terms of modular abstractions and architectures:

HDSA (Chen et al., 2019) is a transformer-based dialogue generation architecture with graph-based dialogue action using hierarchically-disentangled self-attention (HDSA). The model consists of a predictor, which outputs the dialogue action, and a generator, which subsequently maps it into dialogue response. Two versions of HDSA are included, one which uses ground-truth action for generation (gold), and one which uses predicted labels (pred). Note that the 'pred' version is the only one that can be deployed in an interactive set-up.

AuGPT (Kulhánek et al., 2021) is a fully end-to-end dialogue system with fine-tuned GPT2 (Radford et al., 2019) on multi-task objectives, including belief state prediction, response prediction, belief-response consistency, user intent prediction, and system action prediction. The model is trained on MultiWOZ data augmented with the Taskmaster-1 (Byrne et al., 2019) and Schema-Guided Dialogue (Rastogi et al., 2020) datasets.

LAVA (Lubis et al., 2020) is an RNN-based model using latent actions, optimized via SL and policy gradient RL with corpus-based success rate as reward. We use LAVA_kl as the best performing model reported.

LAVA+PLAS (Ours) is our proposed variant of LAVA that is trained in an offline RL set-up using offline critic and PLAS algorithm (Section 4.1).

5.4 Evaluation Metrics

Offline Critic for Evaluation (Ours) For each system, we train an offline critic using offline Q-learning as described in Section 4.2. While theoretically the critic can take any form of dialogue action as input, in our experiments we utilize word-level or latent action. We consider intermediate rewards to be 0 and the final reward is 1 for a successful dialogue or 0 for a failed dialogue, as provided in the MultiWOZ corpus. As final estimated value of the policy, we report the average estimated return of all initial states on the test set.

Standard corpus-based metrics Corpus based evaluation is conducted on MultiWoZ test set using delexicalized responses with the benchmarking evaluation script provided by Budzianowski et al. (2018). A pseudo dialogue is generated, where user turns are taken from the corpus and system turns are generated by the evaluated model. Match rate computes whether all informable slots in the user goal are generated, and success rate computes whether all information requested by the user is provided. For completeness, we also report the BLEU score on target responses.

		SL	SL + PLAS
Corpus	Match	66.06	83.94
	Success	51.95	67.54
	BLEU	0.17	0.14
ConvLab US	Compl.	37.42	47.02
	Success	31.87	39.40
	Book	19.12	36.74
	F1	49.11	57.14
	Turns	21.57	21.99

Table 1: Offline RL in latent space improves task-related metrics on both corpus and US evaluations. Results are averaged across 5 seeds.

US evaluation We use the default ConvLab2 (Zhu et al., 2020) user simulator with the BERT-based NLU module, rule-based agenda policy and template NLG. We conducted 1000 dialogues and report the average number of turns across all dialogues. We focus on three measures: book rate, i.e., how often the system finalized a booking, success rate, i.e., the percentage of dialogues where all information requested by the user is provided by the system and bookings are successfully made, and lastly complete rate, i.e., the number of dialogues that are finished regardless of whether the booked entity matches the user criteria. We also report entity F1 and average number of turns across the simulated dialogues.

With the exception of AuGPT, the systems' dialogue policies require a dialogue state tracker (DST) for online interactions. For this purpose, we utilize a tracker with a joint goal accuracy of 52.26% on the test set of MultiWOZ 2.1 (van Niekerk et al., 2020). This tracker is a recurrent neural model, which utilises attention and transformer based embeddings to extract important information from the dialogue. We perform lexicalization via handcrafted rules using the information from the dialogue state and database query. For handling incomplete lexicalizations due to empty database queries or a wrongly predicted domain by the policy, we replace the response with a generic "I'm sorry, could you say that again?". This is equal to masking such actions while neither punishing nor rewarding the policy.

Human evaluation Human evaluation is performed via DialCrowd (Lee et al., 2018) connected to Amazon Mechanical Turk. The systems are set up identically as in the US evaluation, except that the systems are interacting with paid users instead of a US. Users are provided with a randomly generated user goal and are required to interact with our systems in natural language and to subsequently evaluate them. We ask the user whether their goal is fulfilled through the dialogue, indicating the success rate. We also ask them to rate the overall system performance on a Likert scale from 1 (worst) to 5 (best). For each system we collected 400 dialogues with human workers.

6 Results and Analysis

6.1 Offline Critic for Optimization

Table 1 shows the policy performance after shared multi-task SL training and the performance after subsequent offline RL training with PLAS, averaged over 5 seeds. We observe that offline RL in latent space with the critic estimate as reward signal improves task-related metrics on both corpus and US evaluation. The consistent improvement on offline and interactive evaluations is the result of critic's value estimate as reward signal, which we believe is noteworthy as the policy is never explicitly trained on either metric.

Like policy gradient RL used by LAVA (Equation 3), PLAS leads to a decrease in BLEU score. This is quite common for end-to-end policies trained with RL following SL (Lubis et al., 2020), however the decrease with PLAS is not as drastic. This signals that the policy retains more linguistic variety in the responses, since the reward signal does not overlook context mismatch and thus responses that are out of context are not rewarded. We include a dialogue example in Appendix A to demonstrate the context mismatch issue and how the offline critic addresses it.

6.2 Offline Critic for Evaluation

System performances across metrics Tables 2 and 3 present the corpus- and interaction-based evaluation results of LAVA+PLAS and our baselines. For completeness, we included the human policy, i.e., the behavior policy of the dataset, on the corpus-based evaluation. For LAVA+PLAS, we pick the best model out of the 5 seeds. For the baseline models, we utilize the released pre-trained parameters and re-run all evaluations.

The ranking of the systems differs depending on the evaluation metrics. With corpus-based success and match rates, LAVA far outperforms the other models and even human wizards. This is expected, as LAVA_kl is directly optimized with the corpus-based success rate as reward. In terms of BLEU, HDSA – which is designed for genera-

Policy	Corpus Evaluation			Critic Evaluation
	Match	Success	BLEU	
MultiWOZ (Human)	90.40 ± 1.82	82.30 ± 2.36	N/A	52.68 ± 0.02
AuGPT	83.30 ± 2.31	67.20 ± 2.91	0.17	52.45 ± 0.02
LAVA+PLAS	88.30 ± 1.99	73.40 ± 2.74	0.14	51.76 ± 0.03
LAVA_kl	97.50 ± 1.14	94.80 ± 1.47	0.12	48.95 ± 0.08
HDSA (gold)	91.80 ± 1.70	82.50 ± 2.35	0.21	49.89 ± 0.08
HDSA (pred)	88.90 ± 1.95	74.50 ± 2.70	0.20	49.00 ± 0.09

Table 2: Corpus-based evaluation metrics. 95% confidence intervals are reported.

Policy	ConvLab US Evaluation					Human Evaluation	
	Compl.	Success	Book	F1	Avg. turn	Success	Rating
AuGPT	89.20 ± 1.92	83.30 ± 2.31	85.16 ± 3.34	81.03 ± 1.40	14.50 ± 0.41	90.75 ± 2.85	4.34 ± 0.08
LAVA+PLAS	54.20 ± 3.09	45.30 ± 3.09	61.18 ± 4.51	58.85 ± 2.25	23.54 ± 0.89	63.00 ± 4.75	3.34 ± 0.12
LAVA_kl	49.20 ± 3.10	40.00 ± 3.04	63.20 ± 4.37	54.47 ± 2.24	26.64 ± 1.00	63.25 ± 4.74	3.44 ± 0.12
HDSA (pred)	36.70 ± 2.99	25.90 ± 2.71	6.67 ± 2.37	49.97 ± 2.23	31.32 ± 0.86	55.25 ± 4.89	3.09 ± 0.12

Table 3: Interactive evaluation metrics. 95% confidence intervals are reported.

Fleiss' Kappa			Human Evaluation	
			Success	Rating
Corpus-based	Corpus	Match	-0.623	-0.571
		Success	-0.460	-0.397
		BLEU	0.343	0.299
	Critic		**0.755**	**0.713**
Interactive	US	Complete	0.992	0.984
		Success	0.991	0.984
		Book	0.789	0.802
		F1	0.990	0.978
		Turn	-0.967	-0.956

Table 4: Correlation between evaluation metrics and human judgements. Absolute values shows the strength of the correlation. Negative sign shows inverse correlation.

tion with semantic action – achieves the first rank. With critic evaluation, human policy achieves the highest score. The rankings for evaluation with user simulator and paid workers in Table 3 are consistent, showing another trend entirely. AuGPT outperforms the other systems with a huge margin, LAVA+PLAS and LAVA_kl show a narrower gap in performance compared to corpus-based metrics, while HDSA performs very poorly. The collected dialogues show that the language understanding and generation of AuGPT is superior to the other models, as it leverages a large pre-trained model as a base model and utilizes multiple dialogue corpora for fine-tuning. In other words, it is trained on orders of magnitude more data compared to the other systems. This results in a more natural interaction with both simulated and human users.

It is interesting to note that the critic has a much narrower confidence interval compared to the other metrics. Although the values for some policies are seemingly close, the intervals show that the difference between most of the systems are statistically significant, except for LAVA_kl and HDSA (gold).

Correlation with human judgements Table 4 lists pairwise correlation between human judgements and the automatic metrics. We differentiate between corpus-based metrics such as the standard match and success rates, BLEU and critic evaluation, with interactive metrics that require a form of user, either simulated or paid. Success rates of current standard evaluations have moderate inverse correlation with human judgements due to the context mismatch that occurs during its computation. On the other hand, the theoretically grounded value estimation by the offline critic has a strong correlation with human judgements, showing that our proposed method is a more suitable corpus-based metric to reflect the dialogue system performance. Our study confirms the weak correlation between BLEU and human ratings. All metrics computed based on interaction with US are strongly correlated with metrics from human evaluation. The number of turns is strongly but inversely correlated, which aligns with the intuition that the fewer turns the system needs to complete the dialogue, the better it is perceived by human users. This suggests that while existing US is far from fully imitating human behavior, it provides a good approximation to how the systems will perform when interacting with human users. We advocate that future works report on multiple evaluation metrics to provide a more complete picture of the dialogue system performance.

Note that while US evaluation provides stronger

correlations with human judgements, our proposed use of offline RL critic for evaluation has the benefit of being corpus- and model-independent, whereas for a new corpus and ontology, a new US would need to be designed and developed. Furthermore, an offline evaluation takes significantly less time to perform, making it an efficient choice for the iterative development process.

6.3 Impact of Reward Signal on RL

LAVA+PLAS and LAVA_kl are the only two systems optimized via RL. We observe that they significantly outperform the other on the respective metric they received as reward signal during RL. However, when subjected to interactive evaluation, the gap between their performance is shrinking (see Table 3). This shows on the one hand the power of reinforcement learning methods to optimize the given reward and on the other hand how important it is to define this reward correctly, warranting further research in both extrinsic and intrinsic reward modelling for dialogue (Wesselmann et al., 2019; Geishauser et al., 2021).

7 Conclusion

We propose the use of offline RL for dialogue evaluation based on static corpus. While offline RL critics are typically utilized for policy optimization, we show that they can be trained for any dialogue system as external evaluators that are corpus- and model-independent, while attaining strong correlation with human judgements, which we confirm via an interactive user trial. Not only does the offline RL critic provide a corpus-based metric that is reliable and efficient to compute, it also addresses a number of issues highlighted in the recently published NSF report (Mehri et al., 2022). It is important to note that the proposed framework does not depend on the definition of states, action and rewards. So in principle, one could apply this method beyond task-oriented dialogue systems. For example, one could evaluate a number of chat-bots considering a corpus annotated only with level of engagement achieved in each dialogue and thus measure the level of engagement of the evaluated chat-bots.

Acknowledgements

N. Lubis, C. van Niekerk, M. Heck and S. Feng are supported by funding provided by the Alexander von Humboldt Foundation in the framework of the Sofja Kovalevskaja Award endowed by the Federal Ministry of Education and Research. C. Geishauser and H-C. Lin are supported by funds from the European Research Council (ERC) provided under the Horizon 2020 research and innovation programme (Grant agreement No. STG2018 804636). Google Cloud and HHU ZIM provided computational infrastructure.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Ultes Stefan, Ramadan Osman, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709.

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3806–3812, Minneapolis, Minnesota. Association for Computational Linguistics.

Mihail Eric, Rahul Goel, Shachi Paul, Adarsh Kumar, Abhishek Sethi, Peter Ku, Anuj Kumar Goyal, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tur. 2019. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. *arXiv preprint arXiv:1907.01669*.

Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR.

Raefer Gabriel, Yang Liu, Anna Gottardi, Mihail Eric, Anju Khatri, Anjali Chadha, Qinlang Chen, Behnam Hedayatnia, Pankaj Rajan, Ali Binici, et al. 2020. Further advances in open domain dialog systems in the third Alexa prize socialbot grand challenge. *Alexa Prize Proceedings*.

Christian Geishauser, Songbo Hu, Hsien-Chin Lin, Nurul Lubis, Michael Heck, Shutong Feng, Carel van Niekerk, and Milica Gasic. 2021. What does the user want? information gain for hierarchical dialogue policy optimisation. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pages 969–976. IEEE.

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, et al. 2020. Overview of the ninth dialog system technology challenge: Dstc9. *arXiv preprint arXiv:2011.06486*.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. 2021. AuGPT: Dialogue with pre-trained language models and data augmentation. *arXiv preprint arXiv:2102.05126*.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191.

Kyusong Lee, Tiancheng Zhao, Alan W Black, and Maxine Eskenazi. 2018. DialCrowd: A toolkit for easy dialog system assessment. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 245–248.

Sungjin Lee and Maxine Eskénazi. 2012. POMDP-based let's go system for spoken dialog challenge. In *Proceedings of SLT*.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *ICLR (Poster)*.

Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gasic. 2021. Domain-independent user simulation with transformers for task-oriented dialogue systems. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 445–456, Singapore and Online. Association for Computational Linguistics.

Chia-Wei Liu, Ryan Lowe, Iulian Vlad Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132.

Nurul Lubis, Christian Geishauser, Michael Heck, Hsien-chin Lin, Marco Moresi, Carel van Niekerk, and Milica Gašic. 2020. LAVA: Latent action spaces via variational auto-encoding for dialogue policy optimization. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 465–479.

Shikib Mehri, Jinho Choi, Luis Fernando D'Haro, Jan Deriu, Maxine Eskenazi, Milica Gasic, Kallirroi Georgila, Dilek Hakkani-Tur, Zekang Li, Verena Rieser, et al. 2022. Report from the NSF future directions workshop on automatic evaluation of dialog: Research directions and challenges. *arXiv preprint arXiv:2203.10012*.

Shikib Mehri and Maxine Eskenazi. 2020a. Unsupervised evaluation of interactive dialog with dialoGPT. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 225–235.

Shikib Mehri and Maxine Eskenazi. 2020b. USR: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 681–707.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177.

Tomáš Nekvinda and Ondřej Dušek. 2021. Shades of BLEU, flavours of success: The case of MultiWOZ. *arXiv preprint arXiv:2106.05555*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Govardana Sachithanandam Ramachandran, Kazuma Hashimoto, and Caiming Xiong. 2021. Causal-aware safe policy improvement for task-oriented dialogue. *arXiv preprint arXiv:2103.06370*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Jost Schatzmann. 2008. *Statistical User and Error Modelling for Spoken Dialogue Systems*. Ph.D. thesis, University of Cambridge.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR.

Amanda Stent, Matthew Marge, and Mohit Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *international conference on intelligent text processing and computational linguistics*, pages 341–351. Springer.

Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne. 2021. Transferable dialogue systems and user simulators. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 152–166.

Stefan Ultes, Paweł Budzianowski, Inigo Casanueva, Nikola Mrkšić, Lina Maria Rojas-Barahona, Pei-Hao Su, Tsung-Hsien Wen, Milica Gašić, and Steve J Young. 2017. Domain-independent user satisfaction reward estimation for dialogue policy learning. In *INTERSPEECH*, pages 1721–1725.

Carel van Niekerk, Michael Heck, Christian Geishauser, Hsien-Chin Lin, Nurul Lubis, Marco Moresi, and Milica Gasic. 2020. Knowing what you know: Calibrating dialogue belief state distributions via ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3096–3102.

Siddharth Verma, Justin Fu, Mengjiao Yang, and Sergey Levine. 2022. Chai: A chatbot ai for task-oriented dialogue with offline reinforcement learning. *arXiv preprint arXiv:2204.08426*.

Marilyn Walker, Diane Litman, Candace A Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 271–280.

Paula Wesselmann, Yen-Chen Wu, and Milica Gašić. 2019. Curiosity-driven reinforcement learning for dialogue management. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7210–7214.

Yifan Wu, George Tucker, and Ofir Nachum. 2019. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*.

Annie Xie, James Harrison, and Chelsea Finn. 2020. Deep reinforcement learning amidst lifelong non-stationarity. *arXiv preprint arXiv:2006.10701*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218.

Wenxuan Zhou, Sujay Bajracharya, and David Held. 2020. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. ConvLab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149.

A Example Dialogue

Table 5 shows an example dialogue accompanied by the corresponding offline critic Q estimation and corpus-based success rate.

	Utterances	Q	Corpus succ.
User	can you help me find a [value_pricerange] place to stay in the [value_area] part of town?		
True	sure. there are [value_count] guesthouses there. i would be happy to book [value_count] for you if you like.		
Pred	the [hotel_name] is located at [hotel_address] , postcode [hotel_postcode]. the phone number is [hotel_phone].	0.4730	
User	does it have a star rating of [value_count]?		
True	no. they all have [value_count] stars.		
Pred	*the [hotel_name] is a guesthouse in the [value_area] area. the phone number is [hotel_phone] and the postcode is [hotel_postcode].*	0.4569	
User	then find me [value_count] in the [value_pricerange] price range.		
True	all of the accommodations in the [value_area] are in the [value_pricerange] price range. would you like to try a different part of town?		
Pred	*the [hotel_name] is located at [hotel_address] , postcode [hotel_postcode]. the phone number is [hotel_phone].*	0.4478	
User	find me a nice [value_count] and book for [value_count] people and [value_count] nights from [value_day]		
True	actually , the [hotel_name] is a [value_count] start hotel in the [value_area] area. it s in the [value_pricerange] price range. how does that sound?		
Pred	*i have booked your room at the [hotel_name] for [value_count] nights starting [value_day]. your reference number is [hotel_reference]. is there anything else i can help you with?*	0.4461	
User	that sounds perfect. can you reserve that and provide me with the reference number?		
True	okay , you are all set. the hotel is booked and the reference number is [hotel_reference]. can i help you with anything else today?		
Pred	the reference number is [hotel_reference].	0.4205	
User	thank you. i am also looking for a restaurant that serves [value_food] and that is in the same area as the hotel		
True	[restaurant_name] is an [value_pricerange] restaurant in the same area as the hotel , would you like me to make reservations for you?		
Pred	[restaurant_name] is located at [restaurant_address] , [restaurant_postcode]. the phone number is [restaurant_phone].	0.4167	
User	yes. same group of people , at [value_time] on the same day please.		
True	booking was successful. the table will be reserved for [value_count] minutes. reference number is [restaurant_reference]. what else can i help you with today?		
Pred	i have booked you a table at [restaurant_name]. the reference number is [restaurant_reference]. can i help you with anything else?	0.4095	
User	that was all i needed. thanks. bye.		
True	ok! have a great day!		
Pred	you are welcome. have a great day!	0.4161	1.0

Table 5: Example dialogue comparing offline critic estimate and corpus-based success. "True" denotes responses taken from the corpus, and "Pred" responses from the policy, in this case we use LAVA_kl with which context mismatch often occurs. Note that Q prediction takes "User" and "True" utterances from the beginning up to the previous turn, and "User" and "Pred" of current turn. On the other hand, Corpus-based success takes on "User" and "Pred" utterances for all turns. Predicted responses in italic highlight the context mismatch that can occur when pseudo-dialogue is constructed for dialogue success computation. This is however ignored and the dialogue is considered successful, since all necessary requestable slots are generated by the system. On the other hand, the Q-estimate shows a decrease in value, and the policy is given a lower reward signal for the same dialogue.

Disruptive Talk Detection in Multi-Party Dialogue within Collaborative Learning Environments with a Regularized User-Aware Network

Kyungjin Park[1], Hyunwoo Sohn[1], Wookhee Min[1], Bradford Mott[1], Krista Glazewski[2], Cindy E. Hmelo-Silver[2], and James Lester[1]
[1]Department of Computer Science, North Carolina State University
[2]Center for Research on Learning and Teaching, Indiana University Bloomington
[1]{kpark8, hsohn3, wmin, bwmott, lester}@ncsu.edu
[2]{glaze, chmelosi}@indiana.edu

Abstract

Accurate detection and appropriate handling of disruptive talk in multi-party dialogue is essential for users to achieve shared goals. In collaborative game-based learning environments, detecting and attending to disruptive talk holds significant potential since it can cause distraction and produce negative learning experiences for students. We present a novel attention-based user-aware neural architecture for disruptive talk detection that uses a sequence dropout-based regularization mechanism. The disruptive talk detection models are evaluated with multi-party dialogue collected from 72 middle school students who interacted with a collaborative game-based learning environment. Our proposed disruptive talk detection model significantly outperforms competitive baseline approaches and shows significant potential for helping to support effective collaborative learning experiences.

1 Introduction

Automatic analysis of dyadic dialogue utilizes a broad range of methods for intent recognition (Ahmadvand et al., 2019; Grau et al., 2004; Kim et al., 2010; Maraev et al., 2021). Compared to dyadic conversations, multi-party conversations are characterized by a high degree of complexity due to multi-way group interactions, thus, multi-party dialogue models should take into account group dynamics to reliably model phenomena. For example, previous research investigated giving less weight to participants whose convergence behaviors differ from the rest of the group (Rahimi and Litman, 2018) to examine which utterances should be clustered together (i.e., conversation threads) in multi-party dialogues (Mayfield et al., 2012; Tan et al., 2019).

In education, computer-supported collaborative learning environments promote social aspects of learning through the use of a variety of technological and constructive pedagogical strategies, including problem-based learning and inquiry learning (Dillenbourg et al., 2009; Hmelo-Silver, 2004; Jeong et al., 2019). Collaborative game-based learning environments often provide students with in-game chat features to help promote open discussion and negotiation among team members, facilitating the coordination of their in-game learning activities (Saleh et al., 2021). However, students are not always effective collaborators and may engage in improper communicative behavior, distracting from the group learning experience. The presence of negative socio-emotional engagement in collaborative learning environments can result in disruptive talk and can function as a barrier to the development of high-quality collaborative communication.

Previous work on detecting talk that can cause negative socio-emotional engagement (e.g., off-task behavior, bullying, disruptive talk) in collaborative learning environments investigated computational approaches using language models ranging from classic approaches (e.g., *n*-grams) and word embedding approaches (e.g., BERT). These language models have been combined with classic techniques (e.g., logistic regression, random forest) and deep learning techniques (e.g., long-short term memory networks) (Carpenter et al., 2020; Nikiforos et al., 2020; Park et al., 2021). However, the previous work either makes utterance-by-utterance predictions without taking

Proceedings of the SIGdial 2022 Conference, pages 490–499
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

context into account or treats the entire multi-party conversation sequence as a continuous dialogue flow, despite the potential presence of multiple concurrent message threads with different topics.

In this paper, we propose a novel attention-based, regularized user-aware modeling approach for detecting disruptive talk in multi-party dialogue within a collaborative game-based learning environment. We investigate the use of target-user embeddings to help the prediction model determine the disruptiveness of the sequence more accurately with an additional user-specific network and attention mechanism. We also investigate a sequence-level dropout mechanism during training as a regularization technique that could help avoid overfitting possible diluted conversation sequences (i.e., presence of multiple threads in a sequence) in training data. Experimental results demonstrate that our attention-based, regularized user-aware model offers great potential for addressing disruptive talk detection in multi-party dialogues.

2 Related Work

Diverse prediction tasks have analyzed multi-party dialogue focusing on the asynchronous and entangled nature of group conversations, such as dialogue act classification using group thread history, and thread detection as well as cyberbullying and toxic message detection within group conversations (Anikina and Kruijff-Korbayova, 2019; Blackburn and Kwak, 2014; Ekiciler et al., 2021; Kim et al., 2012; Min et al., 2021; Tan et al., 2019).

Kim et al. (2012) investigated classic machine learning approaches for dialogue act classification, such as Naïve Bayes, support vector machines, and conditional random fields, along with contextual, structural, keyword, and dialogue interaction-based features of utterances for dialogue act classification in multi-party live chat datasets. As a sub-task of a disaster response mission knowledge extraction task, Anikina and Kruijff-Korbayova (2019) proposed a deep learning-based Divide&Merge architecture utilizing LSTM and CNN for predicting dialogue acts. Min et al., (2021) investigated the use of dialogue act prediction utilizing conditional random fields and ELMo contextualized word embeddings in multi-party team communication for providing adaptive team training support.

As multiple participants are involved in multi-party conversation, disentanglement of the conversation based on relevancy is another important task, which could enhance the conversational relevance rate of automated dialogue agents (Shamekhi et al., 2018) or improve summarization quality (Zhang and Cranshaw, 2018). Tan et al. (2019) proposed three LSTM-based context-aware thread detection architectures that automatically captures conversation threads in multi-party and multi-thread conversations, where the proposed model predicts which existing thread the current utterance belongs to (or whether it creates a new thread).

Another task that has received considerable attention in multi-party conversation is cyberbullying. The ability to detect bullying or toxic behavior is crucial to protecting users from cyberbullying. In particular, researchers are increasingly interested in toxic behavior in multiplayer games, such as multiplayer online battle arena (MOBA) games, where players compete against other teams in virtual online game environments (Kordyaka 2018). Blackburn and Kwak (2014) used random forest classifiers to detect toxic behavior in League of Legends using in-game performance, user reports, and chat data. The conversation data included 590,000 utterances, which were labeled via crowdsourcing on whether the conversation was toxic or not. Ekiciler et al. (2021) presented a linguistic analysis of gender-based toxic language usage in a Dota 2 chat dataset and investigated Naïve Bayes classifiers with three different Laplace smoothing parameters as an automatic approach for sexist toxic comment detection. A significant presence of gender discrimination in online games, mainly by young males and intense players, was revealed in their qualitative analysis.

Students' conversations can create disruption in collaborative learning environments, impeding collaborative learning processes. Recent research on bullying, off-task behavior, and disruptive talk in collaborative learning environments examined a range of word embedding techniques as well as a variety of classical machine learning and deep learning techniques (Carpenter et al., 2020; Nikiforos et al., 2020; Park et al., 2021). Nikiforos et al. (2020) explored the automatic detection of aggressive behavior (i.e., bullying) in two K-12 computer-supported collaborative learning environments. They used unigrams to represent words and examined machine learning approaches such as Naïve Bayes with Laplace smoothing,

decision tree classifiers, and feedforward neural networks. The prediction results suggest that approaches based on deep learning outperform other classical machine learning approaches. Carpenter et al. (2020) used dialogue analysis to identify if students' messages were on-task or off-task during collaborative game-based learning. To develop a model capable of reliably detecting off-task behavior, they investigated three different word embedding approaches (i.e., Word2Vec, ELMo, and BERT), various history lengths of previous utterances, and two deep learning and classical machine learning classifiers were trained on a feature set containing contextual information extracted from student chat messages. The empirical evaluations indicated that the LSTM-based off-task behavior detection model with BERT embeddings outperformed other baseline approaches. Park et al. (2021) presented an LSTM-based disruptive talk detection framework in a multi-party dialogue dataset from a collaborative game-based learning environment, utilizing features from chat messages, a range of linguistic features, gender, and pre-test scores. While this work has the potential to improve learning experiences by detecting disruptions within collaborative learning settings, they disregard the unique characteristics of multi-party dialogues. In our work, we improve predictive performance of disruptive talk detection models by incorporating an additional network that embeds the characteristics of the user of a target utterance and a sequence-level dropout mechanism.

3 Corpus

We next describe the collaborative game-based learning environment and its chat-interface, dataset collected from two field studies, and disruptive talk annotation process.

3.1 ECOJOURNEYS Collaborative Game-Based Learning Environment

ECOJOURNEYS is a collaborative game-based learning environment for middle school science education focused on ecosystems (Mott et al., 2019; Saleh et al., 2019) (Figure 1). Students visit a virtual island in the game-based learning environment and are tasked with determining what is causing a mysterious illness among the island's fish population. Students work in groups of four to solve the mystery within the game, where each student works on a different laptop and interacts

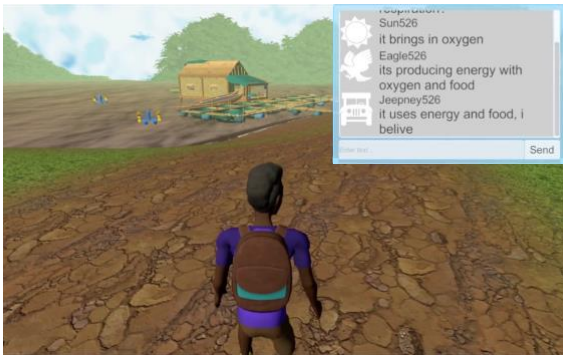


Figure 1: ECOJOURNEYS collaborative game-based learning environment and its in-game chat interface.

with peers in the virtual game environment. Individual students examine the fish illness during gameplay by collecting information and interacting with virtual characters. The virtual non-player characters serve as local experts, providing context for ecosystem concepts and the unfolding narrative (e.g., "Dissolved oxygen is a non-living component that animals and plants require to survive."). After investigating and gathering information, students meet at a virtual whiteboard within the game to share and categorize the information they have gathered and to discuss the most likely cause of the illness. Students are encouraged to exchange ideas, ask questions, and negotiate with their team members during the game's problem-solving activities using the in-game chat interface (Figure 1). This built-in chat system is accessible throughout the game. Each group is led by a facilitator, who is either a researcher or a teacher. The facilitator asks questions and encourages students to communicate with one another using the in-game chat interface. Facilitators can monitor and intervene on students' activities and conversations using an in-game screen, available only for facilitators, to guide students' learning. Facilitators can choose messages from a set of pre-written messages or write free-form messages using the in-game chat interface.

3.2 Dataset

The ECOJOURNEYS collaborative game-based learning environment was used in two classroom-based studies. Students were either in the sixth or seventh grade (11-13 years old) and played ECOJOURNEYS during six classroom periods. In total, 21 groups with 84 students (4 students per group) were involved in the two studies. From the 21 groups, the current work utilizes data from 18

groups consisting of 72 students (31 female and 41 male) who consented to the study and completed all the activities in the collaborative game-based learning environment. There are 9,236 chat messages available in the resulting dataset, with 2,440 messages from facilitators and 6,796 messages from students. We only consider the students' messages during the disruptive talk detection modeling working under the assumption that facilitators would not produce disruptive talk. On average, students in each group sent 382.4 messages (min = 89, max = 900, SD = 229.7).

3.3 Disruptive Talk Annotation

Adapted from prior work on disruptive talk analysis, the present work adopted a binary annotation scheme, *disruptive talk*, and *non-disruptive talk*, (Borge and Mercier, 2019). We labeled student utterances as disruptive talk if it had the potential to distract other group members from learning (e.g., "Um yea. yep, you can't work", "I WILL HAVE A MENTAL BREAKDOWN") and to interfere with deeper learning by interrupting the learning activity repeatedly (e.g., sending emojis multiple times). Otherwise, we labeled the utterance as non-disruptive talk.

Two human annotators labeled the students' chat-based dialogue collected during the study. Approximately 20% of the corpus was labeled by both annotators and an inter-rater agreement of 0.80 was achieved using Cohen's Kappa, indicating substantial agreement among the annotators (Cohen, 1960). All utterances labeled differently between the two annotators were discussed, and agreement was reached for certain situations without changing the high-level definition of disruptive talk we defined above. An example of those situations is when students exchange non-task-related messages, seemingly disruptive, before everyone is logged on and before starting the game, we agreed to label them as non-disruptive. A label was chosen for each utterance for which there was disagreement before proceeding with labeling the remainder of the corpus. Then, the remaining utterances were split in half and independently labeled by the annotators (approximately 40% each). The distribution of disruptive and non-disruptive utterances among the dataset was determined to be 1,864 (27.4%) and 4,932 (72.6%), respectively.

4 Method

4.1 Data Pre-Processing

The disruptive talk detection framework in our previous work utilizes linguistic features from student utterances and student attributes (i.e., gender and prior knowledge level) to determine how those features collectively contribute to prediction performance (Park et al., 2021). Here we keep all feature combinations from our previous work (i.e., sentence embedding, sentiment, Jaccard similarity between utterance and game text, gender, and pre-test scores) with an additional text cleaning pass that can be helpful for dealing with informal chat messages (Table 1).

We adopt a pre-trained BERT model, DistilBERT, a distilled version of BERT, that is a small, fast, and light Transformer model (Sanh et al., 2019). DistilBERT consists of 6 layers in the encoder with 40% fewer parameters than the BERT-base model and outputs 768-dimensional vectors for each word. We utilized a DistilBERT model that was trained on the Wikipedia dataset. For the sentence embedding, rather than taking the average of the embeddings of all the sentence words, we used the first token (i.e., [CLS]), a special token inserted in front of the input sentence in the BERT architecture, as it effectively represents what is in the input sentence and thus has been frequently used for BERT-based classification tasks (Devlin et al., 2018).

Approach	Example	Cleaned
Removed lengthening words[1]	"Hellllllllo"	"Hello"
Replaced slang[2]	"dis", "k"	"This", "Ok"
Spelling correction[3]	"who dat"	"Who that?"
Replaced abbreviated words	"don't", "can't"	"do not", "cannot"
Replaced emoji[4]	":-)"	"Happy"

Table 1: Text cleaning approaches.

[1] http://sentiment.christopherpotts.net/lexicons.html
[2] https://www.computerhope.com/jargon/c/chatslan.htm
[3] https://github.com/Azd325/gingerit
[4] https://pc.net/emoticons/

4.2 Attention-Based Regularized User-Aware Disruptive Talk Detection Modeling

When it comes to predicting disruptive talk based on the current message and a series of previous utterances, separately modeling the characteristics of the target user-specific utterances could be more effective than only utilizing messages from all group members equally; if a student makes a disruptive utterance, there is a higher chance that the same student will generate more disruption than the other group members. We propose an attention-based user-aware network that incorporates a target user-specific network that embeds the utterance histories of the target user as well as a separate network for modeling group-level utterances. We also apply the attention mechanism adapted from Bahdanau et al., (2014) to this output user representation and the hidden states of each time stamp to give weights to the group sequence output based on the user characteristics. An illustration of this attention-based user-aware network is shown in Figure 2

Suppose m_j^i is the feature embedding for the j^{th} message from user i in the n number of group utterance history, $Group_{Seq}$, including the current message.

$$Group_{Seq} = \{m_1^1, m_1^2, m_2^1, m_2^2, m_1^3, \cdots, m_j^i\}_{i=1,\cdots,4}$$

From this group sequence, we have user sequence $User_{seq}^i$ that only includes the utterances from user i.

$$User_{seq}^i = \{m_1^i, m_2^i, \cdots, m_j^i\}$$

The user network takes the utterance sequence, $User_{Seq}^i$, form the target-user only, then outputs a user embedding, $User_{emb}^i$.

$$User_{emb}^i = LSTM\left(User_{seq}^i\right)$$

We can get the attention score between the user embedding and the hidden state, h_t, at each LSTM time stamp of the group sequence.

$$\alpha_t = Softmax(User_{emb}^i \cdot h_t)_{t=1\ldots n}$$

Using this attention score, we can get the user-aware group sequence embedding.

$$Group_{emb} = \sum_{t=1}^{n} \alpha_t * h_t$$

Finally, we get the output probability by using a sigmoid function that takes as input $User_{emb}^i$ and $Group_{emb}$ via concatenation, then determine if the last utterance (e.g., m_j^2 in Figure 2) given to the model is an instance of disruptive talk with the threshold value of 0.5.

$$O = sigmoid(W_o * (User_{emb}^i + Group_{emb}))$$

We expect this attention-based user-aware approach will assist the model's inference on the target utterance by providing specific information about the target student characteristics embedded by the user-specific network, while simultaneously attending to the related utterances from the group sequence.

Additionally, we adopt a training approach that could better deal with the dynamics in multi-party dialogues. Different from dyad conversation, multiple conversation threads in the group conversation could make it difficult for the model to learn consistent and generalized aspects. We adopt a sequence dropout approach, which is one of the discourse perturbation methods used in (Koupaee et al., 2021), applied to the sequence inputs so that the model can learn different representations from the same context messages at every epoch as an approach to model regularization. For the given $Group_{Seq}$, we randomly drop utterances with a sequence dropout rate of *r*, range in (0, 0.5) excluding the target utterance. This sequence dropout rate can be fixed or can be randomly selected from the normal distribution. Figure 3 shows how this approach is applied during training.

We anticipate that by observing the same context message sequence from multiple dimensions, the disruptive talk detection model will learn generalized patterns by avoiding possible overfitting. Note that we do not drop anything from the user network with an assumption that utterances from the same user are consistent. It should be noted that this sequence dropout mechanism is different from the dropout technique commonly used for recurrent neural networks, which drops for the linear transformation of the inputs or the recurrent state, by dropping for the entire input at random time stamps. The sequence dropout is applied to the training data only for effective training through model regularization.

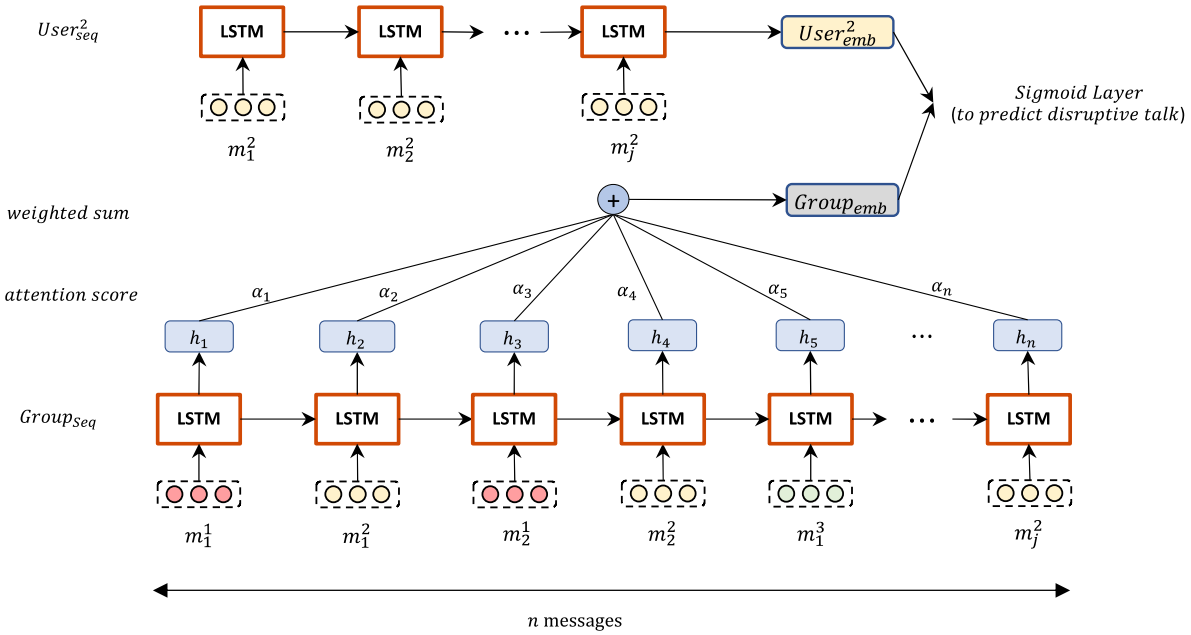


Figure 2: Proposed attention-based user-aware model. This figure illustrates what happens when the current message is from User 2.

4.3 Evaluation

We evaluate our modeling approaches in three steps. First, we compare our attention-based user-aware approach with our baseline model, which is based on a group sequence network with an attention mechanism (i.e., without the user-aware feature). This baseline modeling approach using LSTM-based disruptive talk detection model with DistillBERT as a sentence embedding approach, and 20 context messages, was adapted from our previous work (Park et al., 2021). Second, by comparing models with a fixed sequence dropout rates *r* from 0 to 0.5, and a model that adopts a random rate from normal distribution, we decide whether we would want to fix the sequence dropout rate of *r* or bring a complete randomness into the training phase. We did not raise the *r* over 0.5 (i.e., dropping 50% or more utterances every time) to avoid any possible data loss while training. All results are compared with the baseline model trained on full sequences-only, adopted from our previous work (Park et al., 2021). Furthermore, to account for the nature of randomness of sequence dropout approach, we run the models 5 times and average the results from each fold. Finally, we apply both the attention-based user-aware and the sequence dropout approaches to see that brings an additional performance enhancement.

We evaluate the performance of the disruptive talk detection models using the area under the receiver operating characteristic curve (AUC). AUC is one of the commonly used evaluation metrics for binary classification problems in machine learning, which represents the classification model's ability to separate between classes. The ROC curve shows the trade-off between true positive rate and false positive rate

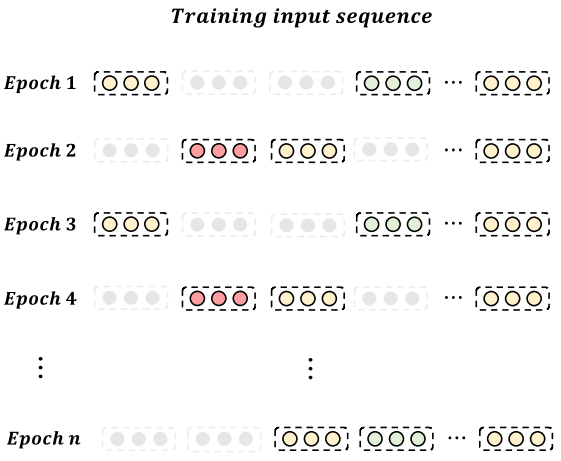


Figure 3: Sequence dropout training approach with a fixed rate of r. For the same training input sequence, the model drops r rate of inputs randomly. If r is chosen at random, a different number of inputs will be removed every epoch.

when varying the threshold values. An AUC of 1 indicates the classifier can perfectly discriminate between two classes, and 0.5 indicates the classifier cannot discriminate between two classes. We also evaluate the performance of the models using the area under the precision recall curve (PR-AUC) since AUC can give over-optimistic scores when the number of positive and negative classes are not balanced (Davis and Goadrich, 2006; Saito and Rehmsmeier, 2015). Like the ROC curve, the PR curve shows trade-off between precision (y-axis) and recall (x-axis) for different threshold values. It should be noted that when evaluating models based on the PR-AUC, it is essential to compare the performance with the PR-AUC of a no-skill classifier (i.e., Random chance), as the baseline performance varies depending on the task and the data distribution. To compare predictive performance, we report the average AUC and the average PR-AUC from cross-validation results.

We apply stratified group-level 10-fold cross-validation to avoid data leakage between training and testing data and retain the class distribution across folds. For each fold, we split the training data into a training and validation set to perform the early stopping based on the validation set. The distribution and the size of the validation set is the same as the test set. For all modeling approaches, we set the number of hidden units to 64, batch size to 32, and the number of epochs to 20, while using early stopping with a patience of 5 to avoid overfitting.

5 Result and Discussion

Table 2 shows evaluation results of the baseline model and the user-aware networks for disruptive talk detection. Our attention-based user-aware modeling approach outperforms the baseline modeling approach with respect to AUC (p=0.065), while it also brings improvement with respect to PR-AUC (p=0.161), where the statistical tests were conducted using the Friedman test, which is the non-parametric statistical test for multiple machine learning classifiers over multiple data sets, with a post-hoc analysis with the Wilcoxon signed rank test (Demšar, 2006). These results suggest that having the user-specific network was helpful for the model to identify whether the target utterance is disruptive or not. This might be because the user-specific network examines how the messages of target students have been developed without being affected by other student messages. The model obtains a clearer sense of the user's potential to be disruptive in a group conversation. In addition, it is possible that giving more weights to the hidden states that are more relevant to the target user embedding was effective to identify where to attend in the potentially noisy group sequence representation for the disruptive talk prediction of the target user.

Model	AUC	PR-AUC
No-Skill	0.5000	0.2466
Baseline	0.8292	0.5504
User-Aware	**0.8480**	**0.5691**

Table 2: Results of attention-based user-aware network (User-Aware). The best performance of each evaluation metric is marked in bold.

Table 3 shows the performance of sequence dropout approach (i.e., sequence dropout applied to a group-level network without a user-aware network) across the different sequence dropout rates and random choice. Except for $r = 0.1, 0.2$, all modeling approaches using different sequence dropout rates outperform the baseline with respect to AUC with a statistical significance ($p < 0.05$) when they were tested with the Wilcoxon signed rank test, while the model with random dropout rates applied perform the best. There were no significant differences in the performances among different dropout rates, except for the model using r is 0.1 or 0.2. These results might suggest that learning patterns from different sequence combinations were helpful for the disruptive talk detection model but dropping too few utterances would bring less significant enhancement to the performance. With respect to PR-AUC, the model

Dropout Rate (r)	AUC	PR-AUC
Baseline (r = 0)	0.8292	0.5504
Random (0, 0.5)	**0.8557***	**0.5649**
0.1	0.8413	0.5492
0.2	0.8426	0.5466
0.3	0.8507*	0.5517
0.4	0.8543*	0.5494
0.5	0.8498*	0.5526

Table 3: Sequence dropout approach across different dropout rate of r. The best performance of each evaluation metric is marked in bold, and * represents there is a statistically significant difference compared to the baseline.

Model	AUC	PR-AUC
Baseline	0.8292	0.5504
User-Aware	0.8480	0.5691
Sequence Drop	0.8557*	0.5649
SeqDrop+User-Aware	**0.8675***	**0.5991***

Table 4: Disruptive talk prediction results. The best performance of each evaluation metric is marked in bold, and * represents there is a statistically significant difference compared to the baseline.

with the random sequence dropout choice demonstrated improved performance compared to the other competitive modeling approaches, although the difference is not statistically significant when compared to the baseline model (p=0.138).

The performance enhancement with the sequence dropout training mechanism suggests that the conversation sequences may have contained noise due to the presence of multiple conversation threads, and that the model had some trouble determining how to extract the essential parts of the conversation sequences that could help with disruptive talk predictions. The model was given the opportunity to learn multiple variants of utterances from the same sequence because of the random dropping of a different subset of sequences at each training epoch. It is possible that this method could help achieve improved predictive performance by regularizing the disruptive talk detection models to effectively deal with noisy conversation data.

Finally, we compare the baseline model with the combined model, which utilizes both the attention-based user-aware and sequence dropout approaches. We compare the performance of this combined model with the models from the previous phases. Here, we adopt the random choice for the sequence dropout rate since it yielded higher performance with respect to both AUC and PR-AUC than the ones with the fixed sequence dropout rate. Results in Table 4 shows that our proposed disruptive talk detection model combining both User-Aware and Sequence dropout approaches. Our proposed regularized user-aware networks significantly outperform the baseline approach for both evaluation metrics (p<0.01 for AUC and p=0.09 for PR-AUC) with an alpha of 0.1. It also outperforms the models using each of the two proposed mechanisms: user-aware only (p<0.01 for AUC and p=0.06 for PR-AUC) and sequence dropout only (p=0.08 for AUC and p=0.16 for PR-AUC). These results suggest that the combined approach brings a synergetic effect to disruptive talk detection prediction. We observed from our repeated experiments (i.e., 5 executions) for all models using sequence dropout during training that the coefficient of variations (i.e., standard deviation / mean) of all approaches are less than 1, which is considered to be low variance between the values. This might suggest that the models were reliably trained even with randomness that resulted from dropping for a different set of utterances in dialogue sequences in each run.

Lastly, we note potential limitations of our research. Because of the nature of stratified group-level sampling where the sampling procedure must take into account both the label distribution and the group index, it is not possible to apply the exact same distribution across different folds, which could result in large performance variations between folds. In addition, while our proposed modeling approach demonstrated a promising result in our testbed collaborative game-based learning environment, the proposed model could be evaluated with other computer-supported collaborative learning environments to demonstrate generalizability of the technique.

6 Conclusion

Multi-party dialogue modeling poses significant challenges because of the complexity driven by group dynamics characterized in multi-party conversations. Detecting disruptive talk in collaborative game-based learning environments is crucial to support high-quality collaborative learning. We have presented a novel deep learning-based disruptive talk detection model that incorporates a user-aware attention network and a random sequence dropout training mechanism, where the model utilizing both approaches significantly outperform the baseline approaches. The proposed model shows significant promise for addressing key challenges in multi-party dialogue prediction. In the future, it will be important to test our model's capability with multi-party dialogue corpora from other computer-supported collaborative learning environments to test the generalizability of our model. It will also be important to implement the disruptive talk detection model in a real-time setting and investigate how it informs adaptive support for collaborative student learning.

Acknowledgments

This research was supported by the National Science Foundation under Grants DRL-1561655, DRL-1561486, IIS-1839966, and SES-1840120. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Ali Ahmadvand, Jason Ingyu Choi, and Eugene Agichtein. 2019. Contextual dialogue act classification for open-domain conversational agents. *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 1273-1276.

Sergio Grau, Emilio Sanchis, Maria Jose Castro, and David Vilar. 2004. Dialogue act classification using a Bayesian approach." In *9th Conference Speech and Computer*.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2012. Classifying dialogue acts in multi-party live chats. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 463-472.

Wookhee Min, Randall Spain, Jason D. Saville, Bradford Mott, Keith Brawner, Joan Johnston, and James Lester. 2021. Multidimensional team communication modeling for adaptive team training: A hybrid deep learning and graphical modeling framework. In *International Conference on Artificial Intelligence in Education*, pages 293-305.

Vladislav Maraev, Bill Noble, Chiara Mazzocconi, and Christine Howes. 2021 Dialogue act classification is a laughing matter. In *Proceedings of the 25th Workshop on the Semantics and Pragmatics of Dialogue*.

Zahra Rahimi and Diane Litman. 2018. Weighting model based on group dynamics to measure convergence in multi-party dialogue. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 385–390, Melbourne, Australia. Association for Computational Linguistics.

Elijah Mayfield, David Adamson, and Carolyn Penstein Rosé. 2012. Hierarchical Conversation Structure Prediction in Multi-Party Chat. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 60–69, Seoul, South Korea. Association for Computational Linguistics.

Ming Tan, Dakuo Wang, Yupeng Gao, Haoyu Wang, Saloni Potdar, Xiaoxiao Guo, Shiyu Chang, and Mo Yu. 2019. Context-aware conversation thread detection in multi-party chat. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6456-6461. 2019.

Pierre Dillenbourg, Sanna Järvelä, and Frank Fischer. 2009. The evolution of research on computer-supported collaborative learning. In *Technology-enhanced learning*, pages 3-19. Springer, Dordrecht.

Cindy E. Hmelo-Silver. 2004. Problem-based learning: What and how do students learn? *Educational psychology review*, 16(3): 235-266.

Heisawn Jeong, Cindy E. Hmelo-Silver, and Kihyun Jo. 2019. Ten years of computer-supported collaborative learning: A meta-analysis of CSCL in STEM education during 2005–2014. *Educational research review*, 28 (2019): 100284.

Asmalina Saleh, Chen Feng, Haesol Bae, Cindy E. Hmelo-Silver, K. Glazewski, Seung Lee, Bradford Mott, and James Lester. 2021. Negotiating accountability and epistemic stances in middle-school collaborative discourse. In *Proceedings of the International Conference on Computer-Supported Collaborative Learning*.

Dan Carpenter, Andrew Emerson, Bradford W. Mott, Asmalina Saleh, Krista D. Glazewski, Cindy E. Hmelo-Silver, and James C. Lester. 2020. Detecting off-task behavior from student dialogue in game-based collaborative learning. In *International Conference on Artificial Intelligence in Education*, pages. 55-66. Springer, Cham.

Stefanos Nikiforos, Spyros Tzanavaris, and Katia-Lida Kermanidis. 2020. Virtual learning communities (VLCs) rethinking: influence on behavior modification—bullying detection through machine learning and natural language processing. *Journal of Computers in Education* 7(4): 531-551.

Kyungjin Park, Hyunwoo Sohn, Bradford Mott, Wookhee Min, Asmalina Saleh, Krista Glazewski, Cindy Hmelo-Silver, and James Lester. 2021. Detecting disruptive talk in student chat-based discussion within collaborative game-based learning environments. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 405-415.

Tatiana Anikina and Ivana Kruijff-Korbayová. 2019. Dialogue act classification in team communication for robot assisted disaster response. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 399-410.

Jeremy Blackburn and Haewoon Kwak. 2014. STFU NOOB! predicting crowdsourced decisions on toxic behavior in online games. In *Proceedings of the*

23rd international conference on World wide web, pages 877-888.

Aslı Ekiciler, İmran Ahioğlu, Nihan Yıldırım, İpekİlkkaracan Ajas, and Tolga Kaya. 2021. The bullying game: Sexism based toxic language analysis on online games chat logs by text mining. In *Conference on Gender Studies and Sexuality*.

Ameneh Shamekhi, Q. Vera Liao, Dakuo Wang, Rachel KE Bellamy, and Thomas Erickson. 2018. Face Value? Exploring the effects of embodiment for a group facilitation agent. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1-13. 2018.

Amy X. Zhang, and Justin Cranshaw. 2018. Making sense of group chat through collaborative tagging and summarization. *Proceedings of the ACM on Human-Computer Interaction* 2(CSCW): 1-27.

Bastian Kordyaka. 2018. Digital Poison Approaching a theory of toxic behavior in MOBA games. In *International Conference on Information Systems*.

Bradford Mott, Robert Taylor, Seung Lee, Jonathan Rowe, Asmalina Saleh, Krista Glazewski, Cindy Hmelo-Silver, and James Lester. 2019. Designing and developing interactive narratives for collaborative problem-based learning. *Proceedings of the Twelfth International Conference on Interactive Digital Storytelling*, pages 86-100, Snowbird, Utah.

Asmalina Saleh, Cindy Hmelo-Silver, Krista Glazewski, Bradford Mott, Yuxin Chen, Jonathan Rowe, and James Lester. 2019. Collaborative Inquiry Play: A Design Case to Frame Integration of Collaborative Problem Solving with Story-Centric Games. *Information and Learning Sciences*, 120(9): 547-566.

Marcela Borge and Emma Mercier. 2019. Towards a micro-ecological approach to CSCL. *International Journal of Computer-Supported Collaborative Learning*, 14(2): 219-235.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1): 37-46.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Mahnaz Koupaee, Greg Durrett, Nathanael Chambers, and Niranjan Balasubramanian. 2021 Don't let discourse confine your model: Sequence perturbations for improved event language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 599-604.

Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233-240.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10(3): e0118432.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.

Generating Discourse Connectives with Pre-trained Language Models: Conditioning on Discourse Relations Helps Reconstruct the PDTB *

Symon Jory Stevens-Guille **Aleksandre Maskharashvili** **Xintong Li**
and **Michael White**
Department of Linguistics, The Ohio State University

Abstract

We report results of experiments using BART (Lewis et al., 2019) and the Penn Discourse Tree Bank (Webber et al., 2019) (PDTB) to generate texts with correctly realized discourse relations. We address a question left open by previous research (Yung et al., 2021; Ko and Li, 2020) concerning whether conditioning the model on the intended discourse relation—which corresponds to adding explicit discourse relation information into the input to the model—improves its performance. Our results suggest that including discourse relation information in the input of the model significantly improves the consistency with which it produces a correctly realized discourse relation in the output. We compare our models' performance to known results concerning the discourse structures found in written text and their possible explanations in terms of discourse interpretation strategies hypothesized in the psycholinguistics literature. Our findings suggest that natural language generation models based on current pre-trained Transformers will benefit from infusion with discourse level information if they aim to construct discourses with the intended relations.

1 Introduction

Traditional approaches to discourse have shown the essential importance of discourse (rhetorical) relations in providing coherence to a text (Mann and Thompson, 1987; Lascarides and Asher, 2008; Kehler and Kehler, 2002). While current approaches to natural language generation (NLG) employing pre-trained models have been shown to excel in generating well-formed texts (Kale and Rastogi, 2020, i.a.), their ability to produce coherent texts structured with the help of discourse connectives is understudied (Maskharashvili et al., 2021). The impetus for the present study is the growing body of evidence that neural models, whether trained fresh (Stevens-Guille et al., 2020) or pre-trained (Maskharashvili et al., 2021), benefit from input which includes specific reference to the discourse structure intended to hold in the output text (Balakrishnan et al., 2019). This line of work is novel in the context of current NLG practice, which frequently omits cues to discourse structure in the input. The previous work is purposefully restricted to producing relatively homogeneous texts (museum descriptions and weather predictions). Given the findings of this work on generating limited sets of discourse relations and connectives, it is informative to study the performance of neural models in generating texts structured with the help of a richer set of discourse relations realized by a wide variety of discourse connectives. We study whether having discourse relation information in the input helps neural models to realize the intended discourse relation. These conditions more closely approximate the context in which robust NLG systems would be deployed. We expect our results to provide insight into whether and how to include discourse structure cues in fully-fledged NLG systems.

We report the results of our experiments using BART (Lewis et al., 2019) and the Penn Discourse Tree Bank (Webber et al., 2019) (PDTB) to generate texts with correctly realized discourse relations. We address a question left open by previous research (Yung et al., 2021; Ko and Li, 2020) concerning whether conditioning the model on the intended discourse relation—which corresponds to adding explicit discourse relation information into the input to the model—improves its performance. While we recognize that a positive answer to this question might seem obvious, it has, to date, not been supported with quantitative evidence. We compare our models' performance to baselines in which i) connective choice is determined by the most frequent connective which realizes the intended relation in the corpus, (ii) connec-

*E-mail: stevensguille.1@buckeyemail.osu.edu

Proceedings of the SIGdial 2022 Conference, pages 500–515
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

tive choice is determine by the most frequent connective in the corpus irrespective of the intended relation to be expressed, (iii) connective choice is determined by off-the-shelf BART-large mask substitution, and (iv) connective choice is determined by off-the-shelf BERT (Devlin et al., 2019) mask substitution. We propose two types of models by fine-tuning BART on PDTB: models that have discourse relation information in the input (D+ models) and models that do not (D- models). We find that our fine-tuned D+ models substantially outperform fine-tuned D- models, while both kinds of fine-tuned models dramatically beat the baselines. In addition, fine-tuned D+ models produce systematically fewer errors than corresponding D- ones when tested against psycholinguistic observations that certain discourse relations tend to be realized implicitly, while others usually are realized by explicit (overt) discourse connectives. It is important to also point out that our fine-tuned models, unlike previous work and some of our baselines, are not given the position into which the connective should be inserted. This more closely approximates the intended usage of end-to-end neural models, where there is no module in which connectives are slotted into predetermined positions in the output string. We find the models' choices for connective positions to be qualitatively good and focus in the sequel on the connective choices themselves.[1]

2 Background

BART, a transformer-based (Vaswani et al., 2017) language model, is trained on purposefully corrupted data so that the model learns to 'denoise' the corrupted input in the process of reconstructing the original data. Fine-tuning BART on different versions of input and output lets us probe whether the underlying language model needs or benefits from explicit cues to consistently reconstruct the intended discourse connective. The PDTB is one of the few corpora developed to identify discourse dependencies in texts. PDTB provides a well-developed ontology of discourse relations; these discourse relations are used to annotate the Wall Street Journal (WSJ) corpus of the Penn Treebank. We construct versions of the corpus differing in (i) whether the order of the arguments in the output is explicitly encoded in the input, (ii) whether the output is the connective or the connective embedded in the corresponding WSJ text, and (iii) whether a discourse relation is included in the input and how specific it is. The third difference is conceptually the most important one since it corresponds to whether the model is conditioned on discourse relation information.

To determine how well the models realize discourse relations, in addition to standard metrics (i.e., recall and precision), we employ more recent metrics inspired by psycholinguistic (Murray, 1997; Sanders, 2005; Yung et al., 2021) and corpus studies (Asr and Demberg, 2012, 2013; Jin and de Marneffe, 2015a) which allow us to find out the degree to which the models' preferences for realizing different discourse relations correspond to reported human preferences for realizing those relations. In particular, it is argued that while some discourse relations are mostly expressed explicitly, by means of a discourse connective (i.e., overt lexical item or items), other discourse relations tend to be expressed implicitly, i.e., without explicit lexical markers. One of the questions we want to answer is whether providing a discourse relation in the input helps models to learn when to realize a discourse relation explicitly and/or implicitly.

Asr and Demberg (2012, 2013) argue that the PDTB provides ample evidence for psycholinguistic patterns of behaviour. In lieu of directly running human judgement experiments on our model outputs, we test our models' consistency with psycholinguistic results indirectly: we compare the distributions in model outputs to those distributions in the corpus which have been argued to support psycholinguistic theories. We focus on the following two hypotheses:

The Continuity Hypothesis: 'Readers have a bias towards interpreting sentences in a narrative as following one another in a continuous manner ... additive additive and causal connectives should lead to less processing facilitation than adversative connectives because the former indicate continuity in the discourse whereas the adversatives indicate discontinuity.' — Murray (1997, p.228-229)

The Causality-by-default Hypothesis: 'Because readers aim at building the most informative representation, they start out assuming the relation between two consecutive sentences is a causal relation ... Subsequently,

[1] In the appendix we provide examples of initial and final connectives which complement the medial connectives used throughout the rest of the paper. We note, however, that our BART-base models prefer producing appropriately positioned initial or medial connectives rather than final connectives.

causally related information will be processed faster, because the reader will only arrive at an additive relation if no causal relation can be established.' — Sanders (2005)

Asr and Demberg (2012) report that in the PDTB, relations that they consider discontinuous and continuous in the sense of Murray (1997) are more likely to be realized explicitly and implicitly, respectively, which is consistent with the continuity hypothesis. Furthermore, they find the proportion of implicit to explicit connectives is highest for causal senses. They conclude this provides support for the hypothesis of causality-by-default.

Asr and Demberg (2013) propose a metric for deriving 'markedness'—how much information about the intended discourse relation is conveyed by a connective or set of connectives—from the PDTB. However, no prior work on the PDTB conditioned models on the discourse relation intended to be communicated. To our knowledge, the following questions, which we report on here, are yet to explored in NLG: (i) whether conditioning on discourse relations improves the prediction of intended discourse connectives; (ii) whether ordering information concerning the arguments to be expressed should be encoded explicitly; and (iii) whether neural models can learn distributions consistent with psycholinguistic results (Sanders, 2005; Murray, 1997) known to be reflected in the training set (Asr and Demberg, 2012).

3 Methods

Our experiments use the BART-base and BART-large implementations of HuggingFace fine-tuned on different versions of the reconstructed PDTB corpus, which we describe in the sequel.[2]The corpus is a modified version of the WSJ texts derived from reconstructing the texts from the string positions provided by the PDTB. Our modifications were intended to make the reconstructions more natural for pre-trained models by using full sentences but without giving away hints to connective location by capitalization or punctuation. An example is provided in Figure 1.[3] The input consists of the set of '<sep>' separated items, while the output is the text the model is trained to produce. The output is either the reconstructed text (marked as full-ouput) or the connective of import in the reconstructed text (marked as conn-only).[4]

We produced in total sixteen different versions of the corpus (see Table 3), twelve of them with discourse relations in the input, which we dub BART_{D+} models, and four without these relations, which we dub BART_{D-} models. Previous work (Asr and Demberg, 2012) found correspondence between the distribution of implicit (respectively explicit) connectives in the WSJ and human behavior reported by Murray (1997); Sanders (2005) concerning which discourse relations are expected (respectively less expected). We produced three versions of the corpus reflecting different levels in the PDTB sense hierarchy as follows:

- Level 1 is the top level (Temporal, Expansion, Comparison, Contingency).
- Level 2 is the set of children of the level 1.
- Full is the set of complete senses, the depth of which is no more than 3.

To determine whether the order of arguments in the PDTB affects the model's choice of connective, we further divided the corpus into versions which included or didn't include explicit encoding of the order of arguments in the output: '12' encodes the case where the first argument precedes the second argument in the reference text, while '21' corresponds to the second argument preceding the fist argument in the reference text. This is in principle useful since the order of arguments in the input need not reflect their order in the output. To control for the influence of generating full texts, we produced versions of the corpus in which the outputs were the discourse connectives without the surrounding WSJ text. Since whether the discourse connective is left implicit or made explicit is something we would like to test every model on, we include no information about whether the connective should be implicit or explicit in the input.

The difference between BART_{D+} and BART_{D-}, corresponding to whether the input includes the discourse connective's type,

[2]We find little difference in the performance of BART-base and BART-large and therefore focus on BART-base throughout the paper. Results on matching for BART-large can be found in appendix G.

[3]Due to reconstruction from string positions, the output text is sometimes missing spaces or punctuation from the end of the arguments. The first letter of every argument in the input is in lower case for uniformity. The context arguments can be empty if the string indices from the PDTB correspond to sentences. The string 'none' is inserted into empty contexts. If the PDTB string indices do not correspond to sentences, the context arguments correspond to the sentences in which the PDTB string indices were embedded.

[4]While the PDTB is licensed from the LDC, the scripts for producing our corpora from it plus the metrics and model details will be made freely available on https://github.com/SymonJoryStevens-Guille/PennGen.

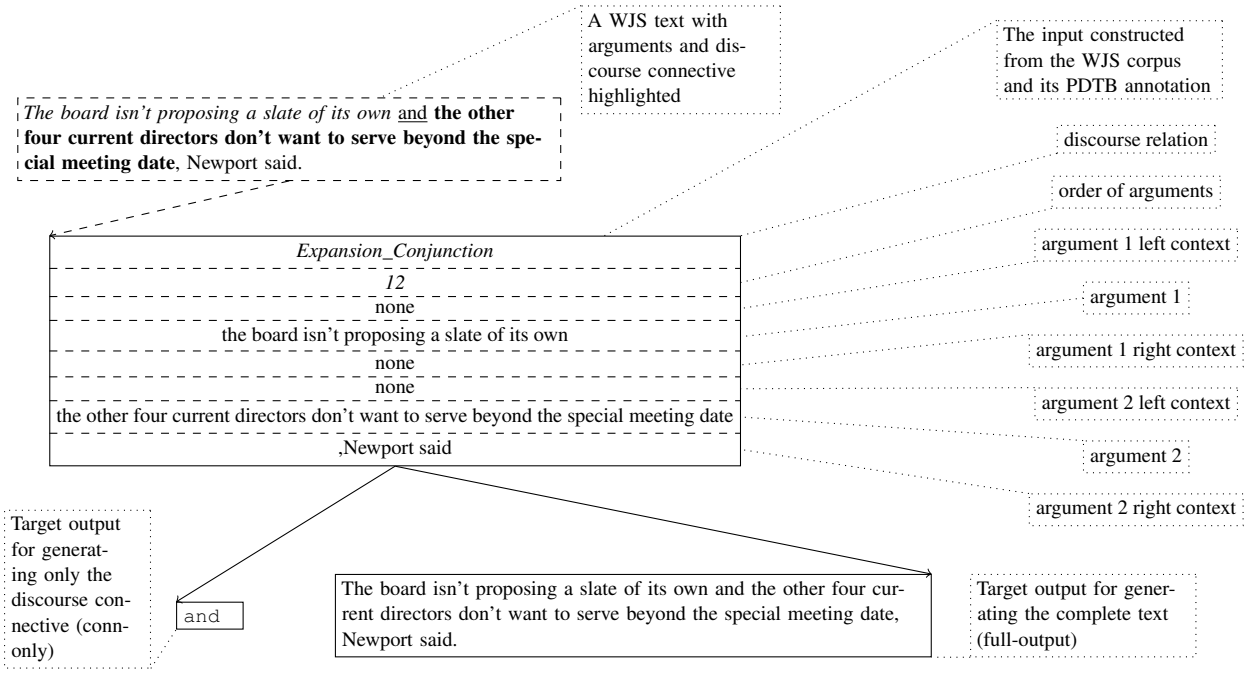


Figure 1: A WSJ text together with its PDTB annotation used in constructing the input to the models and their target outputs. (In the linearized input form, the fields are separated by a <sep> token.)

rounds out the set of distinctions between corpus versions. Details of the corpus split into train/dev/test can be found in Appendix A.

4 Metrics

In order to study whether the models can reconstruct the discourse connectives found in the WSJ, we report model-reference matching. We further consider matching with respect to implicit and explicit discourse connectives.

Implicit and explicit matching, both independently and summed, is our first metric. Second, we consider the proximity of the mismatches between reference and generated connective in terms of the PDTB sense hierarchy. With respect to Figure 1, matching would be producing *and*, either in the embedded sentence in which it occurs (full-output) or on its own (conn-only). Since the type of *and* in this context is Expansion_Conjunction, the mismatch could be by substitution of some other connective from Expansion_Conjunction (=full), from one of the subsenses of Expansion (=level 1), or from some completely different sense (=level 2).

We test the consistency of the models with the continuity and causality-by-default hypotheses by reference to the metrics proposed by (Asr and Demberg, 2012, 2013) to quantify the support for such hypotheses in the PDTB. The usefulness of the distribution of implicit versus explicit connectives is helpfully summed up by Asr and Demberg (2012, pg. 2671): "if readers have a default preference to infer a specific relation in the text, this type of relation should tend to appear without explicit markers." This likewise motivates our use of the metric of markedness, to be discussed below, since markedness quantifies how expected a relation is and, in conjunction with the hypothesis of Uniform Information Density (UID) (Jaeger and Levy, 2006), how likely it is to be explicitly cued versus left to be inferred (Asr and Demberg, 2013).

Asr and Demberg (2012) propose to define implicitness of PDTB senses in terms of the distribution of implicit discourse relations corresponding to the sense in the corpus (# of implicit tokens of senses divided by # of tokens of senses). Following Asr and Demberg (2012, 2013); Jin and de Marneffe (2015b), we focus on the two groups of (sub)types in Table 1, which respectively represent discontinuous and/or noncausal relations and continuous and/or causal relations.[5]

Implicitness and explicitness provide one sort of proxy for continuity and discontinuity in our metrics. We therefore compare the distribution of

[5]We ignore some relations identified by the foregoing authors which don't appear frequently enough in our test set. We report results for the level 1 relations too.

Continuous	Discontinuous
Contingency_Cause	Comparison
Expansion_Instantiation	Temporal_Asynchronous

Table 1: Continuous discourse relation types are shown on the left and discontinuous ones on the right

implicitness predicted by our models to the distribution of implicitness in the test set to determine the fit of the model with respect to the continuity hypothesis.

Following Asr and Demberg (2013); Jin and de Marneffe (2015b), we make use of a metric of markedness, which Asr and Demberg (2013) argue provides a good picture of how likely a given relation is to appear with a connective and to what degree the relation-connective co-occurrence is unique: The higher the markedness, the more likely the relation is to appear with a set of specific connectives. Consequently, it would be more surprising to have that relation cued by a less expected connective or no connective at all—we should then expect both causal and continuous relations to have lower markedness.

Asr and Demberg (2013) report the markedness of level 1 relations, finding the gross cline of Temporal < Contingency < Expansion < Comparison. They argue these results are consistent with the UID, the continuity-by-default hypothesis, and the causality-by-default hypothesis. We consider the degree to which the markedness cline of our model outputs corresponds to the markedness cline of the corpus to provide evidence of whether the model is learning to produce text consistent with the previously mentioned cognitive biases.

Markedness is defined in the equation below, where *npmi* is normalized point-wise mutual information, r belongs to the set of relations R, and c belongs to the set of connectives C minus the null connective.

$$markedness(r) = \sum_{c \in C} p(c|r) \frac{npmi(r;c) + 1}{2}$$

Since the markedness metric doesn't provide a direct probability distribution, significance for differences between markedness must be measured by non-parametric methods. For these purposes we use approximate randomization (AR) (Noreen 1989): we randomly re-sample from the two models' union, producing 30K versions of the results and comparing whether and how many such versions improve over the different model predictions in terms of proximity to the reference score (we describe AR at length in Appendix C).

5 Results

With respect to the types of fine-tuning we experimented with, we find BART$_{D+}$ models routinely exceed BART$_{D-}$ models. We show here that BART$_{D+}$ models seem to recover even some of the distributions found by Asr and Demberg (2012) to support psycholinguistic results concerning discourse structure.[6]

In Table 3 we include the results of our baselines. Both models D+ (=80.5%) and D- (=67.2%) significantly improve over the corresponding baseline D+ and D- models. This improvement is further corroborated by comparing BERT and BART-large off-the-shelf to the corresponding BART$_{D+}$ and BART$_{D-}$ models. Both D+ (=79%) and D- (=71.3%) make over a 20% improvement on both BERT and BART-large off-the-shelf. Since the off-the-shelf models were given intended position information in the form of MASK tokens, the result shows that this positional information, at least without fine-tuning, doesn't suffice to predict the intended connective.[7]

Interestingly, with respect to producing matching explicit (respectively implicit) connectives, the models trained to produce full sentence outputs frequently outperform the models trained to produce only discourse connective outputs. This is shown in Table 3, where the difference in scores is most visible when the model is provided with less or no information concerning the intended discourse relation. This suggests there is some benefit to producing the connective in context, where the fidelity of the decoded connective is improved by the preceding and subsequent strings. But this benefit seems to taper off from depth 2 down.

There seems to be a sweet spot in the level of discourse relation type included in the input: there is little improvement between full and level 2 types

[6]The chosen connective need not occur directly between the arguments in the input. Determining which connective is produced by the full-out model is done by iteratively substituting elements of the input found in the output with the empty string. Once this process is complete the remaining strings will include the connective. Strings which are not in the complete set of connectives are removed to eliminate noise. If no connective is found after this process then the model evidently chose to leave the relation implicit.

[7]Note that the MASK position for implicits is uniformly between the rightmost position of arg1 and the leftmost position of arg2. We chose this position for uniformity in light of the absence of implicit connective span annotation in the PDTB.

Type	Comparison	Contingency	Expansion	Temporal
Reference	0.53624	0.21141	0.34245	0.47499
BART$_{D+}$	0.62666	0.21302	0.32155	0.46709
BART$_{D-}$	0.35860	0.19829	0.29256	0.42175

Table 2: Level 1 markedness scores by model

Type	Level	Order	FullOutput	Match
baseline$_{D+}$		-	-	34%
baseline$_{D-}$		-	-	16%
BART-large		-	+	48%
BERT		-	+	52.4%
BART$_{D+}$	full	+	+	79%
BART$_{D+}$	full	+	-	81.5%
BART$_{D+}$	full	-	+	79%
BART$_{D+}$	full	-	-	80.5%
BART$_{D+}$	2	+	+	79.3%
BART$_{D+}$	2	+	-	79.9%
BART$_{D+}$	2	-	+	79%
BART$_{D+}$	2	-	-	79.8%
BART$_{D+}$	1	+	+	76.5%
BART$_{D+}$	1	+	-	66.9%
BART$_{D+}$	1	-	+	75.7%
BART$_{D+}$	1	-	-	65.5%
BART$_{D-}$		+	+	70.5%
BART$_{D-}$		+	-	69.9%
BART$_{D-}$		-	+	71.3%
BART$_{D-}$		-	-	67.2%

Table 3: Model typology with distributions of matched versus reference discourse connectives. BART-large and BERT are baselines used off-the-shelf; baseline$_{D+}$ is the majority baseline conditioned on discourse relations and baseline$_{D-}$ is the majority baseline unconditioned on discourse relations.

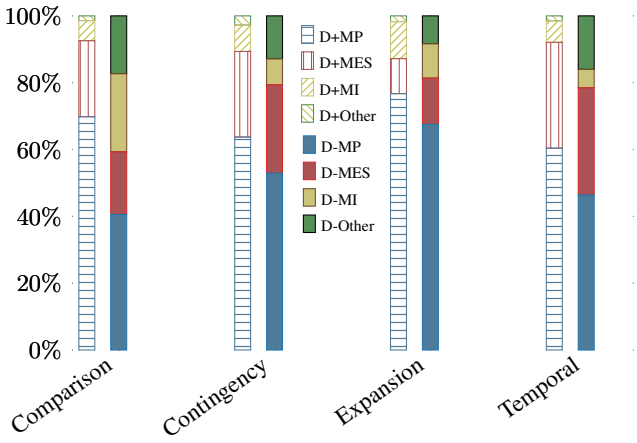


Figure 2: Case of Explicit Connectives: Graph for Models D+ and D- showing MP (Match Prediction); MES (Mismatch with Explicit Sister type); MI (Mismatch with Implicit); and Other (Explicit For Explicit Minus One, Minus Two, and Minus Three level types)

but there is greater improvement between level 2 and level 1—for conn-only the BART$_{D-}$ models even sometimes exceed the level one BART$_{D+}$ models, suggesting the top level type information could hinder connective choice when the connective isn't generated in context.

Despite the boost to connective matching when producing conn-only, the distinction between models which condition on the order of arguments versus those that do not, controlling for other corpus distinctions, is minimal when present. This, too, is visible in Table 3.

The major difference between models with respect to reconstructing the reference connectives is the difference between the BART$_{D+}$ and BART$_{D-}$ models. The BART$_{D+}$ models from the second level to the full level outperform the BART$_{D-}$ models when controlling for the input order, whether the output is full-output or conn-only. In the sequel we report significance results just for the best BART$_{D-}$ model and a corresponding BART$_{D+}$ model: BART$_{D-}$ (-Order,+FullOutput) and BART$_{D+}$ (Depth 3,-Order,+FullOutput). While the level 2 BART$_{D+}$ model ekes out the level 3 model, the difference is uninteresting.

The main distinction in matching between BART$_{D+}$ and BART$_{D-}$ models is due to explicit connectives. Both models perform well with respect to reconstructing implicit connectives, though the differences even here are significant, with the BART$_{D-}$ model even improving over the BART$_{D+}$ model with respect to implicit relations. However, this observation points to a more likely story for BART$_{D-}$'s performance: the BART$_{D-}$ model is less accurate. This is corroborated by it generating an excess of 405 implicits for target explicits compared to BART$_{D+}$'s 275 implicits for target explicits. The overproduction of implicits is further borne out by the differences in F1 shown

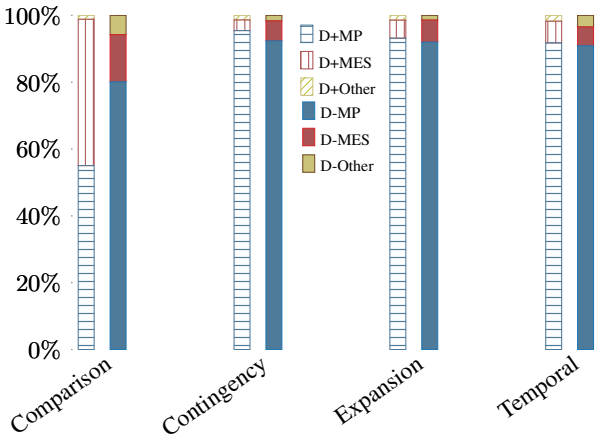


Figure 3: Case of Implicit Connectives: Graph for Models D+ and D- showing MP (Match Prediction), MES (Mismatch with Explicit Sister type), and Other (Explicit For Implicit Minus One, Minus Two, and Minus Three types)

Model	Explicit Match	Implicit Match
BART$_{D+}$	69.8%	89.2%
BART$_{D-}$	54.3%	90.6%

Table 4: Matches for BART$_{D+}$ and BART$_{D-}$.

in Figures 19 and 20 in Appendix D; errors concerning connective choice are exemplified and discussed there too.

Returning to the production of explicit connectives, the improvements of conditioning on discourse structure information are highly significant both with respect to matching per se and with respect to matching explicit connectives. We provide McNemar's test statistics for explicit matches (statistic=157.00, p=0.000), implicit matches (statistic=118.00, p=0.025), and their combination (statistic=313.000, p=0.00) (the tables can be found in Appendix B).

Table 4 shows match results for both implicit and explicit for BART$_{D-}$ and BART$_{D+}$. More fine-grained results are given in Figures 2 and 3. Focusing on the results concerning implicit relations first, the most noticeable difference is with respect to Comparison—the BART$_{D+}$ model produces far fewer matches than the BART$_{D-}$ model. However, within the mismatches here, the BART$_{D+}$ model overwhelmingly produces explicit connectives for reference implicit when the relation can be expressed by both an explicit or implicit connective. In fact the BART$_{D-}$ model makes more severe mismatches on Comparison

BART$_{D+}$: The board isn't proposing a slate of its own **and** the other four current directors don't want to serve beyond the special meeting date, Newport said.

BART$_{D-}$: The board isn't proposing a slate of its own **because** the other four current directors don't want to serve beyond the special meeting date, Newport said.

Figure 4: Full outputs on the input from Figure 1 for BART$_{D+}$ full and BART$_{D-}$ without cues to order. The model's generated connective is bolded.

than the BART$_{D+}$ model, since its productions of explicit connectives for reference implicit are more frequently productions of connectives which simply are not used to express the relation.

With respect to producing matching explicit connectives, the BART$_{D+}$ model exceeds the BART$_{D-}$ model on every top level type. When BART$_{D+}$ doesn't produce a matching explicit connective, it is far more likely to produce an explicit connective which expresses the same relation. For each top level type, the severity of the mismatch is less for BART$_{D+}$ than BART$_{D-}$. Without committing to the position that producing an implicit connective for a relation intended to be expressed explicitly is better or worse than producing an explicit connective for a relation intended to be expressed implicitly, we argue that either mismatch is better than producing a connective which is never used to express the intended relation. On this score, the BART$_{D-}$ model is considerably worse—it is consistently more likely to produce a connective not otherwise used to cue the intended discourse relation.

When the metrics are extended to include whether non-matching connectives chosen by the model fit the intended discourse relation, the BART$_{D+}$ model continues to outperform the best BART$_{D-}$ model. When producing non-matching connectives, we find that the chosen connectives of the BART$_{D+}$ model correspond to the intended discourse relations more frequently than those produced by the BART$_{D-}$ models.

We computed markedness scores for outputs of the BART$_{D+}$ and BART$_{D-}$ models. By applying AR significance tests on markedness score–based statistics, we find that the BART$_{D+}$ output on the test data is significantly closer to the reference data than the output of BART$_{D-}$. We show in Table 5 the distribution of markedness

Type	Contingency_Cause	Expansion_Instantiation	Comparison	Temporal_Asynchronous
Reference	0.182	0.071	0.536	0.436
BART$_{D+}$	0.179	0.061	0.626	0.419
BART$_{D-}$	0.170	0.058	0.358	0.365

Table 5: Markedness Scores for continuity and causality-by-default hypotheses.

	BART$_{D-}$		BART$_{D+}$	
Discourse Relation	Nonsister	Implicit	Nonsister	Implicit
Contingency_Cause	4.3%	2.3%	1.9%	2.7%
Expansion_Instantiation	6.1%	1.6%	2.1%	1.6%
Temporal_Asynchronous	**14.2%**	4.7%	2.3%	4.7%
Comparison	**13.9%**	**16.6%**	1.3%	4.2%
Overall	7.8%	6.6%	1.6%	4.5%

Table 6: Error rates in Mispredicted Nonsister and Mispredicted Implicit of the models' performances w.r.t discourse relations associated with the Continuity and Causality-by-default hypotheses. Proportions are with respect to the sum of the items in the test set meant to express the relation type.

for several discourse relations. We should first note that the BART$_{D+}$ markedness scores are consistently closer to the reference scores than the BART$_{D-}$ scores. Second we note that the continuity hypothesis is partially supported, even just considering this limited set of relations: both Contingency_Cause and Expansion_Instantiation are less marked than both Comparison and Temporal_Asynchronous. This is consistent with our hypotheses that continuous and causal relations should be less marked than discontinuous relations. However, like Jin and de Marneffe (2015b); Asr and Demberg (2013), we found less direct support for the causality-by-default hypothesis, since it is not less marked than Expansion_Instantiation. This is at best consistent with a weak form of the hypothesis, since we have not here reported contexts which would discriminate between the conjunction of the causality-by-default and continuity hypotheses versus just the continuity hypothesis. Our conclusions are further reinforced by Table 2, which shows the BART$_{D+}$ model, in particular, is reasonably close to recovering the markedness exemplified in the test set.

One further difference in distribution is predicted by the causality-by-default and continuity hypotheses for those relations that are or are not continuous or causal, exemplified by the relations found in Table 1. Both these hypotheses posit default inferences. Given the apparent reliability of these defaults with respect to psycholinguistic and corpus studies, we'd expect that learning these defaults would reduce the rate of errors for those relations to which the defaults apply. Consequently, we can compare the proportion of errors for continuous and causal relations to that for discontinuous (and non-causal) relations to determine how likely it is the model learned the default. We expect that explicitly representing discourse relations should support the learning of the default since, by hypotheses, the defaults are correlated with specific relations. Table 6 shows the error proportions.

We find that the D+ model shows a lower error proportion with respect to continuous versus discontinuous relations while the D- model shows a higher proportion of such errors, particularly where the relation to be expressed is discontinuous and more marked. We note that the number of D- Nonsister errors on Temporal_Asynchronous, which dwarf the Implicit errors on the same relation, is consistent with the continuity hypothesis in particular since these relations, which are not subject to default inferences, are important to mark explicitly yet are difficult to mark correctly in the absence of an explicit cue to the relation. On Comparison, D- makes a similar number of Nonsister errors, and also makes more than double the number of overall implicit prediction errors. This makes sense if the model recognizes it's important to signal these relations but erroneously treats

them as if they were in a default relation where leaving the connective implicit would be more expected. However, we do not have a ready explanation for why Temporal_Asynchronous does not have more implicit D- errors.

The differences in connective choice between models sometimes result in wildly divergent meanings. Figure 4 shows the BART$_{D+}$ full and BART$_{D-}$ outputs for the input in Figure 1. Neither model is conditioned on the order of arguments. The BART$_{D-}$ model's output uses *because* to erroneously communicate that the intentions of the directors cause the intentions of the board, whereas the BART$_{D+}$ model correctly identifies the intentions of the board and the intentions of the directors without suggesting either intention is dependent on the other (generates *and*).

6 Related Work

Ko and Li (2020) reported the limits of GPT-2 (Radford et al., 2019) for generating texts with discourse connectives. Their results concern both fine-tuning and off-the-shelf experiments. For fine-tuning they conditioned the model on prompt-response pairs, testing the subsequently fine-tuned model on the appropriateness of its output responses to input prompts in conversation. For GPT-2 off-the-shelf they fed the first argument and a candidate discourse connective to the model and took the output to be the second argument. They found that GPT-2 more frequently produced connectives consistent with the judgements concerning the discourse relation inferred by human subjects when their agreement on the discourse relation is high. Like Ko and Li, we are interested in discourse relation realization. However, in Ko and Li's approach the position of the discourse connective is explicitly given to the model (it's the mask). Also, Ko and Li's fine-tuned model is restricted to 11 connectives. We condition models on both the discourse relation and the arguments to provide fine-grained control of the discourse without restricting the position of the discourse connective.

Yung et al. (2021) found that GPT-2 diverges from human subjects in its judgements concerning the substitution of connectives which the PDTB does not distinguish by type. This provides presumptive evidence that large pre-trained language models could be limited in reconstructing human judgements concerning the sense of connectives and their substitutability.

7 Conclusion

The main conclusion one can draw from our results is that discourse relation information is essential for consistently generating matching discourse connectives. While large-scale human judgement experiments on our models' predictions are the most obvious next step, the improvement of the BART$_{D+}$ models over the BART$_{D-}$ models with respect to exact matching is encouraging, especially in light of recent results showing that humans don't uniformly accept substitution of discourse connectives which express the same discourse relation (Yung et al., 2021). With respect to whether mere arguments suffice to generate a discourse connective that correctly realizes the discourse relation holding between them, our results indicate that the purely distributional meaning of texts induced by the models under-determines the relation expressed by explicit discourse connectives. Directly conditioning on discourse relations in the input significantly improves the likelihood of the model producing a connective which corresponds to the intended discourse relation. One must note that conditioning on the discourse relation is especially important when the relation is marked, as in these cases the model is apt to predict an incorrect default (causal or continuous) relationship just from the arguments.

As for markedness score–based statistics, we can conclude that the presence of discourse relations in the input helped BART$_{D+}$ to learn the discourse connective distribution patterns of the PDTB. These metrics provide a useful avenue for testing how well generation models recover patterns which hold for a variety of different variables, from discourse relations themselves, to the strength of co-occurrences between discourse relations and the words used to communicate them. To the degree these patterns track cognitive dependencies, they encourage integration of cognitive models of discourse coherence and NLG evaluation.

Acknowledgments

We thank three anonymous reviewers for their feedback. This research was supported by a collaborative open science research agreement between Facebook and The Ohio State University. The last author has been a paid consultant for Facebook while the research was conducted.

References

Fatemeh Torabi Asr and Vera Demberg. 2012. Implicitness of discourse relations. In *Proceedings of COLING 2012*, pages 2669–2684.

Fatemeh Torabi Asr and Vera Demberg. 2013. On the information conveyed by discourse markers. In *Proceedings of the Fourth Annual Workshop on Cognitive Modeling and Computational Linguistics (CMCL)*, pages 84–93.

Anusha Balakrishnan, Vera Demberg, Chandra Khatri, Abhinav Rastogi, Donia Scott, Marilyn Walker, and Michael White. 2019. Proceedings of the 1st workshop on discourse structure in neural nlg. In *Proceedings of the 1st Workshop on Discourse Structure in Neural NLG*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

T Jaeger and Roger Levy. 2006. Speakers optimize information density through syntactic reduction. *Advances in neural information processing systems*, 19.

Lifeng Jin and Marie-Catherine de Marneffe. 2015a. The overall markedness of discourse relations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1114–1119, Lisbon, Portugal. Association for Computational Linguistics.

Lifeng Jin and Marie-Catherine de Marneffe. 2015b. The overall markedness of discourse relations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1114–1119, Lisbon, Portugal. Association for Computational Linguistics.

Mihir Kale and Abhinav Rastogi. 2020. Text-to-text pre-training for data-to-text tasks. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 97–102.

Andrew Kehler and Andrew Kehler. 2002. *Coherence, reference, and the theory of grammar*. CSLI publications Stanford, CA.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wei-Jen Ko and Junyi Jessy Li. 2020. Assessing discourse relations in language generation from gpt-2. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 52–59.

Alex Lascarides and Nicholas Asher. 2008. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Aleksandre Maskharashvili, Symon Stevens-Guille, Xintong Li, and Michael White. 2021. Neural methodius revisited: Do discourse relations help with pre-trained models too? In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 12–23, Aberdeen, Scotland, UK. Association for Computational Linguistics.

John D Murray. 1997. Connectives and narrative text: The role of continuity. *Memory & Cognition*, 25(2):227–236.

Eric W. Noreen. 1989. *Computer-intensive methods for testing hypotheses : an introduction*. Wiley, New York.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Ted Sanders. 2005. Coherence, causality and cognitive complexity in discourse. In *Proceedings/Actes SEM-05, First International Symposium on the exploration and modelling of meaning*, pages 105–114. University of Toulouse-le-Mirail Toulouse.

Symon Stevens-Guille, Aleksandre Maskharashvili, Amy Isard, Xintong Li, and Michael White. 2020. Neural NLG for methodius: From RST meaning representations to texts. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 306–315, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019. The penn discourse treebank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*.

Frances Yung, Merel Scholman, and Vera Demberg. 2021. A practical perspective on connective generation. In *Proceedings of the 2nd Workshop on Computational Approaches to Discourse*, pages 72–83.

Relation Data set	Compar.	Contig.	Expan.	Temp.
Train	5297	7592	12605	3302
Dev.	1173	1577	2635	784
Test	1195	1616	2563	774

Table 7: Numbers of occurrences of top level relation types in data sets

	m2y	m2n
m1y	1617	663
m1n	157	825

Table 8: m1 = BART$_{D+}$, m2 =BART$_{D-}$.
statistic=157.000, p-value=0.000
Different proportions of total number of entries with explicit matches (reject H0)

A Data sets collection and statistics

The corpus was split into train/dev/test by selecting the first 70 percent of reconstructed lines for training purposes. To prevent content from one split being encountered in another split, any remaining lines in a WSJ article encountered after the line corresponding to the end of train were removed. This technique is used for preventing spillover of content between dev and test, too, which respectively comprise approximately 15 percent of the corpus. Namely, we have 28796 items in the training set, 6169 in the dev set, and 6149 in the test set.

The breakdown of top level relations distributed through the splits is given in Table 7.

We excluded some items from the corpus if the resulting sequences would be too long, if the relations were not extensions of those defined by level 1 in the foregoing, or to prevent possible repetition of content between train, test, and dev splits.

B McNemar's Significance Results

McNemar's significance test results between BART$_{D+}$ and BART$_{D-}$ models are shown in Tables 8, 9, and 10.

	m2y	m2n
m1y	2459	118
m1n	156	153

Table 9: m1 = BART$_{D+}$, m2 =BART$_{D-}$.
statistic=118.000, p-value=0.025
Different proportions of total number of entries with implict matches (reject H0)

	m2y	m2n
m1y	4076	781
m1n	313	978

Table 10: m1 = BART$_{D+}$, m2 =BART$_{D-}$.
statistic=313.000, p-value=0.000
Different proportions of total number of entries with implicit or explicit matches (reject H0)

C Approximate Randomization with respect to Markedness Stats

We want to see whether markedness scores of the outputs models are close to the reference test data. We compute markedness for the test corpus (i.e., gold reference text), t_{mrk}, which is an n-dimensional vector, where n is a number of discourse relation types. We also compute bd^{+}_{mrk} and bd^{-}_{mrk} vectors for the outputs of the BART$_{D+}$ and BART$_{D-}$ models on the test corpus, respectively. Then, we calculate the mean square distances between markedness scores of the test corpus and produced ones, i.e., $\delta^{+} = MSQ(t_{mrk}, bd^{+}_{mrk})$ and $\delta^{-} = MSQ(t_{mrk}, bd^{-}_{mrk})$. We find that $\delta^{+} < \delta^{-}$, which means that the BART$_{D+}$ model output has markedness score at least as close to the test corpus as one of the BART$_{D-}$ model.

To see whether this difference between BART$_{D+}$ and BART$_{D-}$ is significant, we resort to the Stratified Approximated Randomization (AR) approach. We take the list of outputs of BART$_{D+}$ and BART$_{D-}$, call them $d^{+}_1, \ldots, d^{+}_k$ and $d^{-}_1, \ldots, d^{-}_k$, where k is the size of the test data set. For each i, we randomly assign to c_i either d^{+}_i or d^{+}_i, each with 0.5 probability. In this way we obtain a new list $c_1, \ldots, c_k$. We compute the markedness score for $c_1, \ldots, c_k$, call it c_{mrk}. Then, we calculate $\delta^{c} = MSQ(t_{mrk}, c_{mrk})$. We compare δ^{c} with δ^{+} and δ^{-}. We do this N (sufficiently large) number of times. If out of N checks, δ^{c} was less or equal to δ^{+} in p-percent of cases, we say that BART$_{D+}$ differs from BART$_{D-}$ with p-significance. (Usually, p is taken to be 5.)

D Discussion of Errors

We consider several examples of errors in D- models and compare them to the same outputs of the D+ model. This discussion is necessarily limited by the length of the outputs. We do not suggest these errors are representative of the models error in general, restricting ourselves to brief quali-

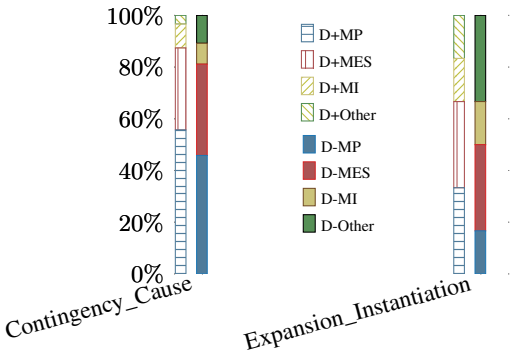


Figure 5: Continuous Explicit Connective Case: Graph for Models D+ and D- showing MP (Match Prediction); MES (Mismatch with Explicit Sister type); MI (Mismatch with Implicit); and Other (Explicit For Explicit Minus One, Minus Two, and Minus Three level types)

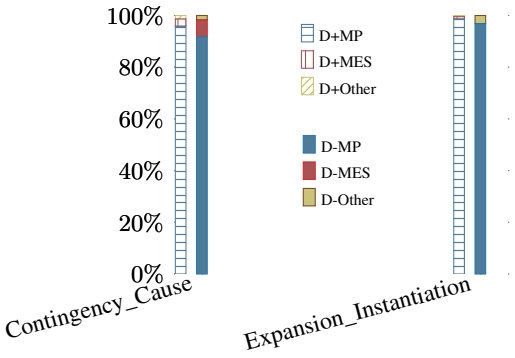


Figure 6: Continuous Case of Implicit Connectives: Graph for Models D+ and D- showing MP (Match Prediction), MES (Mismatch with Explicit Sister type), and Other (Explicit For Implicit Minus One, Minus Two, and Minus Three types)

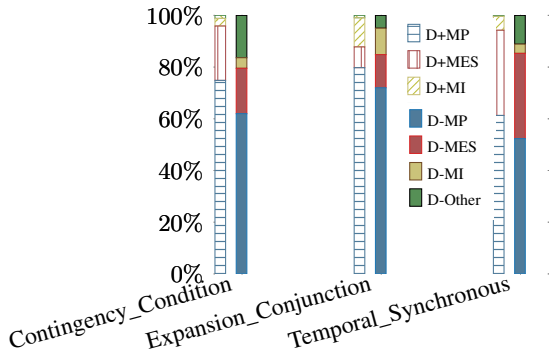


Figure 7: Ambiguous Explicit Connective Case: Graph for Models D+ and D- showing MP (Match Prediction); MES (Mismatch with Explicit Sister type); MI (Mismatch with Implicit); and Other (Explicit For Explicit Minus One, Minus Two, and Minus Three level types)

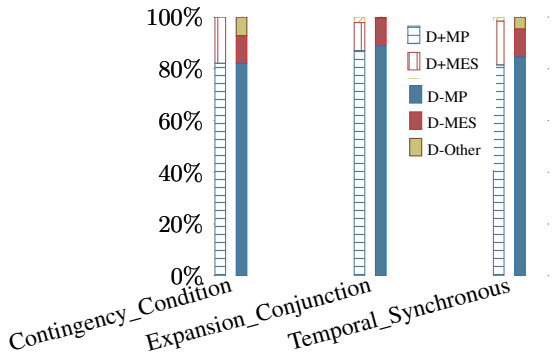


Figure 8: Ambigous Case of Implicit Connectives: Graph for Models D+ and D- showing MP (Match Prediction), MES (Mismatch with Explicit Sister type), and Other (Explicit For Implicit Minus One, Minus Two, and Minus Three types)

tative remarks which complement the quantitative results in the foregoing.

In Figure 9 both models mismatched with the intended temporal_synchronous relation, which is expressed by the connective *while* in the reference text. The D- model's choice produces much more of a hedged judgement of the threat by using *if* than either the reference connective *as long as* or the D+ connective *when*, which seems to require the existence of some time in which the threat is present.

In Figure 10 the D- model mismatched with the intended Comparison_Concession_Arg1-as-denier connective *even if*. The D- model's choice *unless* reverses the intended condition, erroneously suggesting that the banks obtaining financing could prevent British Air from rejecting the proposal described in the text. The D+ model predicts the connective *even if* which matches the reference and communicates the correct dependency between financing and British Air rejecting the proposal described in the text. We note that this is consistent with the results of (Stevens-Guille et al., 2020; Maskharashvili et al., 2021), who found comparison to be quite vexing for LSTM models.

In Figure 11 the D- model mismatched with the intended Expansion_Level-of-detail_Arg2-as-detail connective which is implicit. The D+ model correctly predicts the second sentence to simply provide further comment on the first sentence. Note though that the D- model's connective choice *but* is coherent in the text. This highlights that the two sentences, without the cue to the intended discourse relation, could be understood with respect to a variety of discourse relations.

BART$_{D+}$: Bush assured Roh that the U.S. would stand by its security commitments "*when* there is a threat" from Communist North Korea.
BART$_{D-}$: Bush assured Roh that the U.S. would stand by its security commitments "*if* there is a threat" from Communist North Korea.

Figure 9: Both models mismatch on Temporal_Synchronous, which is expressed by 'while' in the reference text.

BART$_{D+}$: But British Air, which was to have supplied $750 million out of $965 million in equity financing, apparently wasn't involved in the second proposal and could well reject it *even if* banks obtain financing.
BART$_{D-}$: But British Air, which was to have supplied $750 million out of $965 million in equity financing, apparently wasn't involved in the second proposal and could well reject it *unless* banks obtain financing.

Figure 10: D- mismatch on Comparison_Concession_Arg1-as-denier

BART$_{D+}$: The huge drop in UAL stock prompted one takeover stock trader, George KellNER, managing partner of Kellner, DiLeo & Co., to deny publicly rumors that his firm was going out of business. Mr. Kellner said that despite losses on UAL Stock, his firm's health is "excellent."
BART$_{D-}$: The huge drop in UAL stock prompted one takeover stock trader, George Kellner, managing partner of Kellners, DiLeo & Co., to deny publicly rumors that his firm was going out of business. *But* Mr. Kellner said that despite losses on UUAL stock, his firm's health is "excellent."

Figure 11: D- mismatch on Expansion_Level-of-detail_Arg2-as-detail

BART$_{D+}$: The National Cancer Institute also projected that overall U.S. mortality rates from lung cancer should begin to drop in several years *if* cigarette smoking continues to abate.
BART$_{D-}$: The National Cancer Institute also projected that overall U.S. mortality rates from lung cancer should begin to drop in several years *as* cigarette smoking continues to abate.

Figure 12: D- mismatch on Contingency_Condition_Arg2-as-cond

In Figure 12 the D- model mismatched with the intended Contingency_Condition_Arg2-as-cond connective *if*. The D- model's choice of the connective *as* implies that cigarette smoking will continue to abate, while the intended meaning is that the dropping of lung cancer mortality rates in the U.S. depends on cigarette smoking continuing to abate, which abatement, while projected, is not a foregone conclusion.

E Matching Explicit and Implicit Cases of Discontinuous, Continuous, and Ambiguous, Connectives: Figures

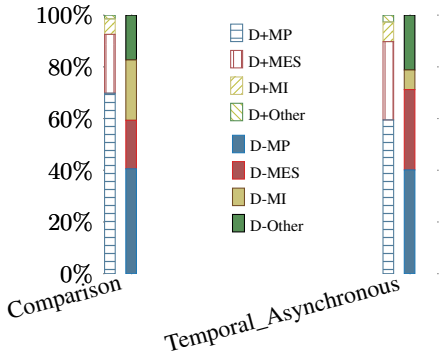


Figure 13: Discontinuous Explicit Connective Case: Graph for Models D+ and D- showing MP (Match Prediction); MES (Mismatch with Explicit Sister type); MI (Mismatch with Implicit); and Other (Explicit For Explicit Minus One, Minus Two, and Minus Three level types)

F Reproducibility Details

We use the pre-trained BART-Large HuggingFace transformer model for our baseline$_{D+}$.

We fine-tuned models, BART$_{D+}$ and $_{D-}$ on BART-Base transformer model. In total, there are 139421184 trainable parameters in this model. The models are fine-tuned using cross entropy loss

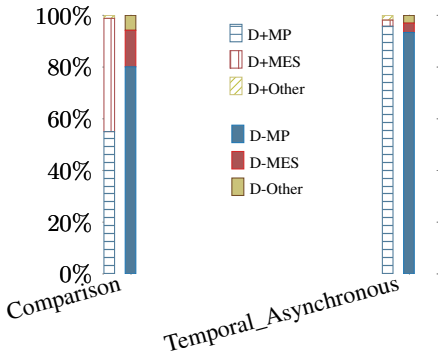


Figure 14: Discontinous Case of Implicit Connectives: Graph for Models D+ and D- showing MP (Match Prediction), MES (Mismatch with Explicit Sister type), and Other (Explicit For Implicit Minus One, Minus Two, and Minus Three types)

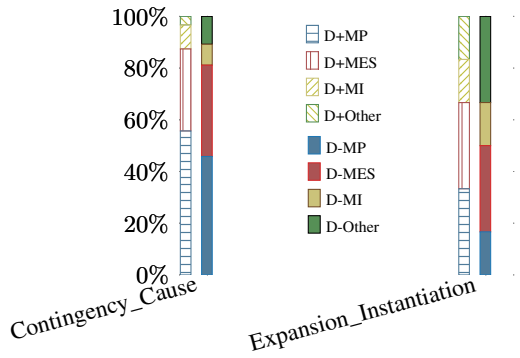


Figure 15: Continuous Explicit Connective Case: Graph for Models D+ and D- showing MP (Match Prediction); MES (Mismatch with Explicit Sister type); MI (Mismatch with Implicit); and Other (Explicit For Explicit Minus One, Minus Two, and Minus Three level types)

without label smoothing. The learning rate is constantly 2×10^{-5} and the batch size is 8 samples. The optimizer is Adam (Kingma and Ba, 2014) where $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$, and the weight decay is 0. The best checkpoint is selected by validation with patience of 10 training epochs. Computing infrastructure we used is made of NVIDIA V100 GPU and an Intel(R) Xeon(R) Platinum 8268 @ 2.90GHz CPU. Training on average took 15 epochs.

G BART-large selected results

We provide match results for BART-base versions of the full depth D+ and D- models in Table 11.

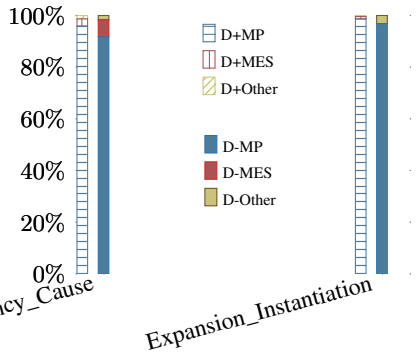


Figure 16: Continuous Case of Implicit Connectives: Graph for Models D+ and D- showing MP (Match Prediction), MES (Mismatch with Explicit Sister type), and Other (Explicit For Implicit Minus One, Minus Two, and Minus Three types)

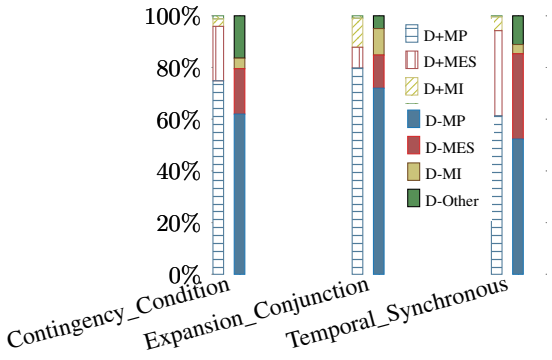


Figure 17: Ambiguous Explicit Connective Case: Graph for Models D+ and D- showing MP (Match Prediction); MES (Mismatch with Explicit Sister type); MI (Mismatch with Implicit); and Other (Explicit For Explicit Minus One, Minus Two, and Minus Three level types)

H Error Rate Examples

Figures 21, 22, and 23 exemplify D- Temporal_Asynchronous Nonsister, Comparison Nonsister, and Comparison Implicit errors respectively.

I Initial and Final Connective Examples

We provide an example of an initial connective generation by D- in Figure 24. A final connective generation by D- is provided in Figure 25, though we note that the reference is here implicit.

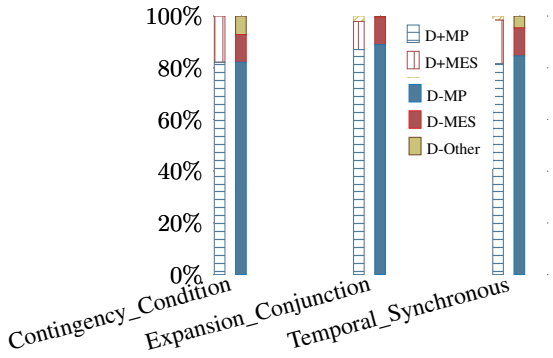


Figure 18: Ambigous Case of Implicit Connectives: Graph for Models D+ and D- showing MP (Match Prediction), MES (Mismatch with Explicit Sister type), and Other (Explicit For Implicit Minus One, Minus Two, and Minus Three types)

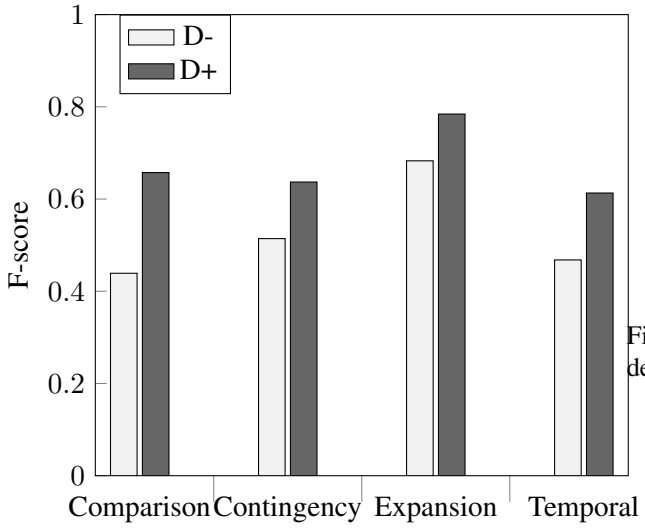


Figure 19: F-score for top level discourse relation types, case of explicit

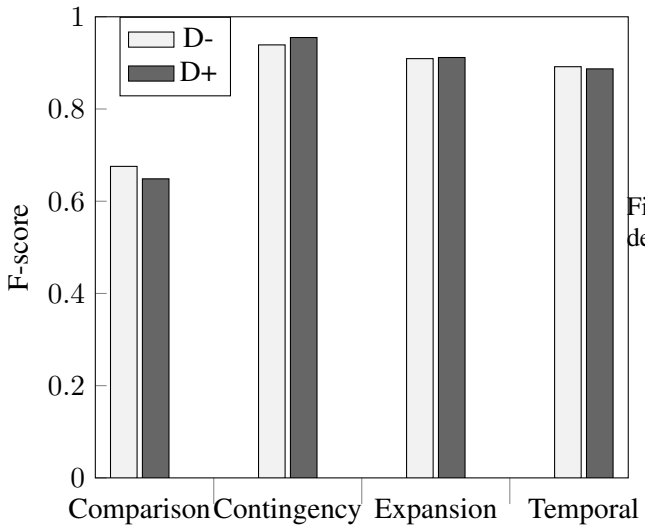


Figure 20: F-score for top level discourse relation types, case of implicit

Type	Depth	Order	FullOutput	Match
BART$_{D+}$	full	+	+	79.7%
BART$_{D+}$	full	+	-	82%
BART$_{D+}$	full	-	+	80.5%
BART$_{D+}$	full	-	-	74.5%
BART$_{D-}$		+	+	73.6%
BART$_{D-}$		+	-	75%
BART$_{D-}$		-	+	71.5%
BART$_{D-}$		-	-	74.7%

Table 11: BART-large fine-tuned selected results.

Figure 21: D- Temporal_Asynchronous_Precedence Nonsister Error

REFERENCE: That follows a more subtle decline in the prior six months **after** Manhattan rents had run up rapidly since 1986.
BART$_{D-}$:That follows a more subtle decline in the prior six months **because** Manhattan rents had run up rapidly since 1986.

Figure 22: D- Comparison_Concession_Arg1-as-denier Nonsister Error

REFERENCE: "There's quite a bit of value left in the Jaguar shares here *even though* they have run up" lately, says Doug Johnson, a fund manager for Seattle-based Safeco Asset Management.
BART$_{D-}$: "There's quite a bit of value left in the Jaguar shares here *and* they have run up" lately, says Doug Johnson, a fund manager for Seattle-based Safeco Asset Management.

Figure 23: D- Comparison_Concession_Arg2-as-denier Error

REFERENCE: But that ghost wasn't fooled; he knew the RDF was neither rapid nor deployable nor a force — *even though* it cost $8 billion or $10 billion a year.
BART$_{D-}$: But that ghost wasn't fooled; he knew the RDF was neither rapid nor deployable nor a force — it cost $8 billion or $10 billion a year.

Figure 24: D- Initial Connective Generation

REFERENCE: But that ghost wasn't fooled; he knew the RDF was neither rapid nor deployable nor a force – *even though* it cost $8 billion or $10 billion a year.
BART$_{D-}$: When Mr. Glass decides to get really fancy, he crosses his hands and hits a resonant bass note with his right hand.

Figure 25: D+ Final Connective Generation

REFERENCE: So far, analysts have said they are looking for $3.30 to $3.35 a share. After today's announcement, that range could increase to $3.35 to $3.40 a share.
BART$_{D-}$:So far, analysts have said they are looking for $3.30 to $3.35 a share. After today's announcement, that range could increase to $4.35 to $2.40 a share **however**.

Toward Self-Learning End-to-End Task-Oriented Dialog Systems

Xiaoying Zhang[1], **Baolin Peng**[2], **Jianfeng Gao**[2], **Helen Meng**[1]
[1]The Chinese University of Hong Kong, Hong Kong
[2]Microsoft Research, Redmond
{zhangxy, hmmeng}@se.cuhk.edu.hk
{bapeng, jfgao}@microsoft.com

Abstract

End-to-end task bots are typically learned over a static and usually limited-size corpus. However, when deployed in dynamic, changing, and open environments to interact with users, task bots tend to fail when confronted with data that deviate from the training corpus, *i.e.,* out-of-distribution samples. In this paper, we study the problem of automatically adapting task bots to changing environments by learning from human-bot interactions with minimum or zero human annotations. We propose SL-AGENT[1], a novel self-learning framework for building end-to-end task bots. SL-AGENT consists of a dialog model and a pre-trained reward model to predict the quality of an agent response. It enables task bots to automatically adapt to changing environments by learning from the unlabeled human-bot dialog logs accumulated after deployment via reinforcement learning with the incorporated reward model. Experimental results on four well-studied dialog tasks show the effectiveness of SL-AGENT to automatically adapt to changing environments, using both automatic and human evaluations. We will release code and data for further research.

1 Introduction

The most common approach of building end-to-end task-oriented dialog systems is to train neural models to imitate human behaviors in fixed task-specific annotated corpora (Gao et al., 2018; Zhang et al., 2020). Existing state-of-the-art approaches usually adopt Pre-trained Language Models (PLMs) (Peng et al., 2020a; Ham et al., 2020; Hosseini-Asl et al., 2020) to build end-to-end dialog systems. However, these data-driven approaches assume an independent and identically distributed (IID) data setting[2], *i.e.,* a static environment[3], and usually exhibit a tendency of failure, when confronted with out-of-distribution (OOD) examples in real-world scenarios, *i.e.,* changing environments.

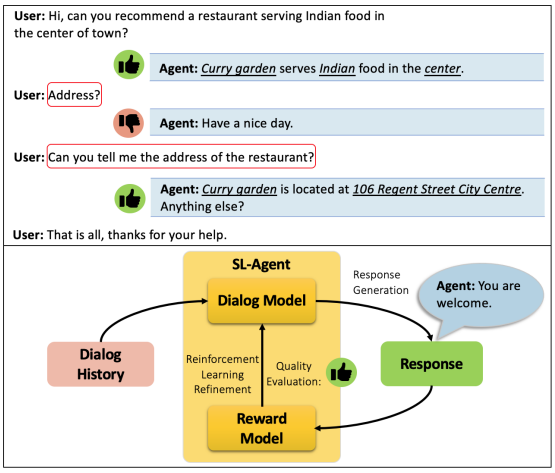


Figure 1: Illustration of the proposed SL-AGENT with a human-bot dialog example. (i) The human-bot dialog example, containing an inappropriate response related to unseen user behaviors (upper part). (ii) Demonstration of the refining process in SL-AGENT with the exhibited dialog example (lower part).

In the context of task-oriented dialog systems, changing environments are quite common and arise from the following two aspects: (i) *unseen user behaviors* – real users may query with unseen language patterns and unknown user goals (*i.e.,* unseen slot values and dialog flows) of the designated tasks outside the pre-built training corpora (Liu et al., 2018; Peng et al., 2020b). For example, real users may query entities in the database but not covered by the training examples. (ii) *task definition extensions* – dialog systems need to handle new functions or new tasks as user and business requirements evolve, *i.e.,* add new slot types (Lipton et al., 2018; Gasic et al., 2014). For example, a restaurant bot designed for the table-booking service may also encounter queries about delivery service after deployment. These human-bot interactions accu-

[1]SELF-LEARNING AGENT.
[2]Assume the same user behaviors at deployment as in the training stage.
[3]Environment is the Agent's world in which it lives and interacts.

Proceedings of the SIGdial 2022 Conference, pages 516–530
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

mulated after deployment are cheap, dynamic and contain useful information (Hancock et al., 2019), *i.e.,* unseen user behaviors are related to the training examples and the probabilistic dialog model may generate appropriate responses. As shown in the upper part of Figure 1, when user queries casually about address, the system fails to provide address in the second response, but gives it in the third response, when user queries in a detailed way (similar to the training examples). Therefore, rather than merely imitating human behaviors in a fixed corpus, task bots are desired to spontaneously learn from the interactions with real users, progressively improve and adapt after being deployed in dynamic and constantly changing environments.

There are several attempts to leverage human-bot interactions to improve task bots in changing environments. For example, Liu et al. (2018); Shah et al. (2018); Dai et al. (2020) propose to query humans for adequate feedback scores or annotations. However, it relies on human annotations or user feedback, which can be costly and sometimes users are unwilling to give any feedback. In addition, these works center on dialog policy optimization or retrieval-based task bots. Automatically adapting task bots to changing environments is imperative for end-to-end dialog model yet under-explored. Furthermore, these works usually omit task definition extensions.

In this paper, we propose SL-AGENT, a novel self-learning framework for building end-to-end task bots in a more realistic changing environment setting with minimum or zero human annotations. It consists of a neural dialog model and a pre-trained reward model, where the dialog model generates responses and the reward model judges the quality of agent responses. Specifically, we devise a data augmentation strategy to construct positive and negative examples based on the given dialog training corpus to endow the reward model with the capability to judge the quality of responses for unlabeled human-bot dialog logs. The bot (including dialog model and reward model) is first trained with the same available training data, then deployed to converse with real users and collect human-bot dialog logs. After that, as shown in the lower part of Figure 1, the bot is refined with the unlabeled human-bot dialog logs via reinforcement learning, where the response quality is judged by the reward model. In this way, the bot can automatically adapt to unseen user behaviors, without extra human annotations. Regarding the problem of extensions in task definitions, machine teaching is utilized to correct representative failed dialogs with minimum human annotations to provide necessary instructions on how to handle new functions. After that, the bot quickly adapts to new functions through the self-learning procedure.

Our contributions are summarized as below:

- We propose a new research problem *i.e.,* how to enable task bots to automatically adapt themselves to changing environments by learning from interactions with minimum or zero human annotations.
- We propose a novel self-learning framework SL-AGENT that equips with a pre-trained reward model trained by the devised data-augmentation strategy to build generative end-to-end task bots in a realistic changing environment setting, with minimum or zero human annotations.
- We conduct comprehensive experiments on four datasets to demonstrate the effectiveness of SL-AGENT for enabling automatic adaptation to changing environments by learning from the unlabeled human-bot dialog logs using both automatic and human evaluations.

2 Related Work

RL for Dialog Policy Learning Reinforcement learning has been widely applied to dialog systems for policy optimization. Young et al. (2013); Peng et al. (2018, 2017); Liu and Lane (2017); Gasic et al. (2014); Tseng et al. (2021) formulate dialog policy learning as a sequential problem and use REINFORCE (Williams, 1992) and/or Q-learning (Watkins and Dayan, 1992) to optimize the dialog policy. SL-AGENT utilizes a similar REINFORCE algorithm but focuses on generative end-to-end optimization.

Adapting to Changing Environments for Dialog Systems Several attempts have been made to deal with changing environments after deployment. Rajendran et al. (2019); Dai et al. (2020) propose to learn from the human-bot interactions but requires lots of human corrections. Shah et al. (2018); Liu et al. (2018); Gašić et al. (2011); Gasic et al. (2014) propose to learn from human-bot interactions via reinforcement learning based on the queried human feedback scores after each dialog. To reduce the efforts of querying humans, Su et al. (2016)

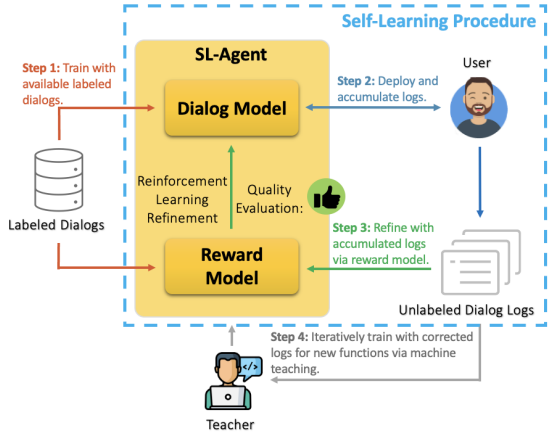


Figure 2: Training pipeline of the proposed SL-AGENT.

introduces a session-level Bi-LSTM reward model trained with extra pre-collected classification corpus to predict the task success of each dialog. Nevertheless, session-level reward model may underestimate the quality of responses in single dialog turns. Different from the works mentioned above, SL-AGENT leverages a turn-level pre-trained reward model built on the given dialog corpus using the devised data augmentation approach and focuses on generative end-to-end dialog systems. Another line of research is using data-augmentation methods to generate diverse user behaviors during the training stage (Gao et al., 2020; Li et al., 2020b). Additionally, Madotto et al. (2020); Liu et al. (2021) continually collect extra labeled data to train task bots but aim to overcome the catastrophic forgetting problem, which is a different research topic (*i.e.,* continual learning) from our paper.

3 SL-AGENT

3.1 Overview

As depicted in Figure 2, SL-AGENT contains two components: (i) a dialog model for generating responses (Section 3.2); (ii) a pre-trained reward model for judging the quality of agent responses and outputting reward scores to guide the refinement of the dialog model (Section 3.3). Specifically, SL-AGENT operates in the following steps: (i) First, the bot (both dialog model and pre-trained reward model) is fine-tuned with the same available annotated task-specific dialogs. (ii) Then, the bot is deployed online to converse with users and accumulate unlabeled human-bot dialog logs. (iii) Next, the dialog model is refined with these human-bot dialog logs via reinforcement learning, using the reward scores from the reward model (Section 3.4). (*iv*) For task definition extensions, machine teaching is utilized to correct representative failed dialogs to provide instructions on how to handle new functions (Section 3.5). After that, the bot further improves through the self-learning procedure.

3.2 Dialog Model

SL-AGENT is a general framework that is compatible with any generative end-to-end dialog models (Peng et al., 2020a; Ham et al., 2020; Hosseini-Asl et al., 2020). In this paper, we employ SOLOIST (Peng et al., 2020a), a pre-trained end-to-end dialog model, resulting in an agent termed SL-SOLOIST[4].

We briefly review SOLOIST for completeness. SOLOIST formulates the end-to-end dialog generation as a sequence generation problem, by sequentially concatenating the inputs and outputs of 4 dialog modules (*i.e.,* NLU, DST, POL, NLG) in a typical dialog system. Each dialog turn is represented as:

$$\boldsymbol{x} = (\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{r}), \tag{1}$$

where $\boldsymbol{s}$ is the entire dialog history, $\boldsymbol{b}$ is the annotated belief state, $\boldsymbol{c}$ refers to DB state fetched from database, and $\boldsymbol{r}$ is the delexicalized agent response. SOLOIST employs a Transformer-based model with parameters $\boldsymbol{\theta_D}$ to characterize the sequence generation probability $p_{\boldsymbol{\theta_D}}(\boldsymbol{x})$. Initialized with GPT-2 (Radford et al., 2019), the model is pre-trained on large-scale annotated dialog corpora, and then fine-tuned with limited task-specific dialogs.

Synthetic Dialog Construction. To identify user behaviors with unseen slot values, we propose to synthesize dialog examples by exhausting database (DB) values and substitute corresponding slot values of in the training set. Specifically, for each dialog turn $\boldsymbol{x}$, we replace slot values in the utterances and user goal with corresponding new values of the randomly sampled DB entry.

3.3 Reward Model

The human-bot dialog logs accumulated after deployment may contain previously unseen user behaviors with unseen language patterns and unknown user goals. To enable the dialog model to identify these new types of user inputs to which the previously trained system cannot respond appropriately, we propose a reward model that judges the

[4]In this paper, SL-AGENT refers to the proposed framework and SL-SOLOIST is an instance of it, which utilizes SOLOIST as its dialog model.

quality of an agent response through a reward score (a positive reward for an appropriate response, a negative reward for an inappropriate response).

We formulate the quality evaluation problem as a binary classification task. Dialog responses are jointly determined by the dialog history, generated belief state, and fetched DB state. Therefore, given the training data $\mathcal{D}$ (annotated with belief states and DB states), we build a turn-level reward model R, which is parameterized by a Transformer $\boldsymbol{\theta_R}$ with the input dialog turn sequence $\boldsymbol{x}$, defined as Equation 1 to characterize the classification probability: $p_{\boldsymbol{\theta_R}}(\boldsymbol{x}) = p_{\boldsymbol{\theta_R}}(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{r})$.

The reward model R is trained using contrastive objective to discriminate between an appropriate response (*i.e.,* positive example $\boldsymbol{x}$) and an inappropriate response (*i.e.,* negative example $\hat{\boldsymbol{x}}$), given the dialog history. Specifically, for each dialog turn, we construct several positive examples $\{\boldsymbol{x}_m\}_{m=1}^{M}$ and negative examples $\{\hat{\boldsymbol{x}}_{\boldsymbol{n}}\}_{n=1}^{N}$ based on the sequence $\boldsymbol{x}$, to add the relevance of real-world scenarios and endow the reward model with the capability of evaluating the response quality. Then we apply a binary classifier on top of the output sequence representation from the Transformer to discriminate between a positive example $\boldsymbol{x}$ ($y = 1$) and a negative example $\hat{\boldsymbol{x}}$ ($y = 0$). The training objective for a single example in the training set $\mathcal{D}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\theta_R}} =& y \sum_{m=1}^{\mathcal{M}} \log\left(p_{\boldsymbol{\theta_R}}(\boldsymbol{x}_m)\right) \\ &+(1-y) \sum_{n=1}^{\mathcal{N}} \log\left(1 - p_{\boldsymbol{\theta_R}}\left(\hat{\boldsymbol{x}}_{\boldsymbol{n}}\right)\right),\end{aligned} \tag{2}$$

Positive Examples. For each dialog turn, we consider two kinds of user utterances: (i) the original user utterance in the training set $\mathcal{D}$, to identify the appropriate response to the user behavior; (ii) the paraphrased user utterances generated based on the original user utterance using back translation (Edunov et al., 2018), to enhance the ability of reward model for identifying user behaviors with diverse language patterns.

Negative Examples. Based on the analysis on 200 human-bot dialog logs collected from the evaluation platform of DSTC8 Track 1 challenge (Li et al., 2020a)[5], we summarize 5 types of dialog turns that have inappropriate responses (in Appendix J). Then, for each dialog turn in the training data $\mathcal{D}$, we construct negative examples $\hat{\boldsymbol{x}}$ (in brackets) according to these 5 types:

- **Repetition** The dialog model failed to understand the user's repeated query and generated the same response twice. (Repeating the response from the previous turn.)
- **Inconsistency** The dialog model generated an incoherent response. (Randomly sampling a response from the dataset $\mathcal{D}$ to replace the original response .)
- **Partial Information** The dialog model partially understood user request and answered incompletely. (For those user utterances with multiple slots request, randomly dropping a slot answer in the original response.)
- **Non-fluency** The dialog model generated a non-fluent response. (Randomly repeating some word tokens in the original response.)
- **Misunderstanding** The dialog model generated the incoherent belief state and response. (Randomly sampling a belief state and response from the dataset $\mathcal{D}$ to replace the original belief state and response.)

To boost the model performance with limited annotated task-specific corpora, we propose to follow the pre-training and fine-tuning paradigm to build the reward model, *i.e.,* pre-train the reward model using large-scale annotated heterogeneous dialog corpora, then fine-tune the pre-trained reward model with annotated task-specific data using the same training objective. The pre-training corpora is Schema dataset (Rastogi et al., 2019).

3.4 Refine with Reinforcement Learning

The interactions between the agent and users can be modeled as a sequential decision problem. As such, the dialog model can be refined via the REINFORCE algorithm (Williams, 1992). The policy is the trained dialog model $p_{\boldsymbol{\theta_D}}(\boldsymbol{x})$, the initial state is the dialog history $\boldsymbol{s}$, and the action space corresponds to the vocabulary set $\mathcal{V}$. The reward perceived by the dialog model is $R\left(\boldsymbol{s}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{r}\right)$ from the reward model. The parameters $\boldsymbol{\theta_D}$ are updated by maximizing the cumulative reward score. The refining procedure is described in detail as follows:

For each RL episode, we randomly sample a dialog turn with dialog history and delexicalized

[5]These human-bot dialog logs contain the evaluation scores and comments from Amazon Mechanical Turks.

response. We run the dialog model to generate belief state $\hat{\boldsymbol{b}}$, based on the input dialog history sequence $\boldsymbol{s}$. At each time step t, we sample a token $\hat{b}_t$ according to the model distribution, where the logits' distribution of the model is first filtered using Nucleus (top-p) filtering (Holtzman et al., 2019), then redistributed via softmax function. Then we retrieve DB state $\hat{\boldsymbol{c}}$ from the database using $\hat{\boldsymbol{b}}$, and sample the delexicalized response sequence $\boldsymbol{r}$ following same sampling procedure, based on the token sequence $(\boldsymbol{s}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}})$. Note that the delexicalized response is given as part of the input. Then we feed the concatenation of dialog history $\boldsymbol{s}$, generated belief state $\hat{\boldsymbol{b}}$, retrieved DB state $\hat{\boldsymbol{c}}$ and the response $\boldsymbol{r}$, i.e. $(\boldsymbol{s}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \boldsymbol{r})$ into the reward model $p_{\boldsymbol{\theta_R}}(\boldsymbol{x})$ to obtain the reward score $R(\boldsymbol{s}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \boldsymbol{r})$. The positive reward is 1, negative reward is -1. The training objective for a single example is represented as:

$$\mathcal{L}_{\theta_D} = -\sum_{t=1}^{T_{\hat{b}}} \log p_{\boldsymbol{\theta_D}}\left(\hat{b}_t \mid \hat{b}_{<t}, \boldsymbol{s}\right) \times R(\boldsymbol{s}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \boldsymbol{r}) - \sum_{t=1}^{T_r} \log p_{\boldsymbol{\theta_D}}\left(r_t \mid r_{<t}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \boldsymbol{s}\right) \times R(\boldsymbol{s}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \boldsymbol{r}), \tag{3}$$

where the length of generated belief state and input delexicalized response are $\boldsymbol{T_{\hat{b}}}$, $\boldsymbol{T_r}$, respectively. Algorithm 1 (in Appendix A) summarizes the self-learning-based RL refining framework for refining the dialog model.

3.5 Minimum annotations via Machine Teaching

To handle the queries about new functions in additional dialog turns, we need to introduce new slot-value pairs, action templates, *etc.*(An example is in Appendix G.) Machine teaching is an efficient approach to training task bots (Simard et al., 2017; Williams and Liden, 2017). In this paper, we implement machine teaching via Conversational Learner (CL) (Shukla et al., 2020). The teaching process is conducted in three steps: (i) The trained task bot is deployed online to fulfill the given goals by interacting with real users, leaving a handful of human-bot dialog logs. (ii) Human experts select a few representative failed dialogs to construct training examples with new functions by adding new action templates, introducing new slot-value pairs, correcting inappropriate responses and annotations (*i.e.,* belief states). (iii) The deployed task bot (*i.e.,* both dialog model and reward model) is trained on these training examples to handle new functions.

Domain	Attraction	Train	Hotel	Restaurant
#Train	50	50	50	50
#Valid	50	50	50	50
#Test	100	200	200	200

Table 1: Data statistics of four single-domain dialog datasets (Peng et al., 2020a; Budzianowski et al., 2018).

4 Experiments

4.1 Experimental Setup

We validate the efficiency and flexibility of proposed SL-AGENT on four different end-to-end dialog tasks using MultiWOZ single-domain dialog datasets (Budzianowski et al., 2018), reorganized by Peng et al. (2020a). Data statistics are shown in Table 1. Based on above datasets, we construct two settings to represent the changing environments – **Setting I** for unseen user behaviors and **Setting II** for task definition extensions.

Implementation Details. To implement the proposed reward model, we conduct experiments with several Transformer-based models and GPT-2 (Radford et al., 2019) (enhanced with auxiliary generation task) shows better performance than others. Therefore, we implement proposed reward model using GPT-2-117M and the multi-task training objective. Full details are in Appendix B.

Automatic Evaluation Metrics. We report the results using the same automatic evaluation metrics following Budzianowski et al. (2018): (i) `Inform`(%) evaluates whether the agent returns an appropriate entity. (ii) `Success`(%) judges whether the agent correctly answers all requested attributes. (iii) `BLEU`(%) measures the word overlap of the generated response against human response. (iv) `Combined`(%) assesses the overall quality, which is defined as: `Combined` = (`Inform` + `Success`) $\times$ 0.5 + `BLEU`.

Human Evaluation Metrics. Following the same evaluation protocol in the DSTC9 Track 1 challenge (Gunasekara et al., 2020), we conduct human evaluations to judge the agent quality. For each dialog session, Amazon Mechanic Turkers are presented with a goal and instructions, then they are required to converse with agent to achieve the goal via natural language. At the end of each dialog session, Turks are required to assess the overall dialog quality using the following five metrics: (i) `Success w/o g`(%) judges whether the agent completes the task. (ii) `Success w/ g`(%)

Model	Attraction			Train			Hotel			Restaurant		
	Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU
SOLOIST$_5$	27.00	14.00	4.07	72.73	32.32	5.43	25.00	3.50	2.93	26.50	2.00	4.71
SOLOIST$_S$	60.00	33.00	8.14	73.74	54.55	6.94	56.00	29.50	7.05	62.50	41.50	7.33
SOLOIST+PARG	60.00	32.00	8.83	75.25	56.06	8.45	58.00	29.00	7.71	64.00	42.00	9.17
SOLOIST-OA	61.00	36.00	8.66	74.75	55.05	7.58	56.50	29.00	7.14	64.50	42.50	8.56
SL-SOLOIST	**64.00**	**40.00**	**8.99**	**75.76**	**61.62**	**10.97**	**60.50**	**39.50**	**8.34**	**75.00**	**44.50**	**10.60**
SOLOIST-TH	66.00	41.00	9.01	77.27	62.87	10.70	60.00	42.50	9.82	70.50	46.00	11.76
SOLOIST$_{50}$	86.00	65.00	12.90	80.81	64.65	9.96	74.50	43.50	8.12	81.00	55.50	12.80

Table 2: End-to-end evaluation results on four tasks. The forth to sixth rows indicate the results of refining with 45 simulated (unlabeled) human-bot dialog logs, based on SOLOIST$_S$. SOLOIST$_{50}$ is quoted from Peng et al. (2020a). (SL-SOLOIST significantly outperforms all baselines in mean with p<0.01 based on Combined.)

Model	Attraction			Train			Hotel			Restaurant		
	Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU
SOLOIST$_S$	60.00	33.00	8.14	73.74	54.55	6.94	56.00	29.50	7.05	62.50	41.50	7.33
SOLOIST-OA	63.00	34.00	8.66	77.78	55.05	8.13	58.50	30.00	7.08	63.00	42.00	10.03
SL-SOLOIST	**70.00**	**36.00**	**8.68**	**78.28**	**60.10**	**9.06**	**62.00**	**33.50**	**7.39**	**70.00**	**45.00**	**10.93**
SOLOIST-TH	68.00	40.00	9.01	76.77	62.63	9.55	62.50	35.50	7.83	70.50	47.50	11.36

Table 3: Automatic evaluation results on four tasks in Real-Scenario Setting. The first row refers to previously reported SOLOIST$_S$. The last three rows refer to refining with 30 real (unlabeled) human-bot dialog logs based on SOLOIST$_S$. (SL-SOLOIST significantly outperforms all baselines in mean with p<0.01 based on Combined.)

judges whether the agent completes the task and provides matched slot values against the database record. (*iii*) `Understanding`(1-5) measures the understanding correctness of user utterances. (*iv*) `Appropriateness`(1-5) indicates the appropriateness, naturalness, and fluency of an agent response. (*v*) `Turns` reports the average number of dialog turns for successful dialog sessions.

Compared Methods. To demonstrate the effectiveness of the proposed reward model in SL-AGENT, we use SOLOIST as the dialog model to compare the performance of different methods.

- SOLOIST$_5$ is trained with 5 labeled dialogs, randomly sampled from the train set.
- SOLOIST$_S$ is trained using synthetic dialogs constructed from the 5 labeled dialogs used for training SOLOIST$_5$.
- SOLOIST+PARG is trained on SOLOIST$_S$ with paraphrased dialogs (Gao et al., 2020; Edunov et al., 2018) constructed from the 5 labeled dialogs, *i.e.,* data-augmentation baseline for adapting to unseen user behaviors.
- SOLOIST-OA is refined with unlabeled human-bot dialog logs based on SOLOIST$_S$ using the session-level reward of task success from online activate reward model (trained using the same 5 labeled dialogs as SOLOIST$_5$) and partially queried session-level human feedback score (Su et al., 2016).
- SL-SOLOIST (Ours) is refined with unlabeled human-bot dialog logs based on SOLOIST$_S$ using the pre-trained reward model in SL-AGENT, which is fine-tuned using the same 5 labeled dialogs as SOLOIST$_5$. Machine teaching is not utilized by now[6].
- SOLOIST-TH is refined with unlabeled human-bot dialog logs based on SOLOIST$_S$ using queried turn-level human feedback score, which is an upper bound.
- SOLOIST$_{50}$ is trained with whole 50 labeled dialogs, which can be regarded as the result of sufficient human corrections, *i.e.,* the highest bound. (Details are shown in Appendix C.)

4.2 Results of Setting I - Unseen User Behaviors

Simulation Evaluation Setup. Deploying a trained agent to interact with real human users and collect dialog logs is labor-intensive and costly for experimental purposes. Hence, we construct a setting to simulate unseen user behaviors. We randomly sample 5 dialogs from the training set as labeled data to train a task bot (*i.e.,* both dialog model and reward model). Note that the remaining 45 dialogs contain unseen user behaviors with unseen

[6]To better demonstrate the self-learning capability of SL-AGENT, machine teaching is only used in the setting of task definition extensions. However, machine teaching can be optionally used to update the bot for better performance in the setting of unseen user behaviors.

language patterns and unknown user goals. Hence, it is applicable to simulate unseen user behaviors by modifying the remaining 45 dialogs as unlabeled imperfect human-bot dialog logs (through adding noise, *i.e.,* corrupting responses[7]). These 45 unlabeled human-bot dialog logs are further used for refining SOLOIST$_S$, resulting in SOLOIST-OA, SL-SOLOIST, SOLOIST-TH. This simulation setting allows us to perform a detailed analysis of the reward model in SL-AGENT without much cost and easily reproduce the experimental results.

Simulation Evaluation Results. The end-to-end evaluation results on four different tasks are presented in Table 2. SOLOIST$_S$ significantly outperforms SOLOIST$_5$ over all evaluation metrics on all tasks, which shows the effectiveness of the proposed synthetic dialog construction for identifying user behaviors with unseen slot values. SL-SOLOIST outperforms SOLOIST+PARG over all the metrics, which demonstrates the higher efficiency of directly learning from human-bot dialog logs. We observe that SL-SOLOIST outperforms SOLOIST-OA by a large margin, and achieves comparable performance with SOLOIST-TH (refining with turn-level human feedback score, *i.e.,* the upper bound). This shows the strong capability of the turn-level pre-trained reward model in SL-AGENT for predicting the quality of responses. We conjecture that our proposed reward model trained with the proposed data-augmentation strategy is more robust to unseen user behaviors and thus ports richer useful information to dialog models. The results verify the vast potential of the proposed SL-AGENT, allowing the bot to automatically adapt to unseen user behaviors without extra human annotations. Results of further policy improvement are shown in Appendix E.

Real-Scenario Evaluation Setup. Simulation setting allows effortless experimental studies to validate the effectiveness of the reward model in SL-AGENT. However, the results are likely biased. Therefore, in the real-scenario setting, we deploy SOLOIST$_S$ online and recruit human users to converse with it. We collect 30 real (unlabeled) human-bot dialog logs to refine SOLOIST$_S$, resulting in the agent SOLOIST-OA, SL-SOLOIST, SOLOIST-TH.

Real-Scenario Evaluation Results. The evaluation results on four tasks are shown in Table 3.

[7]Note that the associated labels of belief states are not used. Construction details are in Appendix D.

Model	Restaurant-Ext			
	Inform	Success	BLEU	Combined
SOLOIST$_S$	54.00	0.00	6.42	33.42
SOLOIST$_S$+TEACH	64.00	18.00	9.34	50.34
SL-SOLOIST+TEACH	**68.00**	**24.00**	**11.76**	**57.76**
SOLOIST-TH+TEACH	68.50	26.00	11.88	59.13

Table 4: Automatic evaluation results on task definition extensions. (Difference in mean is significant with p<0.01 based on Combined.)

We observe that SL-SOLOIST refined using the reward model in SL-AGENT outperforms other methods over all evaluation metrics on all tasks. Furthermore, SL-SOLOIST achieves comparable performance with SOLOIST-TH, even achieves better performance on certain metrics. We conclude that the results of real-scenario evaluation and simulation evaluation are consistent, confirming that SL-SOLOIST enables effective self-learning after deployment by learning from interactions.

4.3 Results of Setting II – Task Definition Extensions

Setup. We follow the domain extension experiment setting in Lipton et al. (2018) to assess the ability of SL-SOLOIST to quickly handle task definition extensions. We extend existing `Restaurant`, denoted as `Restaurant-Ext`, with additional functions by introducing 4 new slots, *i.e., [restaurant_dish]*, *[value_price]*, *[start_time]*, *[end_time]* in added dialog turns (in Appendix G), and corresponding values for each DB entry (in Appendix H). The first slot is about the restaurant's signature dish, and the last three are related to delivery service. We leverage Conversational Learner (CL) (Shukla et al., 2020), a practical machine teaching tool, to visualize and select dialogs for constructing training examples on the `Restaurant-Ext` domain by providing corrections and introducing new slots. Finally, 10 examples are obtained through machine teaching for training, 50 for validating and 50 for testing. We fine-tune the dialog model SOLOIST$_S$ and the previously trained reward model[8], using 10 corrected dialogs, resulting the agent denoted as SOLOIST$_S$+TEACH. Then, SOLOIST$_S$+TEACH is deployed to converse with real human to collect 20 real (unlabeled) human-bot dialog logs, which are then used to refine itself, resulting in SL-SOLOIST+TEACH. To better show the effective-

[8]The reward model used for obtaining SL-SOLOIST in the Table 2. It is trained with 5 labeled dialogs in the train set.

Model	Restaurant				
	SR w/o g	SR w/ g	Under.	Appr.	Turns
SOLOIST$_S$	31.82	29.54	3.86	4.13	10.00
SOLOIST-OA	33.42	30.86	3.89	4.12	9.97
SL-SOLOIST	**43.10**	**36.21**	**3.97**	**4.13**	**9.89**

Table 5: Human evaluation results. SR w/o g: Success rate without grounding, SR w/ g: Success rate with grounding, Under.: Understanding score, Appr.: Appropriateness score.

ness of the reward model in SL-AGENT, we also report the result of SOLOIST-TH+TEACH, which is refined using the turn-level human feedback score.

Results. The evaluation results are presented in Table 4. We observe that SOLOIST$_S$ has zero success rate, which is predictable as it does not have any knowledge of the new functions. SOLOIST$_S$+TEACH outperforms the baseline by 17 points in terms of Combined score, which exhibits the effectiveness of machine teaching for handling new functions. SL-SOLOIST+TEACH lifts the Combined score by approximately 7 points, achieving comparable performance with SOLOIST-TH+TEACH. The results show that SL-SOLOIST+TEACH can adapt to new tasks and continually improve itself by automatically learning from the interactions, revealing, with minimum annotations from machine teaching, SL-AGENT enables flexible adaptations to new functions.

4.4 Interactive Human Evaluation

Setup. We conduct human evaluations to evaluate the performance of SOLOIST$_S$, SOLOIST-OA, SL-SOLOIST interacting with human users, following the evaluation protocol in DSTC9 track 1 challenge (Gunasekara et al., 2020), with 100 Turkers involved and 100 dialogs gathered for analysis, respectively.

Results. The human evaluation results on Restaurant domain are presented in Table 5. The results show that SL-SOLOIST outperforms SOLOIST$_S$, SOLOIST-OA over all the metrics, which are consistent with the automatic evaluation results. The significant improvement on two success rate metrics, especially success rate with grounding, verifies the effectiveness of the reward model in SL-AGENT for refining the dialog agent after deployment without additional human annotations, as it more adequately reflects the system's capability of completing tasks in real scenarios. Two interactive examples are in Appendix F.

Reward model	Restaurant			
	Inform	Success	BLEU	Combined
GPT-2	67.00	41.50	9.30	63.55
BERT	68.00	42.50	9.55	64.80
BERT-Large	66.00	44.00	11.09	66.09
RoBERTa	72.00	45.00	9.23	67.73
RoBERTa-Large	69.50	46.50	10.20	68.20
SL-SOLOIST	**75.00**	**44.50**	**10.60**	**70.35**

Table 6: Ablation study results on using different PLMs for reward models. (Difference in mean is significant with p<0.01 based on Combined.)

4.5 Ablation Study

Impact of different PLMs for reward models. We conduct ablation studies on Restaurant domain to analyze the influence of choosing different PLMs and multi-task training objective on the reward model. We choose several popular PLMs including BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). Note that all the models share the same pre-training and fine-tuning procedure, except that BERT and RoBERTa are trained with quality prediction task while SL-SOLOIST is optimized using multi-task learning. We show in Table 6 that RoBERTa performs better than BERT. GPT-2 (on which SL-SOLOIST is built) trained with single quality prediction task, yields significantly worse performance than other methods. We speculate that bidirectional Transformer encoder enables BERT and RoBERTa to capture richer context information. SL-SOLOIST achieves consistent performance improvements over all the metrics, showing the effectiveness of multi-task learning for the reward model.

5 Conclusion

In this paper, we propose a new research problem *i.e.,* how to enable task bots to automatically adapt themselves to changing environments by learning from interactions with minimum or zero human annotations. In addition, we propose SL-AGENT, a novel self-learning framework. We verify its effectiveness on automatically adapting to changing environments on four dialog tasks by learning from the unlabeled human-bot dialog logs via reinforcement learning with an incorporated pre-trained reward model. As for future work, there are more ways that a task bot could learn to improve itself, *e.g.,* during machine teaching, human experts could provide not only correct labels but also feedback in natural language. We leave the theme of effective machine teaching to future work.

6 Ethical Considerations

During the collection, annotation and evaluation procedure of the human-bot dialog logs, all involved Amazon Mechanic Turkers and human annotators have been informed of the research purpose in advance, and any of their privacy will not be disclosed or violated during the research period. All other used datasets are open-sourced datasets. In summary, we abide by all research ethics.

7 Acknowledgements

This research is affiliated with the CUHK MoE-Microsoft Key Laboratory for Human-centric Interface Technologies. The project is partially sponsored by a grant from the HKSAR Research Grants Council General Research Fund (project number 14207619). In addition, we would like to thank Yifei Yuan, Kun Zhang, Kun Li and Jingyan Zhou in particular for their insightful comments and persevering support.

References

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Yinpei Dai, Hangyu Li, Chengguang Tang, Yongbin Li, Jian Sun, and Xiaodan Zhu. 2020. Learning low-resource end-to-end goal-oriented dialog for fast and reliable system deployment. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 609–618.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1371–1374.

Silin Gao, Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Paraphrase augmented task-oriented dialog generation. *arXiv preprint arXiv:2004.07462*.

Milica Gašić, Filip Jurčíček, Blaise Thomson, Kai Yu, and Steve Young. 2011. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 312–317. IEEE.

Milica Gasic, Dongho Kim, Pirros Tsiakoulis, Catherine Breslin, Matthew Henderson, Martin Szummer, Blaise Thomson, and Steve J. Young. 2014. Incremental on-line adaptation of pomdp-based dialogue managers to extended domains. In *INTERSPEECH*.

Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, et al. 2020. Overview of the ninth dialog system technology challenge: Dstc9. *arXiv preprint arXiv:2011.06486*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *arXiv preprint arXiv:2005.00796*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Jinchao Li, Baolin Peng, Sungjin Lee, Jianfeng Gao, Ryuichi Takanobu, Qi Zhu, Minlie Huang, Hannes Schulz, Adam Atkinson, and Mahmoud Adada. 2020a. Results of the multi-domain task-completion dialog challenge. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, Eighth Dialog System Technology Challenge Workshop*, volume 7.

Shiyang Li, Semih Yavuz, Kazuma Hashimoto, Jia Li, Tong Niu, Nazneen Rajani, Xifeng Yan, Yingbo Zhou, and Caiming Xiong. 2020b. Coco: Controllable counterfactuals for evaluating dialogue state trackers. *arXiv preprint arXiv:2010.12850*.

Zachary Lipton, Xiujun Li, Jianfeng Gao, Lihong Li, Faisal Ahmed, and Li Deng. 2018. Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Bing Liu and Ian Lane. 2017. Iterative policy learning in end-to-end trainable task-oriented neural dialog models. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 482–489. IEEE.

Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2018. Dialogue learning with human teaching and feedback in end-to-end trainable task-oriented dialogue systems. *arXiv preprint arXiv:1804.06512*.

Qingbin Liu, Pengfei Cao, Cao Liu, Jiansong Chen, Xunliang Cai, Fan Yang, Shizhu He, Kang Liu, and Jun Zhao. 2021. Domain-lifelong learning for dialogue state tracking via knowledge preservation networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2301–2311.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrea Madotto, Zhaojiang Lin, Zhenpeng Zhou, Seungwhan Moon, Paul Crook, Bing Liu, Zhou Yu, Eunjoon Cho, and Zhiguang Wang. 2020. Continual learning in task-oriented dialogue systems. *arXiv preprint arXiv:2012.15504*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020a. Soloist: Few-shot task-oriented dialog with a single pretrained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2020b. Raddle: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. *arXiv preprint arXiv:2012.14666*.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Kam-Fai Wong, and Shang-Yu Su. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*.

Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. *arXiv preprint arXiv:1704.03084*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Janarthanan Rajendran, Jatin Ganhotra, and Lazaros C Polymenakos. 2019. Learning end-to-end goal-oriented dialog with maximal user task success and minimal human agent use. *Transactions of the Association for Computational Linguistics*, 7:375–386.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 41–51.

Swadheen Shukla, Lars Liden, Shahin Shayandeh, Eslam Kamal, Jinchao Li, Matt Mazzola, Thomas Park, Baolin Peng, and Jianfeng Gao. 2020. Conversation learner–a machine teaching tool for building dialog managers for task-oriented dialog systems. *arXiv preprint arXiv:2004.04305*.

Patrice Y Simard, Saleema Amershi, David M Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, et al. 2017. Machine teaching: A new paradigm for building machine learning systems. *arXiv preprint arXiv:1707.06742*.

Pei-Hao Su, Milica Gasic, Nikola Mrksic, Lina Rojas-Barahona, Stefan Ultes, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. On-line active reward learning for policy optimisation in spoken dialogue systems. *arXiv preprint arXiv:1605.07669*.

Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne. 2021. Transferable dialogue systems and user simulators. *arXiv preprint arXiv:2107.11904*.

Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning*, 8(3-4):279–292.

Jason D Williams and Lars Liden. 2017. Demonstration of interactive teaching for end-to-end dialog control with hybrid code networks. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 82–85.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam

Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented dialog systems that consider multiple appropriate responses under the same context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9604–9611.

A RL Refining Algorithm

Algorithm 1 Self-learning-based RL refining framework.

Input:

Training examples $\mathcal{D}$ in the form of dialog turns;

Trained agent with dialog model $p_{\boldsymbol{\theta_D}}(\boldsymbol{x})$ and reward model $p_{\boldsymbol{\theta_R}}(\boldsymbol{x})$.

Output:

Refined agent with updated dialog model $p_{\boldsymbol{\theta}^*_{\boldsymbol{D}}}$.

1: **while** not converged **do**
2: Randomly sample a dialog turn, i.e. token sequences of dialog history $\boldsymbol{s}$;
3: Run dialog model $p_{\boldsymbol{\theta_D}}$ on dialog history $\boldsymbol{x} = (\boldsymbol{s})$ to generate belief state $\hat{\boldsymbol{b}}$;
4: Retrieve DB state $\hat{\boldsymbol{c}}$ from a database using generated belief state $\hat{\boldsymbol{b}}$;
5: Sample corresponding response $\boldsymbol{r}$ based on dialog history $\boldsymbol{s}$, belief state $\hat{\boldsymbol{b}}$ and DB state $\hat{\boldsymbol{c}}$;
6: Use the reward model to predict the quality of the belief state and response with reward score, $R(\boldsymbol{s}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{c}}, \boldsymbol{r})$;
7: Calculate the loss according to Equation 3;
8: Update the parameters of the dialog model, $\boldsymbol{\theta_D} \leftarrow \boldsymbol{\theta_D} + \alpha \nabla_{\boldsymbol{\theta_D}} \mathcal{L}_{\boldsymbol{\theta_D}}$.
9: **end while**

B Implementation Details

User : I would like to find an expensive restaurant that severs Chinese food. System : Sure, which area do you prefer ? User : How about in the north part of town. [BOS] Belief State : Restaurant { pricerange = expensive, food = Chinese, area = north } [EOB] DB : Restaurant 2 match [EOD] System: The [restaurant_name] is a great [value_food] restaurant. Would you like to book a table there ? [EOS]

Figure 3: Illustration of the training example, *i.e.,* the processed dialog turn in the training data.

To construct training examples as shown in Figure 3, we tokenize the dialog turn sequence using byte pair encodings (Sennrich et al., 2015) and delexicalize responses by replacing slot values with corresponding special slot tokens (Lei et al., 2018). We conduct experiments with several Transformer-based models and GPT-2 (Radford et al., 2019) (enhanced with auxiliary generation tasks) shows

better performance than others. Therefore, we implement proposed reward model based on Huggingface Pytorch Transformer (Wolf et al., 2020) using GPT-2-117M. We pre-train reward model for 10 epochs using Schema dataset (Rastogi et al., 2019), which contains 22,825 dialogs in 17 domains. The reward model is pre-trained on two 24G Nvidia P40 with a mini-batch of 8 and learning rate of 5e-5, using Adam optimizer (Kingma and Ba, 2014), where the training examples are truncated or padded to the max length of 500.

We fine-tune the pre-trained reward model and dialog model (*i.e.,* pre-trained SOLOIST) for 20 epochs with limited number of labeled task-specific dialogs for new tasks. During refinement, top-p is selected as 0.5 for all models. We perform gradient clipping with the max norm as 1 for learning model parameters, with the batch size as 1 and learning rate as 5e-6. The dialog model is refined on a single 24G Nvidia P40 until converging on the validation set. During testing, Nucleus filtering is also used for decoding with top-p as 0.5.

C Experimental Details

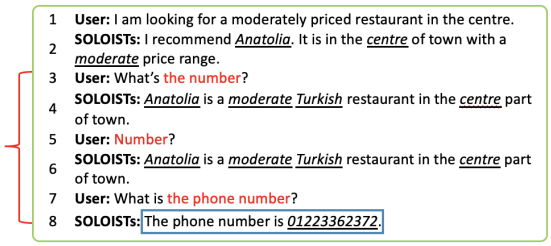
To demonstrate the effectiveness of SL-AGENT, we use SOLOIST as the dialog model to compare the performance of different methods, since existing state-of-the-art task-oriented dialog models share similar input-output pairs and training objectives as SOLOIST. (We report the results in mean of 5 runs with 5 different seeds.) (i) To obtain SOLOIST$_S$, we implement the synthetic dialog construction method by exhausting DB values. For each dialog turn of the 5 labeled dialogs, we randomly sample five DB values from the database to replace the original slot values. (ii) To obtain SOLOIST+PARG, we use the Transformer-based machine translation checkpoints (English-German, German-English) (Edunov et al., 2018) to generate 10 paraphrased user utterances for each dialog turn of the 5 labeled dialogs (based on the empirical analysis of translation quality). Then we use these annotated data (with paraphrased user utterances) to train SOLOIST$_S$ for obtaining SOLOIST+PARG. (iii) To obtain SOLOIST-OA, we use the method described in Section 8 to construct successful dialogs and failed dialogs. For successful dialogs, we use the original 5 labeled dialogs, and the dialogs containing paraphrased user utterances. To construct the failed dialogs, we randomly select 2-3 dialog turns in each dialog and corrupt responses according to the negative example construction method in Section 3.3. Then we use these annotated dialogs to train the session-level reward model of (Su et al., 2016). When testing the performance in the simulated setting, we refine the SOLOIST$_S$ with fully correct dialogs and dialogs containing corrupted responses. To achieve better performance, we largely query for session-level human feedback score in both simulated setting and real-scenario setting.

Model	Restaurant			
	Inform	Success	BLEU	Combined
SOLOIST$_S$	62.50	41.50	7.33	59.33
SL-SOLOIST	75.00	44.50	10.60	70.35
SL-SOLOIST$_{+20}$	**75.00**	**52.00**	**11.89**	**75.39**

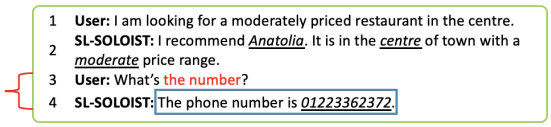
Table 7: End-to-end evaluation results of Policy Improvement in the Restaurant domain. SL-SOLOIST$_{+20}$ refer to continually refining with 20 real (unlabeled) human-bot dialogs based on SL-SOLOIST (reported in Table 2).

D Simulated Human-Bot Corpora Construction

The unlabeled simulated human-bot corpora is constructed as follows: (i) we remove belief state annotations; (ii) we add negative examples by corrupting responses according to the negative example construction method in Section 3.3. We will release the simulated human-bot corpora for reproducible research. Note that directly replacing the belief states and responses with the generated ones is trivial. However, such approach cannot imitate realistic human-bot interactions. As the user utterances are strictly fixed, "users cannot react to the agent responses accordingly and appropriately". Therefore, we also conduct experiments through conversing with real users in the real-scenario setting and demonstrate the results in Table 3. Furthermore, building a user simulator is inapplicable in our changing environment setting. (i) It is difficult to build reliable user simulators. Building agenda-based user simulators requires sophisticated human expertise for designing rules. (ii) Building model-based user simulators requires sufficient labeled data. Furthermore, model-based user simulators merely imitate expert behaviors in the training corpus, cannot provide user behaviors that are unseen from task bots.



(a) An interactive example between user and SOLOISTs



(b) An interactive example between user and SL-SOLOIST

Figure 4: Two interactive examples. (a) An interactive example between user and SOLOIST$_S$. (b) An interactive example between user and SL-SOLOIST.

E Policy Improvement

Policy Improvement Setup. To demonstrate the effectiveness of SL-AGENT for continually learning from collected human-bot dialog logs, we deploy SL-SOLOIST online and recruit human users to converse with it to achieve the assigned user goal. We collect 20 real human-bot dialog logs to refine SL-SOLOIST, resulting in the agent SL-SOLOIST$_{+20}$. (When refining the SL-SOLOIST, we do not use the knowledge about the user's goal. The response quality is judged by the reward model in SL-SOLOIST.)

Policy Improvement Results. The evaluation results on `Restaurant` are shown in Table 7. We observe that SL-SOLOIST$_{+20}$ refined with 20 real (unlabeled) human-bot dialogs outperforms SL-SOLOIST by approximately 5 points in terms of `Combined` score. We conclude that SL-SOLOIST enables continual self-learning after deployment by automatically learning from interactions.

F Interactive Example

Figure 4 depicts two interactive examples where the same user interacts with SOLOIST$_S$ and SL-SOLOIST to complete the same task. We observe that, in the first four dialog turns, the two agents has the same performance and both correctly recommend a satisfied restaurant. However, as shown in Figure 4 (a), when user queries about the phone number ("what's the number?") in the fifth turn, SOLOIST$_S$ fails to understand user's intent and generates incoherent response, still trying to provide recommendation. The user has to continually query about phone number in the following consecutive turns. As demonstrated in Figure 4 (b), SL-SOLOIST correctly provides the phone number, when user first queries about it. Comparing the two examples, we show that SL-AGENT enables adapting to unseen user behaviors in an automatic way.

G An Example of Task Definition Extensions

Figure 5 depicts an example of task definition extensions.

t	Speaker	Utterance (u_t)
1	User	Hi, I'm looking for a place with Tuscan food in any price range.
2	System	I'm sorry, there is not a *Tuscan* restaurant listed. Would you care to try something else?
3	User	How about any Korean restaurants?
4	System	*Little Seoul* is a *Korean* restaurant.
5	User	Phone number please.
6	System	The phone number is *01223308681*. Is there anything else I can help you with?
7	User	Does the restaurant offer delivery service? How much does the delivery charge?
8	System	Yes, and the delivery fee is *4 pounds*. Would you like more information about the service?
9	User	No. Thank you, goodbye.
10	System	Thank you. Goodbye.

Figure 5: An example of task definition extensions. Task bots need learn to provide information about the extended delivery service in additional dialog turns (in Red) as user requirements evolve.

H An Example of `Restaurant-Ext` DB Entry

An example of `Restaurant-Ext` DB entry is shown in Figure 6.

I Item Examples of the Input Dialog Turn Sequence

J Negative Example Construction

```
{
  "address": "Finders Corner Newmarket Road",
  "area": "east",
  "food": "international",
  "id": "30650",
  "introduction": "",
  "location": [
    52.21768,
    0.224907
  ],
  "name": "the missing sock",
  "phone": "01223812660",
  "postcode": "cb259aq",
  "pricerange": "cheap",
  "type": "restaurant",
  "delivery fee": "5 pounds",
  "dish": "Greek Chicken Pasta",
  "start_time": "09:50",
  "end_time": "22:30"
},
```


Figure 6: An example of `Restaurant-Ext` DB entry. Newly added DB information about the extended function is in the red square.

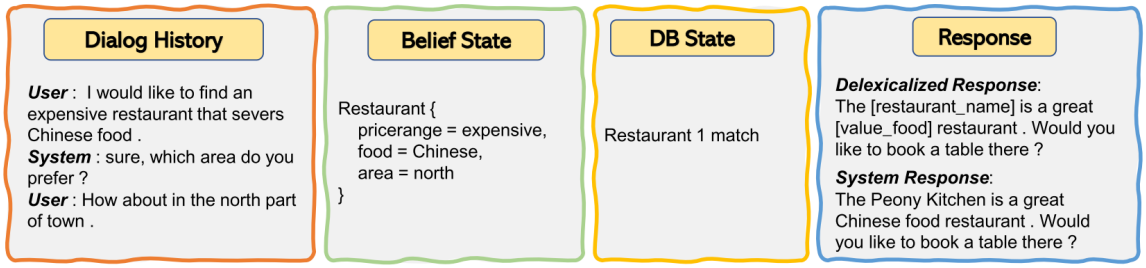


Figure 7: Item examples of the input dialog turn sequence for SOLOIST, cited from (Peng et al., 2020a).

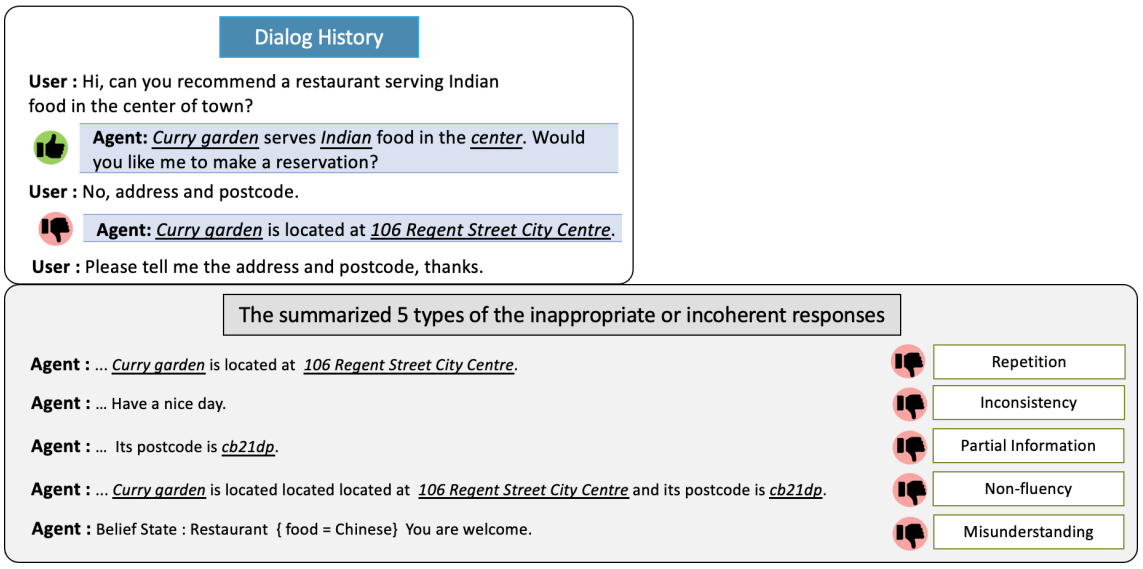


Figure 8: The summarized 5 types of dialog turns that have inappropriate or incoherent responses. (a) Dialog history (top). (b) 5 types of the inappropriate or incoherent responses (bottom).

Combining Structured and Unstructured Knowledge in an Interactive Search Dialogue System

Svetlana Stoyanchev[1] **Suraj Pandey**[2] **Simon Keizer**[1]
Norbert Braunschweiler[1] **Rama Doddipatla**[1]
[1] Speech Technology Group - Toshiba Europe Ltd. Cambridge, UK
[2] The Open University, Milton Keynes, UK
[1] {firstname.lastname}@crl.toshiba.co.uk
[2] surajjung@gmail.com

Abstract

Users of interactive search dialogue systems specify their preferences with natural language utterances. However, a schema-driven system is limited to handling the preferences that correspond to the predefined database content. In this work, we present a methodology for extending a schema-driven interactive search dialogue system with the ability to handle unconstrained user preferences. Using unsupervised semantic similarity metrics and text snippets associated with the search items, the system identifies suitable items for the user's unconstrained natural language query. In a crowd-sourced evaluation, the users were asked to chat with our extended restaurant search system. Based on objective metrics and subjective user ratings, we demonstrate the feasibility of using this unsupervised low latency approach to extend a schema-driven search dialogue system to handle unconstrained user preferences.

1 Introduction

We extend a schema-driven dialogue system with the ability to handle unconstrained user queries and to allow users to specify preferences flexibly as they would when using a search engine.

Interactive search dialogue systems, such as search for restaurants, hotels, trains, books, shows, venues, are task-oriented systems that provide a natural language interface for interactive search and information extraction. In these systems, a user typically starts by typing (or speaking) a search query. Next, the system's policy chooses an optimal action, which may be either asking the user to provide additional information or presenting one or more search result options. Once the system presents an option, the user may provide another query, narrowing down or changing the search criteria, or ask a question about the presented option(s).

In a schema-guided approach to designing a dialogue interface (Rastogi et al., 2020a), a set of 'informable' and 'requestable' slots derived from the fields of the underlying database table (or schema), define the natural language interface capability. A user can specify the values of 'informable' fields as search criteria and ask questions to retrieve information stored in the 'requestable' fields. The schema-guided method simplifies authoring dialogue systems for new domains. With this approach, a dialogue interface for a new database may be bootstrapped from the schema/ontology and database content of the domain.[1]

One of the drawbacks of the schema-guided approach is that the criteria by which the user can search for an item and the types of questions that the user may ask are limited by the database schema. For example, a restaurant search query *'Find a romantic place that serves great wines'* cannot be handled by a schema-driven system unless the schema includes the relevant properties of the restaurant atmosphere and wine quality. It is possible to design a system to notify the user of its limitations using help messages (Komatani et al., 2005), but the constraint on the interaction remains, as the user is unable to retrieve items using criteria other than those defined by the 'informable' slots. Given that in many domains, additional unstructured information beyond the database fields may be available, it is natural to extend schema-guided systems to use this unstructured information. Kim et al. (2020) describe a system that extends the schema-guided functionality with the ability to ask follow-up questions. In this work, we propose to extend the information search dialogue interface functionality to retrieve items for unconstrained user queries.

To handle user queries that cannot be grounded in terms of a domain schema and ontology, we propose to use semantic similarity metrics to retrieve search results from unstructured data. We evaluate the proposed approach with crowd-sourced

[1]The schema/ontology refers to the definition of the database tables.

Proceedings of the SIGdial 2022 Conference, pages 531–540
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

users interacting with the restaurant search system through a chat interface. In previous work, restaurant search systems were evaluated by giving users predefined 'goals' which primed users and lead to rigid interactions. In our evaluation, the users are given a general instruction to find an ideal restaurant and are free to specify any search query. The results show the users' preference for the proposed flexible system that allows the use of unconstrained search queries. We release the dialogues with the automatically annotated intents and the subjective user judgements collected during the evaluation to the research community.[2]

2 Related Work

Interactive search can be modelled as task-oriented dialogue using structured knowledge, symbolic dialogue state representation, and a statistical policy that addresses both task and conversational phenomena, such as clarifications and social dialogue acts (Budzianowski et al., 2018; Yan et al., 2017). However, users of dialogue systems that are based only on structured knowledge are limited in expressing their preferences by the underlying database schema. In response to an out-of-schema user request, a task-oriented dialogue system may produce an informative help message guiding the user to adapt to its limitations (Komatani et al., 2005; Tomko and Rosenfeld, 2004). Alternatively, system capabilities may be extended beyond a domain API. For example, Kim et al. (2020) proposes a method for handling user's follow-up questions in task-oriented dialogue systems. To support pragmatic interpretation, Louis et al. (2020) explores users' indirect responses to questions. To extend a task-oriented system to handle natural preferences, a corpus of natural requests for movie preferences was collected using preference elicitation (Radlinski et al., 2019).

End-to-end approaches to dialogue, where the system generates a response without explicitly modelling intent or storing a dialogue state, have been successfully applied to open-domain chitchat (Serban et al., 2016). The use of unstructured knowledge was shown to improve open domain chitchat systems (Dinan et al., 2018; Ma et al., 2022; Zhou et al., 2018). In recent work, interactive search has been modelled as end-to-end generation task using text and images as the knowledge source (Varshney and Anushkha Singh, 2021).

Search tasks in natural human-human dialogues can be complex and are often resolved interactively (Trippas et al., 2017, 2018), motivating the need for methods capable of handling natural conversational phenomena as well as extracting information and generating knowledge-grounded responses. In this work, we evaluate a task-oriented dialogue system with a semantic-level policy extended with the use of unstructured knowledge.

Task-oriented dialogue systems require accurate models to extract information from unstructured text. Pretrained transformer models, such as BERT (Devlin et al., 2019), have shown to be effective in extracting information from text, leading to significant improvements on many NLP tasks, including open-domain question answering, FAQ retrieval, and dialogue generation (Wang et al., 2019; Sakata et al., 2019; Kim et al., 2020). Following previous work, we use BERT in an unsupervised setting to extract relevant information from text (Izacard and Grave, 2021; Zhan et al., 2020).

3 Method

3.1 System Overview

We implement a schema-driven restaurant search dialogue system that uses a database with 422 restaurants in Cambridge, UK.[3] Following the database schema used in previous work (Henderson et al., 2014), each restaurant is described in terms of the following attributes: *name*, *cuisine*, *price range*, *area*, *phone number*, and *address*. As in previous systems, the price range is mapped to *cheap, moderate*, or *expensive* and location to *north, south, east, west,* or *city centre*. In contrast to the schema in the DSTC2 domain (Henderson et al., 2014), *cuisine* in our database is mapped to a list of values rather than a single value for each entity. In addition to the specific restaurant attributes, each restaurant is associated with a set of text snippets including *meals* (breakfast, lunch, dinner), *special diets* (e.g., vegan, gluten free) and *reviews*. Only positive reviews (rating 4 or 5 stars) are used, as we expect user queries to mention desirable properties of the restaurant. 99% of the snippets are reviews and the average number of text snippets per restaurant is 147, varying between 2 and 1.6K.

[2] https://github.com/sstoyanchev/Unstructured_restaurant_search_dialogues_Sigdial2022

[3] The database is compiled by crawling the Web in January 2021.

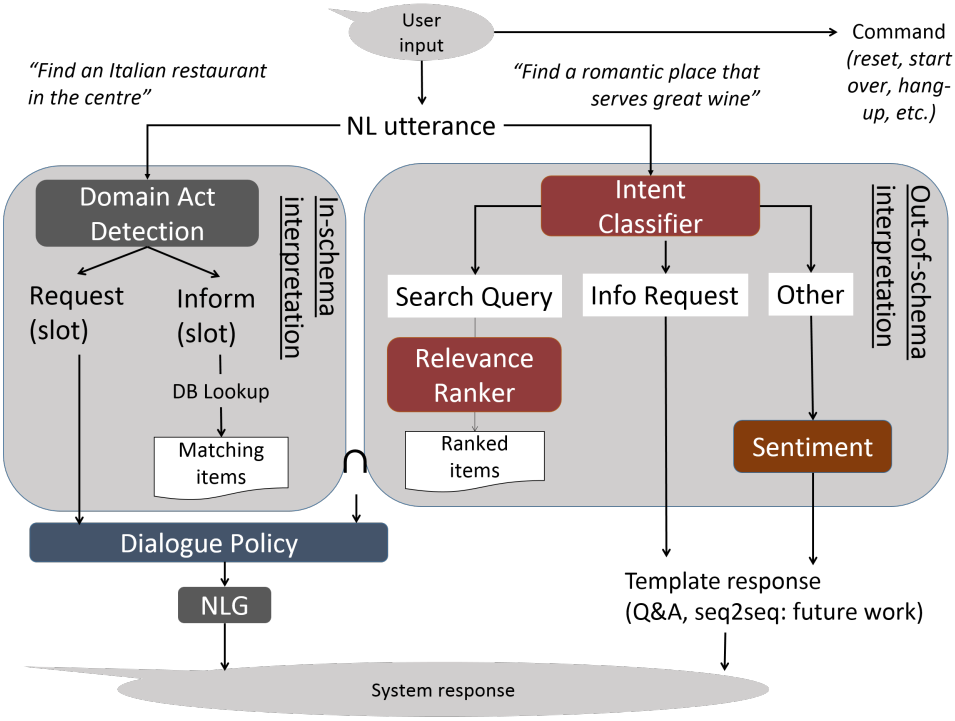


Figure 1: System diagram showing processing of in-schema and out-of-schema user input.

Figure 1 outlines the system architecture. The left-hand side shows the components handling in-schema user acts, such as requesting and providing information for one of the restaurant-specific attributes. The Domain Act Detection module interprets in-schema user acts (Stoyanchev et al., 2021) and a database lookup results in a new list of matching restaurants. A statistical dialogue policy component trained in simulation in the purely schema-driven DSTC2 domain generates the system response for in-schema user utterances (Keizer et al., 2021). The right-hand side of Figure 1 shows the components for handling out-of-schema utterances that do not mention any of the schema-specific attributes, e.g. *'Find a romantic place that serves great wine'*. In (Kim et al., 2020), the authors build a binary model that determines whether to access unstructured data for follow-up question answering. In the proposed system, a prediction of the intent classifier triggers access to the unstructured dataset of restaurant reviews.

3.2 Intent Classification

A task-oriented dialogue system is designed to handle generic dialogue acts and domain-specific intents. Dialogue act taxonomies (Core and Allen, 1997; Bunt et al., 2010) distinguish general purpose acts based on the surface form of the utterance, such as inform or question. However, in interactive search dialogue, the user can formulate a query either with a question, *'Can you find me...?'* or a statement *'I would like...'*. Hence, a distinction between the surface forms is not sufficient and instead, we define the intents specific to the search task *Search Query* (SQ), *Info Request* (IRq), and *Other*. Utterances labeled as SQ include initial and follow-up queries triggering information extraction and resulting in a retrieved set of items. Utterances labeled as IRq include information seeking requests related to one of the restaurants in context, e.g. *'Are dogs allowed?'*, which may trigger a question answering module.[4] The *Other* class includes utterances that are neither SQ nor IRq, for example, an exclamation *'Great!'*. While these utterances do not trigger data access, it is important to detect them and respond appropriately to, in order to maintain fluent conversation.

We tune the pre-trained uncased BERT transformer model (Devlin et al., 2019) on this 3-way classification task. Table 1 shows the statistics for the training dataset. We obtain the initial training dataset from two publicly available task-oriented dialogue corpora: Schema Guided Dialogue (SGD) and Frequently Asked Questions (FAQ) (Rastogi et al., 2020b; Kim et al., 2021). SGD contains semi-

[4]Question answering from unstructured data for this system remains future work.

Intent	Initial Dataset		Collected with the system	Overall	Average #words±stdev
	SGD (1,698)	FAQ (2,198)	(554)	(4,450)	
Search Query (SQ)	72%	-	47%	33%	11.124±5.794
Info Request (IRq)	-	100%	41%	54%	7.186±2.21
Other	28%	-	12%	12%	4.458±2.026

Table 1: Statistics of the dataset used to train the intent classifier showing the numbers of utterances extracted from the schema-guided dialogue corpus (SGD), DSTC9 Beyond Domain APIs track (FAQ), and collected with the system.

automatically generated task-oriented dialogues in 26 domains, including restaurant search, annotated with dialogue acts. We confirm that the initial utterances in the restaurant search domain are search queries and use them as the training examples for the SQ class[5]. Since the utterances in the SGD dataset are authored by people, they include a wide variety of queries outside of our system's domain schema. We use the utterances from the restaurant search domain in SGD annotated as 'AFFIRM', 'NEGATE', or 'SELECT' for the *Other* class. The FAQ dataset includes 2.2k manually authored questions in the restaurant search domain for the Beyond Domain APIs track of The Ninth Dialog System Technology Challenge (Kim et al., 2021; Budzianowski et al., 2018). We use the questions as examples for the IRq class.

Next, we train an intent classifier using the data from the SGD and FAQ datasets and evaluate the system internally. We collect an additional 554 utterances where the authors and colleagues interacted with the system using a web-based chat interface. As the initial set of SQ did not contain any follow-up queries, the intent classifier tended to fail on such utterances. We manually annotate the collected utterances with the dialogue act label and include them in the training set.

3.3 Relevance Ranking

The Relevance Ranker accesses unstructured data producing a ranked list of candidate items (restaurants in Cambridge) for a user's search query. The unstructured data includes reviews and restaurant details extracted from the Web, stored as text snippets associated with each item. Restaurants with snippets that have higher semantic similarity to the user query are more likely to be relevant for the user (see Table 2).

[5]30% of the initial SGD utterances were manually examined to confirm that they correspond to search queries.

Query: I am looking for a place that serves vegan food and also allows dogs inside.

Relevant snippets	
Special diets	Vegan friendly
Review	It was such a happy surprise that they allowed dogs inside their premises.

Table 2: Query and relevant snippet examples

First, we score each snippet with the semantic similarity according to the user's query. In previous work, we have shown that a supervised model based on BERT encoding and trained on in-domain data achieves F1=.86 on the binary task of identifying relevant query-snippet pairs (Pandey et al., 2021). However, using such a model is computationally expensive as it requires 60k snippets to be classified during run-time, making it intractable for a real-time system. Instead, we use a less accurate low latency approach which achieves F1=.66 on this task.

To measure semantic similarity between the query and the snippet, the user query (Q_i) and each snippet (S_j) are mapped into a fixed-sized vector using an encoding function E. The cosine similarity score between the user request and each snippet is used to measure the semantic similarity:

$$Score(Q_i, S_j) = cos(E(Q_i), E(S_j)) \quad (1)$$

As the encoder, we use pretrained Sentence-BERT (SBERT) optimized on the semantic similarity task and further tuned on the SNLI corpus of semantic entailment which was previously shown to improve sentence classification performance (Reimers and Gurevych, 2019; Bowman et al., 2015). The intuition is that the tuned model's capability will extend to capture not only semantic similarity between the encodings but also the entailment, which may be a relation between search

query and a relevant snippet. SNLI is a collection of 570,000 sets of premises and hypotheses sentences annotated with the labels *contradiction*, *entailment*, and *neutral* as in the example in Table 3. We use the pairs of *Premise&Entailment* as positive examples and *Premise&Contradiction/Neutral* as the negative examples to further tune the SBERT model.

Premise:	A boy is jumping on skateboard in the middle of a red bridge.
Entailment:	The boy does a skateboarding trick.
Contradiction:	The boy skates down the sidewalk.
Neutral:	The boy is wearing safety equipment.

Table 3: Example from SNLI dataset

Next, the items (restaurants) are ranked based on the average score of the top M snippets for each item. The top N items are returned.[6]

A user's search query may specify a schema-specific attribute as well as additional information. For example, a query *'Italian restaurant with great desserts'* specifies a food type (Italian) as well as an out-of-schema preference (great desserts). Such queries are processed both by in-schema and out-of-schema modules. The in-schema processing narrows down the set of results to *Italian* and the out-of-schema processing ranks the restaurants based on the snippets' similarity with the query. The result is the ranked list of Italian restaurants where the restaurants with the snippets mentioning the high quality of desserts (if there are any) are at the top.

3.4 System Response

If at least one domain-specific user action (inform-slot or request-slot) is detected in the user's utterance, the utterance is considered in-schema. The system's response to an in-schema utterance may be an *Offer* (e.g., *'Zizzi is an Italian restaurant in the centre'*), a clarification (e.g., *'Did you mean in the centre?'*), or a request for additional information (e.g., *'What price range do you prefer?'*). The response act is selected using a statistical policy which maximizes the expected reward, and the surface form of the response is generated using templates. The out-of-schema user utterances do not mention any of the slots and can not be directly handled with the policy trained on a schema-driven dataset. The system uses the prediction of the intent classifier to determine the method of response selection.

When an out-of-schema utterance is classified as search query, the system updates the state with an inform action and the dialogue policy selects the response act. If the *Offer* act is selected, the system presents the top-ranked restaurant and its top-matching review is appended to the template-generated description. See Figure 2(b) for examples of system responses.

While searching for an item, a user may ask questions about the previously discussed items (restaurants). The intent classifier labels such questions as *Info Request*, differentiating them from the search queries produced in a question form. If the user requests information about one of the schema attributes (phone number, address, price range, etc.), the statistical policy determines the system's response. However, a user may also ask for information outside of the database schema, such as *'Are dogs allowed inside?'*. Currently, the system informs the user that the question cannot be answered signalling understanding of the user's intent. In future work, we plan to handle such questions with a question answering model, e.g. following the approach proposed in (Kim et al., 2020).

According to the initial data collection with the dialogue system, 12% of user utterances are neither search queries nor information requests (see Table 1). These utterances are typically exclamations like *'Sounds great!'* or *'Too bad!'* and are labeled as *Other* by the intent classifier. The system responds to these utterances by generating a sentiment-appropriate template-based response. A user utterance labeled as *Other* is processed with an off-the-shelf RoBERTa model trained on $\sim$ 58M tweets and fine-tuned for sentiment analysis with the TweetEval benchmark (Barbieri et al., 2020). The model outputs either a positive, negative, or neutral sentiment class of the input text. Based on the predicted sentiment, the system selects one of the appropriate template responses, e.g. *Brilliant! How else can I help you?* for a positive sentiment or *'OK, calm down now.'* for a negative sentiment, maintaining the dialogue flow and adding a bit of template-based humour.

[6] We use empirically chosen M=5 and N=5 in this work.

4 Evaluation

Our goal is to evaluate the use of unstructured data in an interactive task-oriented dialogue system. The proposed approach involves two statistical models: 1) intent classifier and 2) relevance ranker, which accesses unstructured data, depending on the prediction of the intent classifier. We first show the performance of the intent classifier on the collected dataset and then describe the human evaluation of the overall system.

4.1 Intent Classification

Model	Train/Test data	Accuracy
SVM	All	88.2%
BERT	All	**99.8%**
BERT	Initial/Collected	70.9%

Table 4: Intent classification accuracy.

We evaluate the intent classification performance using the dataset described in Table 1. We compare the performance of the BERT model with bag-of-words SVM baseline using stratified 10-fold cross validation on the full dataset of 4.5K utterances. The BERT model and the SVM model achieve 99.8% and 88.2% overall accuracy (see Table 4).[7] This shows that the pre-trained transformer model is able to effectively capture the distinction between the three intent classes in the restaurant search domain. The intent classifier trained on the initial data subset (the utterances from SGD and FAQ) achieved accuracy of only 70.9% on the utterances collected with the dialogue system, which is not sufficient for the interaction with the real users.

4.2 Human Evaluation

4.2.1 Experimental Setup

System	Intent classifier	Relevance Ranker	Snippet in Offer
SCHEMA	-	-	-
RAND-RANK	+	-	+
EXP-RANK	+	+	+

Table 5: Experimental Conditions.

When evaluating schema-driven dialogue systems with recruited experiment participants, the users are typically given concrete goals, e.g. *'You are looking for a cheap Italian restaurant and don't care about the price range'*. Since the proposed system handles unconstrained queries, we instruct the users to imagine a situation like going out with a family, a romantic date, a business lunch, or use any other preferences. Given more general instructions, the user can come up with authentic in-schema or out-of-schema search preferences. The users are instructed to retrieve at least three restaurant options and ask for the address of their preferred restaurant. In all experimental conditions, the system initiated the dialogue with a general prompt: *'Please describe the restaurant you would like'*. The users are instructed to use the command 'start over' if they feel that the system is stuck.

We evaluate three system variants:

- SCHEMA: the baseline condition where the system handles only in-schema user queries. In the initial prompt, the system informs the user of its limitations: *'You can search by food type, price range, or area (centre, north, etc)'*. For the out-of-schema user queries the system responds with the indication of non-understanding, e.g. *'Sorry, I don't understand. Please rephrase'*.
- RAND-RANK: the control condition where the system responds to the user's out-of-schema requests with a randomly selected search result.
- EXP-RANK: the experimental condition where the system that handles unconstrained requests using the method described in Section 3.

Table 5 summarizes the differences between the three systems. The SCHEMA baseline does not use the intent classifier and the relevance ranker. Its *Offer* act does not include the text snippet associated with the restaurant. RAND-RANK is used as the control condition to isolate the effect of the relevance ranker. The RAND-RANK system uses the intent classifier and includes a snippet in the system's *Offer* dialogue act. However, it assigns random relevance scores to the text snippets resulting in a random selection of the proposed restaurant. If RAND-RANK and EXP-RANK receive similar user ratings, the preference over SCHEMA may be due to the intent classifier and snippets in the offer output. We hypothesize that the users

[7] This result was achieved after 5 epochs.

SCHEMA
U: i would like a great place to watch a sports game on tv. anything that serves fresh burgers and good beer
S: Sorry, I don't understand. Please rephrase your utterance.
U: start over
S: Let's start over! What type of restaurant are you looking for?
U: american food
S: There are no restaurants serving the americas food.
U: burger restaurants
S: There are no restaurants that serve barbeque food.
U: start over
S: Let's start over! What type of restaurant are you looking for?
U: sushi
S: I did not quite catch that. Could you please rephrase?
U: start over
S: Let's start over! What type of restaurant are you looking for?
U: thai
S: Banana Leaf is a restaurant serving thai food. You can type 'next option' or another query.
U: next option
S: How about Nooms' Kitchen Thai Cuisine?
U: what is the address
S: The address of Nooms' Kitchen Thai Cuisine is 80 Victoria Street.

(a)

EXP-RANK
U: i am looking for a thai place that serves spicy food
S: Thaikhun Cambridge is a restaurant that serves thai food. This Thai restaurant has taken authentic very serious as you will find a tuk tuk parked outside the restaurant. Fortunately they also take th cooking serious as the food the usual thai dishes that are well cooked and can be enhanced by the chef if you prefer the spicy and... You can type 'next option' or another query.
U: are there any other thai places that serve authentic dishes
S: Sala Thong Restaurant is a restaurant serving thai food. Sala tong is the real thing, serve real thai style crusine. If you miss thai this is the place to go. You can type 'next option' or another query.
U: i am really craving some fresh sushi. do you have any places near me
S: Thaikhun Cambridge is a restaurant serving thai food. Lunchtime stop for refreshments, had one special & one choice from main menu. Pad Thai & massaman curry delicious. Thai fish cakes small but tasty. Will visit again until you open in Norwich. You can type 'next option' or another query.
U: any good japanese food that serves spicy and fresh wontons
S: YO! Cambridge is a restaurant that serves japanese food. hot street food available authentic japanese food innovative fusion dishes watering range traditional classics natural ingredients aromatic flavours sushi rich packed mouth known fresh flavoursome also. You can type 'next option' or another query.
U: what is the address of cambridge
S: The address of YO! Cambridge is 1-2 Petty Cury, Lion Yard.

(b)

Figure 2: Example dialogue for the SCHEMA and EXP-RANK systems.

prefer EXP-RANK over both RAND-RANK and SCHEMA.

	SCHE-MA	**RAND-RANK**	**EXP-RANK**	**Over-all**
#Dlg	81	81	81	243
#Utts	636	610	557	1803
SQ	47.3%	32.5%	34.8%	38.4%
IRq	25.3%	36.1%	32.9%	31.3%
% Out-of-schema				
SQ	44.5%	47.5%	29.4%	41.1%
IRq	11.8%	28.2%	11.5%	14.2%

Table 6: Statistics for the user utterances based on the automatic predictions. % of Search Query (SQ) and Info Request (IRq); % of in-schema SQ and IRq utterances.

4.2.2 Data

We collected 243 dialogues (81 for each system variant), with a total of 1,803 user utterances summarized in Table 6. 38.4% are classified as search queries and 31.3% are classified as information requests. The SCHEMA system is unable to process out-of-schema queries, leading to the longer dialogues where users have to change their initial query.

41.1% of the search queries are out-of-schema (no slots are detected) indicating that the users' preferences constructed without specific instructions are likely to mention information other than price range, area, and food type. Surprisingly, we find that in EXP-RANK system, only 29.4% of search queries are out-of-schema. We notice that the users tend to provide queries with both in-schema and out-of-schema info, e.g. *'I'd like to find a mexican restaurant that has excellent customer service'* which are considered in-schema yet they can benefit from the use of relevance ranking.

Despite the instructions given to the users to find out the address of their preferred restaurant, 14.2% of information requests ask about the information outside of the schema. This shows the need for the more flexible question answering capability. Example dialogues with the SCHEMA and the

Question	SCHEMA	RAND-RANK	EXP-RANK
Self-Reported Subjective Ratings			
I was able to find a satisfactory restaurant option.	4.086±1.769	4.358±1.559	**4.888±1.423**†
The restaurant descriptions matched my preferences.	4.074±1.787	4.308±1.578	**4.876±1.354**†
The system understood me well.	3.592±1.909	3.753±1.684	**4.518±1.696**†
The conversation felt natural.	3.666±1.936	3.679±1.723	**4.209±1.633**
I would recommend the system to my friends.	3.358±2.020	3.543±1.837	**4.320±1.808**†
Objective Metrics			
Average dialogue length (# exchanges)	7.9	7.5	**6.9**
Success rate	80.2%	91.4%	**95.1%**
'Start over' rate	3.8%	**1.5%**	**1.5%**

Table 7: Average scores and standard deviation for the subjective user judgements and objective metrics. † indicates a statistical significance with the SCHEMA condition ($p < .05$). Success rate is the % of the dialogues where a user made a choice of a restaurant in the questionnaire.

EXP-RANK systems are shown in Table 2.

4.2.3 Results

We asked the users to score each dialogue on a scale from 1 (strongly disagree) to 6 (strongly agree) for the five subjective statements shown in Table 7. For all statements, the users prefer the EXP-RANK over SCHEMA and over RAND-RANK. The difference in the scores between SCHEMA and EXP-RANK is statistically significant based on the two-tailed t-test with p<0.05 for all statements (except for *'The conversation felt natural'*). The biggest difference (nearly 1 point) between the scores of EXP-RANK and SCHEMA systems was observed for the questions *'The system understood me well'* and *'I would recommend the system to my friends'*. We did not observe a significant difference in subjective ratings between the RAND-RANK and SCHEMA systems. These results suggest that relevance ranking together with the intent classification and additional information in the system response lead to the higher user rating.

We also report the objective scores: average dialogue length, success rate, and 'start over' rate. The dialogues with the EXP-RANK system are the shortest, while the dialogues with the SCHEMA system are the longest, on average 6.9 and 7.9 exchanges respectively. This result shows that users were able to complete the task quicker using the EXP-RANK system than the baseline systems.

In the questionnaire, the users were asked to record the name of their preferred restaurant or 'None' if no restaurants matched their preference. Success rate is the % of the dialogues where the user indicated a preferred restaurant name in the questionnaire. EXP-RANK system achieves the highest success rate of 95.1% in comparison with 91.4% and 80.2% for the RAND-RANK and SCHEMA conditions.

The users had an option to use *'start over'* command when they felt stuck in the dialogue. We observe a higher proportion of *'start over'*s in the SCHEMA system than in the other two systems which use intent classifier and return a suggestion for out-of-schema response leading to fewer non-understanding system responses. This result indicates a functional improvement of the RAND-RANK over the SCHEMA system, which, however, was not reflected in the users' subjective ratings.

5 Conclusions

In this work, we propose a hybrid design for information navigation dialogue systems combining structured and unstructured knowledge. We present a restaurant search dialogue system where the users specify preferences flexibly as they would to a search engine. The proposed system uses the structured knowledge in a database to extract matching restaurants when a user's natural language search query mentions one of the database fields and unstructured text when it does not. The system is evaluated in the interactive experiments with crowdsourced users. The results show a preference for the proposed approach. In future work, we will extend the system to answer follow-up questions and introduce a response generation model.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium.

Harry Bunt et al. 2010. Towards an ISO standard for dialogue act annotation. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

Mark G. Core and James F. Allen. 1997. Coding dialogs with the damsl annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35, Cambridge, MA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *In NAACL-HLT*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *CoRR*, abs/1811.01241.

Matthew Henderson, Blaise Thomson, and Jason D Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the 15th Annual SIGdial Meeting on Discourse and Dialogue*, pages 263–272.

Gautier Izacard and Édouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880.

Simon Keizer, Norbert Braunschweiler, Svetlana Stoyanchev, and Rama Doddipatla. 2021. Dialogue strategy adaptation to new action sets using multi-dimensional modelling. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2021, Cartagena, Colombia, December 13-17, 2021*, pages 977–983. IEEE.

Seokhwan Kim, Mihail Eric, Karthik Gopalakrishnan, Behnam Hedayatnia, Yang Liu, and Dilek Hakkani-Tur. 2020. Beyond domain APIs: Task-oriented conversational modeling with unstructured knowledge access. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 278–289, 1st virtual meeting. Association for Computational Linguistics.

Seokhwan Kim, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, and Dilek Hakkani-Tur. 2021. Beyond domain apis: Task-oriented conversational modeling with unstructured knowledge access track in dstc9. *arXiv preprint arXiv:2101.09276*.

Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. 2005. User modeling in spoken dialogue systems to generate flexible guidance. *User Model. User Adapt. Interact.*, 15(1-2):169–183.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7411–7425, Online. Association for Computational Linguistics.

Longxuan Ma, Mingda Li, Wei-Nan Zhang, Jiapeng Li, and Ting Liu. 2022. Unstructured text enhanced open-domain dialogue system: A systematic survey. *ACM Trans. Inf. Syst.*, 40(1):9:1–9:44.

Suraj Pandey, Svetlana Stoyanchev, and Rama Doddipatla. 2021. Towards handling unconstrained user preferences in dialogue. In *Proceedings of The 12th International Workshop on Spoken Dialog System Technology (IWSDS)*.

Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. 2019. Coached conversational preference elicitation: A case study in understanding movie preferences. In *Proceedings of the Annual SIGdial Meeting on Discourse and Dialogue*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020a. Schema-guided dialogue state tracking task at DSTC8. In *Proceedings of the AAAI Dialog System Technology Challenges Workshop*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020b. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Wataru Sakata, Tomohide Shibata, Ribeka Tanaka, and Sadao Kurohashi. 2019. Faq retrieval using query-question similarity and bert-based query-answer relevance. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1113–1116.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C. Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*.

Svetlana Stoyanchev, Simon Keizer, and Rama Doddipatla. 2021. Action state update approach to dialogue management. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7398–7402.

Stefanie Tomko and Roni Rosenfeld. 2004. Speech graffiti vs. natural language: Assessing the user experience. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 73–76, Boston, Massachusetts, USA. Association for Computational Linguistics.

Johanne R. Trippas, Damiano Spina, Lawrence Cavedon, Hideo Joho, and Mark Sanderson. 2018. Informing the design of spoken conversational search: Perspective paper. In *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval*, CHIIR '18, pages 32–41, New York, NY, USA. ACM.

Johanne R. Trippas, Damiano Spina, Lawrence Cavedon, and Mark Sanderson. 2017. How do people interact in conversational speech-only search tasks: A preliminary analysis. In *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, CHIIR '17, pages 325–328, New York, NY, USA. ACM.

Deeksha Varshney and Asif Ekbal Anushkha Singh. 2021. Knowledge grounded multimodal dialog generation in task-oriented settings. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 421–431, Shanghai, China. Association for Computational Lingustics.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5881–5885.

Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building task-oriented dialogue systems for online shopping. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4618–4625, San Francisco, California, USA. AAAI Press.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. *An Analysis of BERT in Document Ranking*, page 1941–1944. Association for Computing Machinery, New York, NY, USA.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W. Black. 2018. A dataset for document grounded conversations. *CoRR*, abs/1809.07358.

How Much Does Prosody Help Turn-taking? Investigations using Voice Activity Projection Models

Erik Ekstedt
KTH Speech, Music and Hearing
Stockholm, Sweden
erikekst@kth.se

Gabriel Skantze
KTH Speech, Music and Hearing
Stockholm, Sweden
skantze@kth.se

Abstract

Turn-taking is a fundamental aspect of human communication and can be described as the ability to take turns, project upcoming turn shifts, and supply backchannels at appropriate locations throughout a conversation. In this work, we investigate the role of prosody in turn-taking using the recently proposed Voice Activity Projection model, which incrementally models the upcoming speech activity of the interlocutors in a self-supervised manner, without relying on explicit annotation of turn-taking events, or the explicit modeling of prosodic features. Through manipulation of the speech signal, we investigate how these models implicitly utilize prosodic information. We show that these systems learn to utilize various prosodic aspects of speech both on aggregate quantitative metrics of long-form conversations and on single utterances specifically designed to depend on prosody.

1 Introduction

Turn-taking is the fundamental ability of humans to organize spoken interaction, i.e., to coordinate who the current speaker is, in order to avoid the need for interlocutors to listen and speak at the same time (Sacks et al., 1974). A dialog can be viewed as a sequence of turns, constructed through the joint activity of turn-taking between the two speakers. A turn refers to segments of activity where a single speaker controls the direction of the dialog.

In conversational systems, turn-taking has traditionally been modeled using threshold policies which recognize silences longer than a chosen duration as transition-relevant places. Although these types of models are commonly used, it is well known that they are insufficient for modeling human-like turn-taking (Skantze, 2021). Studies of human-human conversation have shown that turns are frequently shifted with a gap of just 200ms (Levinson and Torreira, 2015), or even with a slight overlap. Thus, given that humans also need some time to prepare a response, it would be infeasible for humans to just use silence as a cue to turn-taking. Instead, it has been suggested that they are able to project turn completions already while the other person is speaking (Sacks et al., 1974; Levinson and Torreira, 2015; Garrod and Pickering, 2015). In addition, humans produce so-called *backchannels* (short feedback tokens such as "mhm") in a timely manner, often in overlap with the other speaker (Yngve, 1970).

A common research question in phonetics, psycho-linguistics, and conversational analysis concerns the various cues (including speech, gaze, and gestures) that humans use to detect or project turn-shifts (Duncan, 1972). When it comes to speech, a common distinction is made between the prosodic (non-lexical) and lexical (textual, syntactic, semantic) components of the speech signal. For example, De Ruiter et al. (2006) argued, based on listening experiments, for the importance of syntactic information over intonation (pitch) in turn-taking, while Bögels and Torreira (2015) showed that intonation is important when syntactic completion is ambiguous. However, such studies often require human listening experiments which are costly, anecdotal, and constrained in time resolution and are therefore limited to small amounts of conversational contexts. An alternative approach is to use computational models (Laskowski et al., 2019) to investigate what type of information they are sensitive to.

Ekstedt and Skantze (2022) recently proposed Voice Activity Projection, **VAP**, which is a general, self-supervised turn-taking model. The model incrementally projects the future speech activity of the two speakers directly from raw audio waveforms. The model can be trained on lots of data, without human annotations, and is agnostic with respect to different types of speech information, as it does not depend on explicitly extracted features. This makes the VAP model potentially suitable as a data-driven approach for investigating the role of

Proceedings of the SIGdial 2022 Conference, pages 541–551
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

prosody in turn-taking.

In this work, we train VAP models on a large dataset (Godfrey et al., 1992; Cieri et al., 2004) of dyadic spoken interactions and evaluate it on specific turn-taking metrics, while perturbing the input audio to omit certain sources of prosodic information. We analyze the performance over different tasks to investigate three research questions:

1. Do Voice Activity Projection models trained on raw waveforms learn to pick up prosodic information that is relevant to turn-taking?

2. When/how is prosodic information important for turn-taking predictions?

3. What is a suitable time resolution for such models to best represent prosody?

2 Background

Prosody refers to the non-verbal aspects of speech, including *intonation* (F0/pitch contour), *intensity* (energy), and *duration* (of phones and silences). It has been found to serve many important functions in conversation, including prominence, syntactic disambiguation, attitudinal reactions, uncertainty, topic shifts, and turn-taking (Ward, 2019). Studies on both English and Japanese have found that level intonation (in the middle of the speaker's fundamental frequency range) tends to serve as a turn-holding cue, whereas either rising or falling pitch can be found in turn-yielding contexts (Gravano and Hirschberg, 2011; Local et al., 1986; Koiso et al., 1998). When it comes to intensity, studies have found that speakers tend to lower their voices when approaching potential turn boundaries, whereas turn-internal pauses have a higher intensity (Gravano and Hirschberg, 2011; Koiso et al., 1998). Regarding duration and speaking rate, Duncan (1972) found a "drawl on the final syllable or on the stressed syllable of a terminal clause" to be a turn-yielding cue (in English). This is also in line with the findings of Local et al. (1986).

When it comes to lexical information, a very strong cue to turn-taking is of course whether the utterance is syntactically or pragmatically complete (Ford and Thompson, 1996). Thus, even if prosodic cues can be found near the end of a turn-shift, it is not clear to what extent such cues provide additional information compared to lexical cues, or if they are redundant. In an experiment by De Ruiter et al. (2006), subjects were asked to listen to a conversation and press a button when they anticipated a turn ending. The speech signal was manipulated to either flatten the intonational contour, or to remove lexical information by low-pass filtering. The results showed that the absence of intonational information did not reduce the subjects' prediction performance significantly, but that their performance deteriorated significantly in the absence of lexical information. From this, they concluded that lexical information is crucial for end-of-turn prediction, but that intonational information is neither necessary nor sufficient. Ekstedt and Skantze (2020) also found that it is possible to build fairly reliable turn-taking models using only lexical information.

However, it has also been argued that while lexical information is important for turn-taking, there are many cases where a phrase may be syntactically complete, but it is unclear whether the turn is in fact yielded or not (Ford and Thompson, 1996). To investigate this, Bögels and Torreira (2015) performed a similar experiment as De Ruiter et al. (2006), but selected the stimuli so that they contained several syntactic completion points (e.g. "Are you a student / at this university?"), and where the intonation phrase boundary provided additional cues to whether the turn was yielded or not. They found that subjects indeed made better predictions with the help of intonation and duration.

Most previous attempts at modeling prosody in turn-taking have been limited in that they (I) only use instances of mutual silence for predicting turn shifts (and therefore do not model projection of turn completion), and (II) only use fairly superficial, hand-crafted features, such as the extracted pitch slope or pitch level right before the pause (e.g., Gravano and Hirschberg 2011; Meena et al. 2014). Apart from the problem that such features might be too simplistic, they also typically require speaker normalization of the pitch (Zhang, 2018).

In this work, we investigate various forms of turn-taking events (including projection of both turn shifts and backchannels). We also use a more agnostic modeling approach, using latent speech representations that are learned in a self-supervised manner and extracted from the raw waveform (van den Oord et al., 2018). If our model is indeed able to pick up relevant prosodic information from these representations, it means that we do not have to do any special prosodic feature engineering or speaker normalization.

3 Voice Activity Projection Model

Ekstedt and Skantze (2022) proposed a generic turn-taking model that does not predict specific turn-taking events at specific moments in time. Instead, the model is given the task of Voice Activity Projection (**VAP**), which means that it has to incrementally predict the future voice activity (**VA**) of each interlocutor in a dialog. The prediction target at each incremental step is defined by a window of 2 seconds containing the future VA for both speakers. The window is discretized into 8 separate bins (4 for each speaker) where each bin is assigned a value of one if more than half of its frames are active, to produce an 8 bit binary digit, corresponding to 256 unique classes.

The VAP model consists of an encoder that processes raw audio waveforms, along with the current VA information, to produce latent representations of a defined frame frequency f_{enc}Hz which are then fed into the predictor network. The predictor is a causal sequence network that processes the context available up until the current frame and outputs a probability distribution over the 256 VA classes, see Figure 1.

The encoder consists of two sub-modules, a speech module which processes raw waveforms, x, specifically a CPC (van den Oord et al., 2018) model that outputs frame-level representations $h_{speech,t} \in \mathbb{R}^{256}$, at f_{enc} Hz. A second VA module, matching the frame rate of the speech encoder, processes the current VA frame vector $v_t^f \in \{0,1\}^2$, along with a concise representation of the VA history. The VA history features provide long-ranging contextual information outside of the receptive field of the acoustic model. This history is defined as the activity ratio of speaker A over speaker B for regions of size {-inf:60, 60:30, 30:10, 10:5, 5:0} seconds into the past, where 0 is the current time step, resulting in a vector $v_t^h \in \mathbb{R}^5$ with values between 0 and 1, for each frame. The VA module projects the VA features to vectors $h_{va,t}, h_{his,t} \in \mathbb{R}^{256}$ which are added to the speech representation $h_{speech,t}$ to produce the encoder output h_t, for each frame t. The dialog input waveforms are volume normalized, resampled to 16kHz, mixed to a single channel and split into 10s segments (using a 1s overlap).

The predictor consists of a causal, decoder only, transformer (Vaswani et al., 2017), with linear attention (Press et al., 2022), using a hidden size of 256, 4 layers, 8 heads, and 0.1 dropout. The output

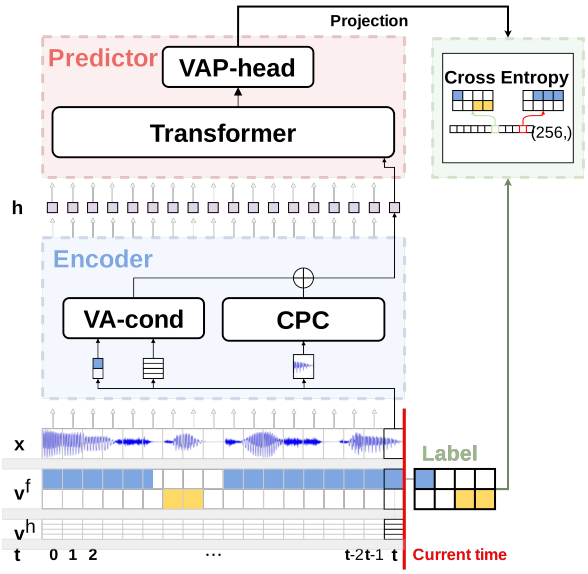


Figure 1: The VAP model processes the input features at time t. The input to the model is the combined speech waveforms of the two speakers (x_t), the VA frames of the window (v_t^f), and the longer VA history (v_t^h). The waveform and VA features are processed separately, projected to a common feature space, and added together to produce the predictor input, h_t. The predictor consists of a causal transformer feeding into the VAP-head to produce the output projection. The green box illustrates the various outputs of the different models that we compare. Source: (Ekstedt and Skantze, 2022)

of the transformer model is fed to the VAP head, a final linear layer, which outputs logits associated with the 256 VA classes. Since transformer models are powerful but come with the cost that they scale quadratically in compute, with respect to input length, we are interested in whether using a slower frame rate of the sequence model has any significant impact on the turn-taking performance. Following previous work, we utilize a pre-trained CPC (Rivière et al., 2020) encoder which produces output representations at 100Hz, and for two of our three models, we include a single additional convolutional layer which projects the representations to 50 and 20Hz. In other words, we train three models which use different frame rates of the predictor.

3.1 Turn-taking Metrics

The Voice Activity Projection in itself is just a distribution of 256 possible futures. However, Ekstedt and Skantze (2022) also showed how this distribution can be used to predict various turn-taking events as zero-shot classification tasks. We utilize three of these metrics, namely *Shift/Hold*, *Shift-prediction*, and *Backchannel-prediction*, and

will briefly explain them here.

Shift/Hold: This metric evaluates how well the model predicts the next speaker during mutual silence, i.e., whether the current speaker will Hold the turn, or whether the turn will Shift to the other speaker. The frames used for evaluation start 50ms into the silence, covering a total of 100ms consecutive frames.

Shift prediction: This metric evaluates how well the model can continuously predict an upcoming Shift in the near future, while a speaker is still active. We follow prior work and consider a range of 500ms that covers the end of a VA segment, before a Shift-event (as defined above), as positive samples. Similarly, we sample negative ranges, of the same duration, from regions where a single speaker is active but far away (2s) from any future activity of the other speaker.

Backchannel (BC) prediction: This metric evaluates how well the model can continuously predict an upcoming BC in the near future (similar to (Mueller et al., 2015; Ruede et al., 2017)). BCs are defined as short and isolated VA segments ($\leq 1s$), preceded by $\geq 1s$ of silence and followed by $\geq 2s$ of silence by the same speaker. We consider regions of 500ms before a BC as positive samples and the negatives are sampled similarly to the *Shift prediction* metric, with the addition of allowing for non-active segments, i.e., backchannels can be predicted during silences as well.

4 Training and Data

We train three different VAP models with different frame-level frequencies: 20, 50, and 100Hz. We use the combination of two dyadic conversational datasets, Switchboard (Godfrey et al., 1992) and Fisher[1] (Cieri et al., 2004), resulting in 8288 unique dialogs. We set aside a test set of 5% (of each dataset) and split the remaining dialogs into a 90/10 train/validation split used for training. We use the AdamW (Kingma and Ba, 2015; Loshchilov and Hutter, 2019) optimizer and an early stopping criteria on the validation loss with a patience of 10 epochs. The code is implemented in Python using the PyTorch (Paszke et al., 2019), PyTorch-Lightning (William et al., 2020) and Wandb (Biewald, 2020) libraries, and are publicly available[2].

[1]Because of limited access we only use Part 1 of the full corpus.

[2]https://github.com/ErikEkstedt/conv_ssl

4.1 Data perturbation

In order to investigate the role of prosody in the model's turn-taking predictions, we perturb the input audio waveform of the test data in five ways to omit parts of the signal encoding for various prosodic features:

F0 flat: the intonation contour is flattened to the average F0 of each speaker and segment.

Low pass: the signal is low pass filtered by down-/up-sampling of the waveform similar to Weston et al. (2021). This effectively removes all high-frequency phonetic information, while only the F0 and intensity contours are relatively intact. We use a cutoff frequency of 400Hz across all samples.

Intensity flat: The intensity contour is flattened to the average value of each speaker over all speech frames (as determined by the VA features). We note that this transformation is difficult to perform without including acoustic artifacts despite having access to speech boundaries given by the VA features. Breaths become very loud and the gain inside smaller segments of silence is prominent.

Duration average: Each phone in a segment is scaled to the average duration, of that specific phone, across the dataset.

F0 shift: The intonation contour is shifted by 90% of the original value for each speaker over each active speech segment. This should (in theory) not affect the turn-taking predictions. However, we include this perturbation to verify that the transform in itself does not have a too strong effect (e.g., through artifacts).

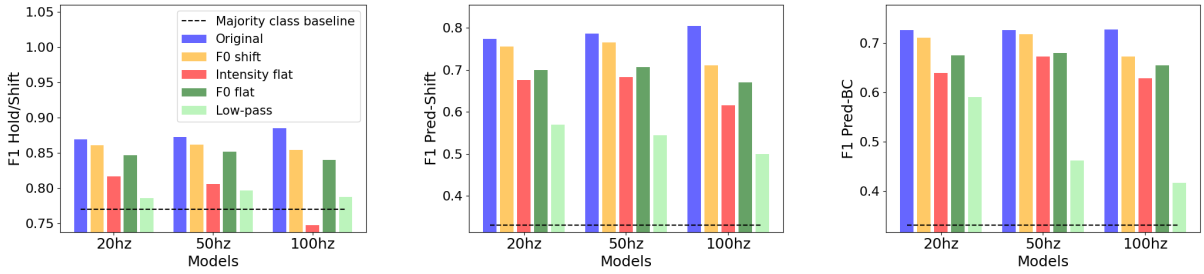
All perturbations were done using Praat (Boersma and Weenink, 2022; Jadoul et al., 2018) and the Torchaudio[3] library.

5 Aggregate Turn-taking Evaluation

In this experiment, we evaluate the models on the turn-taking metrics described in Section 3.1, on a withheld test set, using the original audio and the respective augmentations, with the exception of *Duration average*[4], listed above. The performance across models and metrics is visualized in Figure 2. We note that the Shift/Hold metric is highly imbalanced, containing a substantially larger amount of holds, indicated by the high baseline weighted F1 (≈ 0.77). The remaining metrics are balanced by design, resulting in a lower baseline value ($\approx .33$).

[3]https://pytorch.org/audio

[4]We do not have access to phone aligned annotations of the datasets.



(a) Shift/Hold (b) Shift prediction (c) Backchannel prediction

Figure 2: Aggregate results for the three tasks on the Switchboard and Fisher test set, depending on model frequency and perturbation. Majority class baseline is shown with the dashed black line.

The least intrusive augmentation across all models and metrics is, as expected, the *F0 shift* transformation. However, the artifacts introduced still seem to have some effect on the models. Interestingly, it has the greatest impact on the 100Hz model, indicating that a higher frame rate of the predictor model could make it more sensitive to detailed phonetic information disregarded by the slower versions.

On the Shift/Hold metric, all models are similarly and substantially impacted by the *Low pass* augmentation, lowering the performance towards baseline performance. This augmentation omits almost all information other than the F0 and intensity contours and shows that the model does rely on more complex cues to predict the next speaker. *F0 flat* interestingly has the least negative effect, across all models (disregarding *F0 shift*). This is surprising, given that pitch seems to be the most frequently used prosodic cue in computational turn-taking models. However, while *Intensity flat* severely affects the 100Hz model, making it worse than the baseline, it has a lesser effect than *Low pass* for the other two.

On the *Shift prediction* and *Backchannel prediction* tasks, where the evaluation point occurs inside of an ongoing utterance, all models are substantially affected by the *Low pass* transform, and the higher the frame rate of the model, the larger the impact. The transformation removes faster phonetic information obfuscating phones, words, and their durations (or boundaries), which are more discernible to models operating on higher frame rates, making the impact variation across models less surprising. However, this variation is greater on the *Backchannel prediction* task, with a large difference of effect between the 20 and 100hz models. The second most impactful perturbation is *Intensity flat*, which indicates, in accordance with the turn-taking literature in general, that shifts and backchannels are preceded by changes (arguably drops) in the intensity contour of the current speaker.

6 Utterance-level Analysis

While the analysis above gives an overall estimate of how important prosody is, it has been hypothesized that prosody is especially important when the semantic/pragmatic completion is ambiguous, as discussed in Section 2. To focus their analysis on such situations, Bögels and Torreira (2015) constructed question templates where a short and a long version, sharing initial lexical information, were recorded through scripted interviews (in Dutch). As an example, a short/long question pair "did you drive here?" and "did you drive here this morning?" contain the same initial words up to a common completion point (after the word "here"), which we will refer to as the *short completion point*, SCP. Note that in order for the listener (or the model) to predict a turn-shift towards the end of the short utterance, but not at the corresponding place in the long utterance, it has to rely on prosody. Through listening experiments, where the participants are asked to press a button when they expect a turn shift, Bögels and Torreira (2015) found that the reaction time was indeed much faster after the short version than after a long version cut after the SCP.

For our experiments, we created a similar set of 9 long/short utterance pairs in English (see Table 1 in the Appendix) using the Google TTS[5] service and produced 10 versions of each long/short pair using 5 male and 5 female voices. An example of such a

[5] https://cloud.google.com/text-to-speech

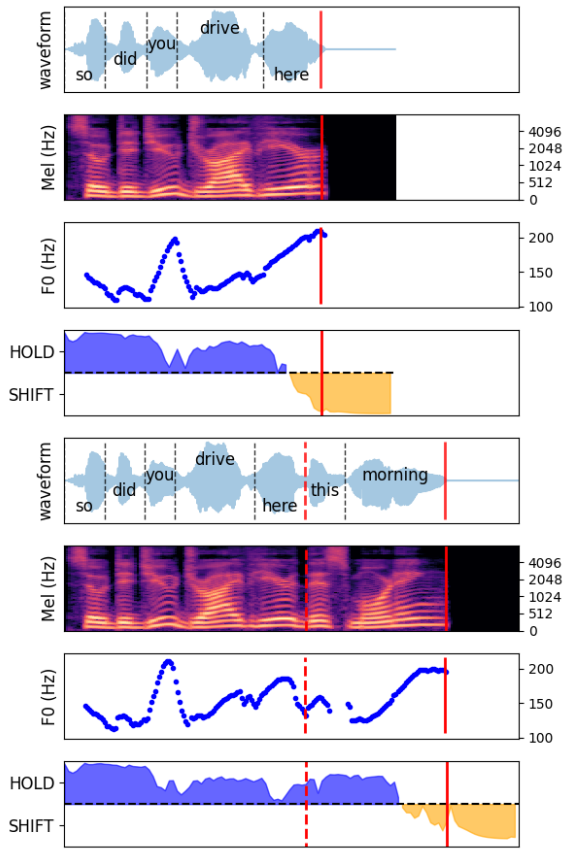
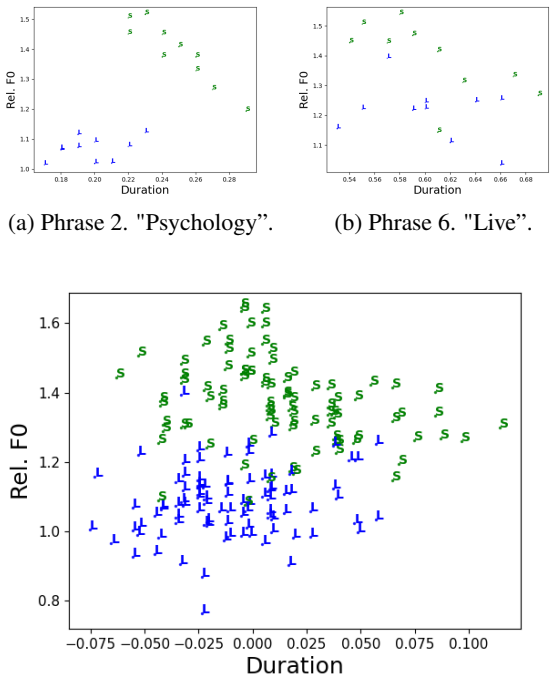


Figure 3: A short/long phrase pair. The plots show the waveforms, mel-spectrograms, F0 contours, and the model assigned Shift/Hold comparison, for the short and long versions respectively. The blue color in the bottom plots indicates a majority probability (over 50%) for Hold whereas the yellow indicates Shift. The short completion point (SCP) is shown as a red dashed line for the long utterance and the filled red line shows the end time of the last word in each utterance.



(a) Phrase 2. "Psychology". (b) Phrase 6. "Live".

(c) All phrases.

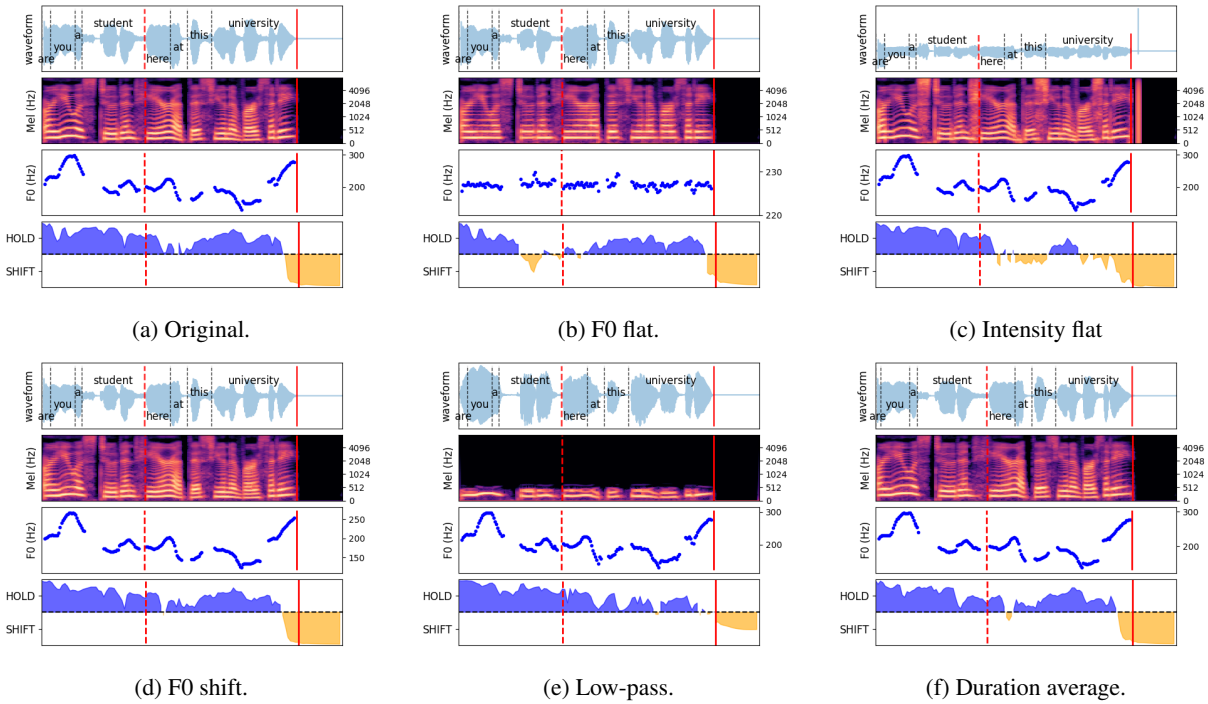
Figure 4: Duration and maximum relative F0 over the last syllable at the "short completion point" for the (L)ong and (S)hort versions of the synthesized voices. The x- and y-axis corresponds to mean-shifted duration and relative F0 peak.

pair is visualized in Figure 3. In the figure, we have also visualized the VAP model's *Shift prediction*, as described in Section 3.1.

As can be seen in the figure, for this example, the model correctly assigns a high probability to Hold until towards the end of each utterance, where it changes to Shift. This clearly illustrates the model's ability to project turn shifts before the utterance is complete, and before the large rise in final pitch has actually happened. In addition, we see how the model makes a clear distinction between the two utterances at the short completion point (SCP), where it predicts a Hold for the longer variant. This illustrates that the model is indeed sensitive to prosody, as that is the only information that is different up until that point. Additional samples and visualizations are publicly available[6].

Since we rely on artificially generated utterance pairs, we can of course not be certain to what extent they reflect similar prosodic patterns as those generated by humans. We therefore perform a similar analysis of the phrases as Bögels and Torreira (2015), by measuring the duration and maximum F0 frequency over the last syllable of the short completion point. In their analysis, they showed that longer duration and a higher rise in F0 are associated with the end of a turn, separating the measures at the SCP of the short phrase from the long, as shown in Figure 4a. We obtain similar distributions from 4 of our 9 phrases, but note that the others are not as easily separated, but show more uniform distributions for the duration dimension as shown in Figure 4b. However, from listening to the phrases, we still consider all recordings natural enough to be included in our further analysis. Although both duration and pitch might sometimes clearly indicate turn-shifts according to the literature, there is no guarantee that this is actually the case for

[6] https://erikekstedt.github.io/conv_ssl/



(a) Original.

(b) F0 flat.

(c) Intensity flat

(d) F0 shift.

(e) Low-pass.

(f) Duration average.

Figure 5: Model output from a female TTS voice saying “Are you a student here at this university?" (**long**).

all types of phrases. This indicates that simple models that only track these superficial features might not capture the whole picture. We provide the mean-shifted duration and relative F0 rise over all generated phrases in Figure 4c.

We compare the performance of the VAP model on the short and long versions of each phrase to investigate whether it can recognize the prosodic differences and correctly predict the short completion point as either a Hold (long phrase) or a Shift (short phrase). In addition to the original recordings, we include evaluations of the performance on the perturbed versions to investigate whether any specific perturbation changes the predictions of the model more than the others. We use the 50Hz model, as it performs comparably to the 100Hz model on the original audio, while being less affected by the *F0 shift* transform, indicating less sensitivity to arbitrary artifacts introduced by the perturbations.

The model output on the long version of the phrase “Are you a student here at this university?", for the various perturbations, is visualized in Figure 5. Inspection of the original performance in Figure 5a indicates that the model is sensitive to prosodic information and assigns a higher likelihood of a Hold at the SCP located on the word "student”. However, for the *F0 flat* perturbation, in Figure 5b, we note that the model flips and assigns a higher Shift-probability at the SCP, which indicates that if the dynamics of the F0 contour is omitted, the model cannot recognize that the speaker will continue to speak. Interestingly, the *Intensity flat* perturbation also affects the output of the model, but after the SCP is completed. Here, the model does have access to the F0 contour and correctly assigns a larger Hold-probability at the SCP, but then changes prediction to indicate that a Shift is probable following the word "here”. As a final note, the *Low pass* transform, which filters out all phonetic information while keeping both the intensity and F0 contour, does produce predictions close to that of the original audio, while being slightly less certain of a Shift after the entire utterance is completed, as seen in Figure 5e. We also provide the corresponding visualizations over the short version of the same speaker and phrase in Figure 7 in the Appendix.

To get an aggregate evaluation of the model across all 9 phrases and 10 voices, we define three regions in each utterance, up until the SCP point (for both long and short phrases), namely **hold**, **predictive** and **reactive**, and measure the average Shift probability predicted by the model in those regions. The *hold* region covers the start of the utterances until 200ms before the SCP, where the *predictive* region begins. The final *reactive* region is the very last frame of the SCP where the entire last word (of

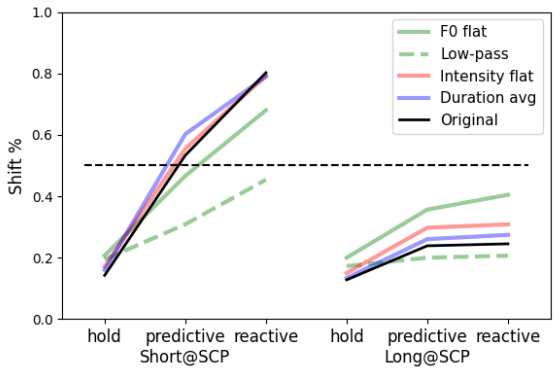


Figure 6: Shift probabilities for the 50Hz model on the short completion point over the *hold*, *predictive* and *reactive* regions over all short and long phrases.

the short utterance) has been processed. Over the long utterances, the model should consistently predict a low shift probability, given that the speaker will continue their turn, while the shift probabilities should increase over the regions of the short utterances. The aggregate model performance over all phrases is visualized in Figure 6.

The left part of Figure 6 displays the average Shift probabilities for the points on the SCP for the short phrases (Short@SCP) which preferably should start low and rise consistently. The right part of the figure shows the corresponding performance but on the long phrases (Long@SCP) and should be consistently low, indicating that the speaker will continue their turn. Looking at the non-perturbed signal (Original), and comparing the left and right figures, we see that the model is indeed sensitive to prosody, confirming the anecdotal observation from Figure 3. The *Low pass* transform clearly hinders the model from predicting a Shift, indicating that pitch and intensity in themselves are not enough. Among the other perturbations, *F0 flat* seems to have the largest negative effect, which confirms that intonation is important for disambiguating turn completion when lexical information is not enough. Duration seems to be less important, which aligns with the observation in Figure 4c.

7 Conclusion and Discussion

In this work, we train general computational models of turn-taking, provide analytical methods suitable for evaluating their performance on turn-taking tasks, and investigate how they utilize prosodic information in the speech signal. We investigate the models' reliance on prosody by extending psycho-linguistic experiments designed to measure the effect of prosody on turn-taking in human subjects. We conclude by addressing our three research questions below.

Do Voice Activity Projection models trained on raw waveforms learn to pick up prosodic information that is relevant to turn-taking? We apply specific prosodic perturbations to the input signal and show a deterioration across all models on the tasks of turn-taking and backchannel prediction, indicating that prosodic cues are utilized by the models. We note that phonetic information has the largest impact on these measures and that F0 information is less important for turn-taking in general. Even more convincing are perhaps the specific comparisons of the models' ability to predict Shift vs Hold at syntactic completion points, where the lexical information is identical. This task requires access to the prosodic dynamics of the signal and should be impossible to distinguish based on lexical information alone.

When/how is prosodic information important for turn-taking predictions? Overall, we show that all models are most sensitive to the *low-pass* transform, indicating that phonetic information is important for turn-taking in general. We note that intensity is at least as important as pitch when applied to actual human long-form conversations, but that pitch plays a more important role for the disambiguation at syntactically equivalent completion points. Interestingly, we note that the importance of duration plays a less important role, indicating that the F0-contour is the most reliable cue in the presence of lexical ambiguity. Another interesting observation in Figure 6 is that even if intonation seems to be the most important individual cue, flattening it does not completely collapse the distinction between turn-holding and turn-yielding. Thus, there must also be redundant information in intensity and/or duration. This shows that prosody is indeed a complex set of signals, which the model has captured.

What is a suitable time resolution for such models to best represent prosody? In our analysis of the turn-taking metrics, we note a negligible performance degradation when decreasing the frame rate of the predictor model. We note that high-frequency models tend to focus more on phonetic information, indicated by their sensitivity to the *Low pass* transformation. The faster models seem more sensitive to general acoustic artifacts, as indicated by the larger performance drop on the *F0*

shift perturbation, which should not have an impact on turn-taking cues in general. Overall, we favor the slower models given their lower memory and computational requirements, their robustness, and comparable performance.

It should be noted that the models were not trained on perturbed versions of the data, which include highly unnatural speech (i.e., no humans speak with a perfect flat intonation contour). Thus, the evaluations of Section 6 can be considered out-of-distribution. Nevertheless, it is interesting that for many of these perturbations, the models still perform relatively well. Also, the drops in performance are typically in line with what could be expected from the literature. For future work, it could be valuable to train multiple models, on data with different prosodic perturbations, and compare their performance for further analysis. Another interesting approach could be to identify actual instances of syntactically ambiguous phrases, rather than relying on TTS. Moreover, it would be interesting to include a larger linguistic context, and investigate whether the importance of prosody decreases.

8 Acknowledgments

This work was supported by Riksbankens Jubileumsfond (RJ), through the project *Understanding predictive models of turn-taking in spoken interaction* (P20-0484), as well as the Swedish Research Council, through the project *Prediction and Coordination for Conversational AI* (2020-03812).

References

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Paul Boersma and David Weenink. 2022. Praat: Doing phonetics by computer.

Sara Bögels and Francisco Torreira. 2015. Listeners use intonational phrase boundaries to project turn ends in spoken interaction. *Journal of Phonetics*, 52:46–57.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The fisher corpus: a resource for the next generations of speech-to-text. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Jan Peter De Ruiter, Holger Mitterer, and N. J. Enfield. 2006. Projecting the end of a speaker's turn: A cognitive cornerstone of conversation. *Language*, 82:515 – 535.

S Duncan. 1972. Some Signals and Rules for Taking Speaking Turns in Conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.

Erik Ekstedt and Gabriel Skantze. 2020. TurnGPT: a transformer-based language model for predicting turn-taking in spoken dialog. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2981–2990.

Erik Ekstedt and Gabriel Skantze. 2022. Voice activity projection: Self-supervised learning of turn-taking events. In *Proc. Interspeech 2022*.

Cecilia Ford and Sandra Thompson. 1996. Interactional units in conversation: syntactic, intonational, and pragmatic resources for the management of turns. In E Ochs, E Schegloff, and A Thompson, editors, *Interaction and grammar*, Studies in interactional sociolinguistics 13, chapter 3, pages 134–184. Cambridge University Press, Cambridge.

Simon Garrod and Martin J. Pickering. 2015. The use of content and timing to predict turn transitions. *Frontiers in Psychology*, 6:751.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, page 517–520, USA. IEEE Computer Society.

Agustın Gravano and Julia. Hirschberg. 2011. Turn-taking cues in task-oriented dialogue. *Computer Speech & Language*, 25(3):601–634.

Yannick Jadoul, Bill Thompson, and Bart de Boer. 2018. Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics*, 71:1–15.

Diederik. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR*.

Hanae Koiso, Yasuo Horiuchi, Syun Tutiya, Akira Ichikawa, and Yasuharu Den. 1998. An analysis of turn-taking and backchannels based on prosodic and syntactic features in Japanese map task dialogs. *Language and Speech*, 41:295–321.

Kornel Laskowski, Marcin Wlodarczak, and Mattias Heldner. 2019. A scalable method for quantifying the role of pitch in conversational turn-taking. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 284–292, Stockholm, Sweden. Association for Computational Linguistics.

Stephen C. Levinson and Francisco Torreira. 2015. Timing in turn-taking and its implications for processing models of language. *Frontiers in Psychology*, 6:731.

J Local, J Kelly, and W Wells. 1986. Towards a Phonology of Conversation: Turn-Taking in Tyneside English. *journal of Linguistics*, 22(2):411–437.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Raveesh Meena, Gabriel Skantze, and Joakim Gustafson. 2014. Data-driven models for timing feedback responses in a map task dialogue system. *Computer Speech & Language*, 28(4):903–922.

Markus Mueller, David Leuschner, Lars Briem, Maria Schmidt, Kevin Kilgour, Sebastian Stueker, and Alex" Waibel. 2015. Using neural networks for data-driven backchannel prediction: A survey on input features and training techniques. In *Human-Computer Interaction: Interaction Technologies*, pages 329–340. Springer International Publishing.

Adam Paszke et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux. 2020. Unsupervised pretraining transfers well across languages.

Robin Ruede, Markus Müller, Sebastian Stüker, and Alex Waibel. 2017. Enhancing Backchannel Prediction Using Word Embeddings. In *Proc. Interspeech 2017*, pages 879–883.

Harvey Sacks, Emanuel Schegloff, and G Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50:696–735.

Gabriel Skantze. 2021. Turn-taking in Conversational Systems and Human-Robot Interaction : A Review. *Computer Speech & Language*, 67:101178.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

Ashish Vaswani et al. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Nigel Ward. 2019. *Prosodic Patterns in English Conversation*. Cambridge University Press.

Jack Weston, Raphael Lenain, Udeepa Meepegama, and Emil Fristed. 2021. Learning de-identified representations of prosody from raw audio. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11134–11145. PMLR.

Falcon William et al. 2020. Pytorchlightning/pytorch-lightning: 0.7.6 release.

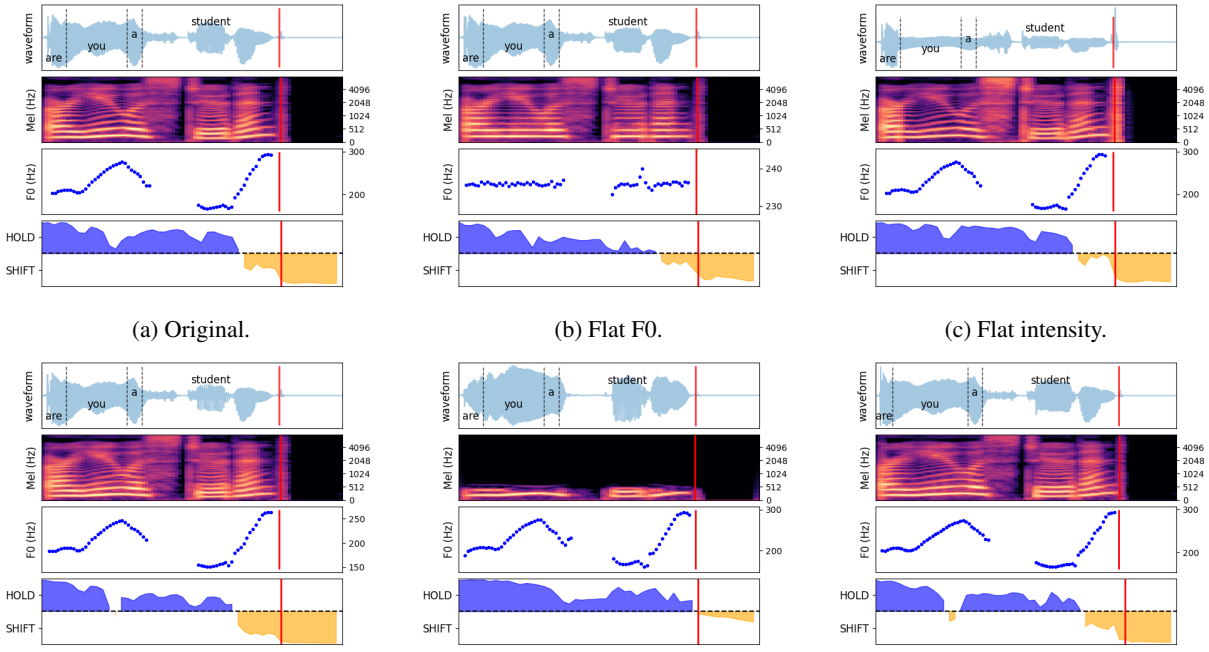
Victor H. Yngve. 1970. On getting a word in edgewise. *Chicago Linguistics Society, 6th Meeting, 1970*, pages 567–578.

Jingwei Zhang. 2018. A comparison of tone normalization methods for language variation research. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

A Appendix

Table 1: The 9 phrases used in the utterance-level analysis.

Item	Short	Long
1	Are you a student?	Are you a student **here at this university?**
2	Do you study psychology?	Do you study psychology **here at this university?**
3	Are you a first-year student?	Are you a first-year student **here at this university?**
4	So do you play basketball?	So do you play basketball **on Thursdays?**
5	Have you participated in any experiments before?	Have you participated in any experiments before **here at this university?**
6	Do you live by yourself?	Do you live by yourself **or with someone else?**
7	So you work on the side?	So you work on the side **in a supermarket in addition to your studies?**
8	Did you come here by bike?	Did you come here by bike **this morning?**
9	Did you drive here?	Did you drive here **this morning?**



(a) Original. (b) Flat F0. (c) Flat intensity.

(d) F0 shift. (e) Low-pass. (f) Duration avg.

Figure 7: Model output from a female TTS voice saying “Are you a student?" (**short**).

What makes you change your mind? An empirical investigation in online group decision-making conversations

Georgi Karadzhov
University of Cambridge
georgi.karadzhov@cl.cam.ac.uk

Tom Stafford
University of Sheffield
t.stafford@sheffield.ac.uk

Andreas Vlachos
University of Cambridge
av308@cam.ac.uk

Abstract

People leverage group discussions to collaborate in order to solve complex tasks, e.g. in project meetings or hiring panels. By doing so, they engage in a variety of conversational strategies where they try to convince each other of the best approach and ultimately reach a decision. In this work, we investigate methods for detecting *what* makes someone change their mind. To this end, we leverage a recently introduced dataset containing group discussions of people collaborating to solve a task. To find out what makes someone change their mind, we incorporate various techniques such as neural text classification and language-agnostic change point detection. Evaluation of these methods shows that while the task is not trivial, the best way to approach it is using a language-aware model with learning-to-rank training. Finally, we examine the cues that the models develop as indicative of the cause of a change of mind.

1 Introduction

Research in group decision-making has shown that a group that collaborates in order to make a decision can outperform even the most knowledgeable individual within it (Mercier and Sperber, 2011). People engage in discussions in a variety of settings, such as project meetings and study groups. In these scenarios, people incorporate a variety of conversational strategies to introduce their ideas and convince each other of them, aiming ultimately to reach a consensus. Fundamentally, before committing to a decision, most of the participants in a group have different ideas of what the correct answer might be, but through discussion they are able to convince each other, and ultimately some of the participants change their mind. While previous research has shown that people who reach a consensus tend to perform better at certain tasks (Navajas et al., 2018; Niculae and Danescu-Niculescu-Mizil, 2016; Concannon et al., 2015), *how* people reach a consensus is understudied. Successfully identifying what makes someone change their mind, is an important step in studying group dynamics, persuasion and collaboration.

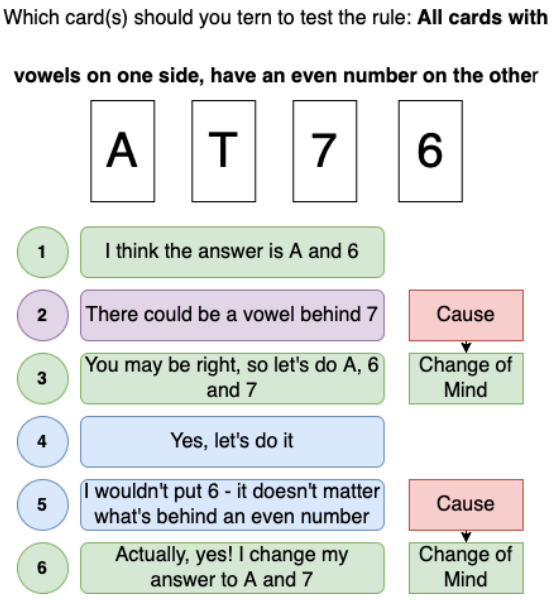


Figure 1: Sample conversation containing change of mind and what caused it. Participants are solving the Wason card selection task, where they should pick cards with letters and numbers on them.

In this work, we take advantage of a dataset previously introduced by us (Karadzhov et al., 2021), which contains group discussions of people solving a cognitive task. The dataset contains 500 dialogues, where people engage in various deliberation patterns to communicate their solution to the problem. The participants are presented with the Wason card selection task (Wason, 1968), which is a classic problem used in the study of decision making and has been useful in testing the potential benefits and mechanisms of group discussion (Maciejovsky et al., 2013). The Wason card selection task provides a controlled setup with quantifiable measures of success and improvement, which

Proceedings of the SIGdial 2022 Conference, pages 552–563
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

makes it very suitable for the study of individual biases and strategies. In the example in Figure 1, participants engage in a collaborative discussion where they iterate through 3 different solutions, where one of the participants changes their mind twice (in utterances 3 and 6). In the example, conversation utterances 2 and 5 are the arguments that cause that change of mind, and are the target utterances that we would like to predict. Put formally, in order to investigate *what* made someone change their mind in group decision-making conversations we formalise the task as detecting the utterance that causes the change of mind (or conversational turning point, which is used in this paper interchangeably).

In this work, we draw similarities between conversational turning points and change point detection. Change point detection investigates when a change will occur in a stream of data, and is traditionally applied in domains such as finance (Chen and Gupta, 1997; Oh and Han, 2001), engineering (Turner et al., 2013; Lai, 1995), climate data (Reeves et al., 2007; Khapalova et al., 2018), and genetics, (Wang et al., 2011; Hensman et al., 2013). Change point detection is concerned with either identifying a change post-hoc (offline change point detection or segmentation), or predicting a change point before it occurs - online change point detection (Adams and MacKay, 2007). In this work, we are concerned with the latter - identifying a change of mind before it occurs. We are doing this by trying to predict which utterance will cause a change of mind.

To evaluate our approaches, we develop a framework that quantifies the performance of models for change point detection in conversations, adopting practices from previous work (Burg and Williams, 2020). In terms of modelling, we first adapt a method for Bayesian online change point detection (Adams and MacKay, 2007), that was previously used in engineering and finance. This method is language-agnostic as it ignores any kind of linguistic cues. Next, we explore standard approaches to text classification as a method for predicting conversational turning points, showing that they are comparable to the language-agnostic models. We further improve on these methods, by investigating learning-to-rank training for the prediction of what causes a change of mind. We demonstrate that by altering the training procedure and by incorporating the RankNet loss (Burges et al., 2005), we can substantially improve over the language-agnostic and text classification approaches.

Overall, our results demonstrate that the task of detecting conversational turning points is feasible but not trivial. Approaches such as bag-of-words, or a simple all-positive baseline for change point detection have a performance of 0.18 area under the precision-recall curve. On the other hand, a combination of our learning-to-rank model and a positional prior led to an AUC of 0.25. Finally, we conclude this study with a qualitative analysis, where we demonstrate different patterns and linguistic phenomena that may indicate a cause of change of mind.

2 Related Work

The effect of conversation in group decision-making has previously been explored in the field of psychology. Both Navajas et al. (2018) and Mercier and Sperber (2011) show that there are conditions where a group of people who collaborate on a problem can outperform even the best individual group member. Moreover, previous research (Navajas et al., 2018; Niculae and Danescu-Niculescu-Mizil, 2016) has shown that groups of people who can reach a consensus through discussion have a higher group performance than just discussing or voting on a solution. Concannon et al. (2015) has found that disagreement markers at the beginning of the conversation lead to productive discussions. Likewise, both De Kock and Vlachos (2021) and Hallsson and Kappel (2020) study the effect that disagreement has on constructive conversations, showing how people who are disagreeing with each other can work together. Therefore, we hypothesise that it is interesting to study the conversations where someone changes their mind i.e. they disagreed at first but ultimately reach a consensus.

Other research is concerned with which specific linguistic phenomena are associated with conversations that can change someone's mind (or persuasive conversations). Zeng et al. (2020) investigate how topics and discourse change during a conversation, as well as their contribution to the persuasiveness of the conversation. Similarly, Hidey et al. (2017) analysed the prevalence of claims and premises in persuasive vs. non-persuasive dialogues. Both of these papers leverage the online forum *Change my View*, where participants argue pro and against a certain topic. Unfortunately, the topics discussed in this forum are open to interpre-

tation and personal opinion. Therefore, they do not have a clear quantitative measure of whether someone changed their mind. Further, while Zeng et al. (2020) and Hidey et al. (2017) study which phenomena may indicate that the conversation is more persuasive, they do not try to predict *when* will someone change their mind, and what is the specific utterance that caused that.

Identifying when a change in a sequence of observations occurs is traditionally studied in the context of change point detection, in the field of signal processing (Page, 1954; Truong et al., 2020). Formally defined, if we observe a sequence of a variable $[x_1, x_2, ..., x_n]$, a change point occurs when two adjacent elements of that observation differ substantially. Another way to define change points is by treating them as delimiters between different subsets of observed data. In this work, we adopt a version of the former definition - we are interested in the event that causes a subsequent observation to differ substantially from the previous ones.

In terms of methods, change point detection can be broadly divided into online and offline methods. Online methods (Adams and MacKay, 2007) focus on detecting a change point in a stream of data, and are evaluated based on the ability to predict a change point *before* it occurs (i.e. before the value changes substantially). On the other hand, offline (Smith, 1975; Green, 1995) methods by design work retrospectively on a sequence of datapoints, aiming at solving the task of segmentation. Offline methods incorporate bi-directional information to determine when a change point occurred (i.e. the data points before and after the change point), whilst online methods rely only on the observed information. In this work we focus exclusively on online change point detection, as we would like to predict what causes a change before we observe the change it causes. Arguably, detecting a change of mind post-hoc should be a more trivial endeavour, as s model could learn cues such as agreement markers or solution proposals.

A different way to approach this would be from the point of view of survival analysis and reliability engineering (Read and Vogel, 2016; Diamoutene et al., 2021; Nikulin et al., 2011). Previous research relies on the concept of hazard function (also referred to as "time-to-failure") which is defined as the instantaneous risk of an event occurring at a point in time. The premise is that, as more time passes, the likelihood of an event occurring increases. Practically speaking, in engineering the hazard function captures the intuition that as more time passes since the last maintenance, the likelihood of a breakdown of apparatus increases. We hypothesise that we observe a similar phenomenon in conversations – as the dialogue progresses, it is more likely for a participant to change their mind.

3 Data

In this work, we are investigating what makes someone change their mind in group decision-making. In order to select a dataset to work on, we considered the following factors:

- The dataset should contain group discussions.
- When engaging in conversation, the group should collaborate in order to reach a decision
- The conversation should have a quantifiable measure of success

With these criteria in mind, there are two datasets that could be used - a corpus of people playing a photography geo-location game (Niculae and Danescu-Niculescu-Mizil, 2016), or a dataset of groups playing the Wason card selection task (Karadzhov et al., 2021). Unfortunately, the former is not publicly available, so in this work we focus on the latter. The dataset was introduced by our previous work (Karadzhov et al., 2021), and it aims at evaluating how people collaborate and engage in deliberation (henceforth we refer to it as DeliData). Each group is presented with 4 cards, each having a letter or a number on it. Then, the participants have to answer the question "Which card(s) should you tern to test the rule: **All cards with vowels on one side, have an even number on the other**" (see Figure 1). The intuitive but wrong answer to the question is to turn the vowel and the even number, which is due to confirmation bias and is the most common answer given to the task. The correct answer is to turn over the vowel and the odd number.

In our experimental setup in DeliData (Karadzhov et al., 2021), we formed groups of 2 to 5 participants, first asking each member of the group to solve the task on their own. Then all of the participants engaged in a discussion about the task, being able to submit intermediate and final solutions. Each participant, apart from payment for their participation, was offered a bonus for submitting the correct solution, i.e. selecting the

correct cards. A conversation is successful if the final solutions submitted by each group member were on average more correct (in terms of number of cards selected correctly) than the initial ones before the conversation took place. In DeliData (Karadzhov et al., 2021), we found that after discussing the solution, 64% of the groups perform better at the Wason task, compared to their initial performances. Moreover, in 43.8% of the groups who had at least one correct answer as their final solution, none of the participants had solved the task correctly by themselves, thus confirming the hypothesis that groups can perform better than even the most knowledgeable individual.

Statistics of the DeliData corpus are presented in Table 1. DeliData contains 500 dialogues, with an average length of 28 utterances per dialogue. Additionally, 50 of those dialogues are annotated with deliberation cues and other conversational phenomena, such as argument structure or when someone proposes a solution. In this work, we use the solution proposals as an indication of *when* someone changes their minds, thus helping us identify *what* made them change their mind. These annotations were carried out by 3 annotators in a controlled setting, with a high inter-annotator agreement (0.5-0.75 Cohen's kappa).

# of dialogues	500
# of utterances	14003
AVG group size	3.16
# of dialogues with intermediary submissions	220
# of intermediary and final submissions	1179
# of annotated dialogues	50
# of annotated change of mind	262

Table 1: DeliData statistics

3.1 Gold data

In order to evaluate what made someone change their mind, we take advantage of the 50 dialogues manually annotated by Karadzhov et al. (2021). If an utterance contains a solution proposal that is different to the previously proposed solution by the same participant, it is considered an expression of a change of mind. Leveraging this annotation, our gold data is defined as follows: Given an utterance that expresses a change of mind, we select the last utterance made by a different person as the utterance that caused this change of mind. In Figure 2 (left), the 3rd utterance is an expression of a change of mind, annotated in DeliData. Therefore, the last utterance not said by participant U1, would be considered what caused the change of mind.

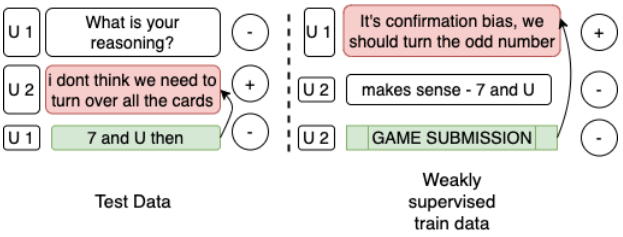


Figure 2: Test (left) and weakly supervised training (right) data for what caused a change of mind. The circles on the right of each example show the annotation used in our experiments: + denotes an utterance that caused a change of mind

3.2 Weakly supervised training set

Given that the gold annotated data is limited, we devised a way to leverage the unannotated data as a weakly annotated training set. For the 450 unannotated dialogues, each participant had to submit at least 1 solo solution, and 1 final solution. In 220 of these dialogues, at least 1 user had submitted an intermediate submission, thus we consider these dialogues as our training data.

Following the approach used for the gold data, we consider these weak annotations with a similar rule: for every submission expressing a different solution, we select the last utterance not made by the same user as the utterance that made them change their mind. In the example in Figure 2 (right), participant U2 made a submission, but because the last utterance before the submission was made by them, we mark the utterance by participant U1 as the one that made them change their mind.

As already mentioned, for this weakly supervised data we assume that every time a participant submits a new solution to the game, it can be attributed to the most recent utterance by a different participant. While this is reasonable, the reverse is not true - we can't be sure that if someone changed their mind, they submitted a new solution. Hence there will be utterances that could have caused a change of mind, but are not annotated as such. Therefore in our training protocol we take into account this limitation by proposing the learning-to-rank training described in Section 4.3.

4 Models

4.1 Language-agnostic models

First, we consider modelling options that do not utilise the language directly, but rather rely on proxy signal to predict when a change point would occur. In particular, we investigate 3 variants for language-agnostic change point detection - Hazard Function, Sequence length probability, and Bayesian Online Changepoint Detection (Adams and MacKay, 2007).

4.1.1 Hazard Function

The hazard function encapsulates the intuition that an event is more likely to occur the more time passes. It is defined as the probability of an event happening now, divided by the sum of probabilities of the event happening in the future. To calculate the hazard function we use Equation 1, where Tcp denotes the number of time steps since the last change point, and $P(X_{Tcp} = CP)$ is the probability of change point occurring at time step T:

$$H(Tcp) = \frac{P(X_{Tcp} = CP)}{\sum_{t=Tcp}^{\infty} P(X_t = CP)} \quad (1)$$

In the calculation of the hazard function, we consider only the distance from the last change point, thus disregarding information at what point of the conversation we are. Essentially, every time a change point occurs, the function starts over. For example, if a change point occurs in the 9th utterance of a conversation, the probability at the 10th utterance would be the same as in the first utterance.

4.1.2 Sequence length Probability

Recognising that it is important to model not only the information since the last change point, but to also consider information about how many utterances have been exchanged in the conversation as a whole, we propose an alternative model - sequence length probability. The assumption behind this method is that conversational turning points are more likely to occur at certain time steps in a conversation. For example, people may change their minds more at the end of a conversation rather than just after the first few utterances. This approach estimates the likelihood of encountering a change point at a specific time step *since the beginning* of the observed process. To model that, we calculate what is the chance for a change point occurring at time step T.

4.1.3 Bayesian change point detection

Adams and MacKay (2007) proposed a Bayesian approach to modelling when a change point would occur. Their method performs a prediction based on two variables - time since the last change point (similarly to the hazard function) and an observed variable at each time step. In the case of the DeliData dataset, we extract the observed variable from a method we call **solution tracker**, which gives an estimate what is the group's performance at every utterance. The solution tracker keeps a record of the solution proposed by each participant and then averages their individual score to calculate the group performance. The solution tracker first records each of the participants' solo submission. After the group phase starts, every time a participant mentions one of the 4 cards or the words '*all*' and '*none*', the solution tracker recalculates participant's individual score, as well as the aggregated group performance. The solution tracker incorporates a fairly simplistic rule-based approach to tracking solutions, and is thus imperfect. Nevertheless, it is a reasonable measure to track as it is a proxy for team performance.

Following the approach introduced by Adams and MacKay (2007), we are interested in two probabilities - the growth probability, indicating that a change point will not occur in the next time step (Equation 2), and the change point probability, showing that a change point would occur (Equation 3).

$$P(X_{T+1} \neq CP) = P_r(T-1)\pi_T^{(r)}(1 - H(r_{t-1})) \quad (2)$$

$$P(X_{T+1} = CP) = \sum_{r_{t-1}} P_r(t-1)\pi_t^{(r)} H(r_{t-1}) \quad (3)$$

These probabilities are computed using:

- $P_r(t-1)$ - run length estimation, which is the probability of the length of the run since the last change point, given the observed data and the current time step
- $\pi_t^{(r)} = P(X_1..X_t)$ - predictive probability, i.e. how likely is to observe a specific sequence of values

- $H(r_{t-1})$ - hazard function, as described in section 4.1.1 and equation 1

It is important to note that Adams and MacKay (2007) consider the model parameter before and after the change point as independent of each other, thus any positional information is lost.

4.2 Text-based Models

We recognise that the output of the solution tracker is unlikely to contain all the information needed to determine whether an utterance would cause someone to change their mind, hence we experimented with linguistic models to perform this prediction. We use a neural network, where the input is the last two utterances of a certain time step in a conversation, and the predicted output is whether or not we will encounter a change of mind in the next time step. Henceforth we will refer to this model as the **linguistic model**.

4.3 Learning to rank training

Given that changes of mind are often not stated by the participants, we presume that the annotation of the utterances that causes them would be incomplete, and we will be dealing with a lot of false negatives in the training. Thus, we propose to use learning to rank as follows: given a pair of inputs, one that is annotated as a cause of a change of mind and one that is not, we use the model to score the positive input higher than the negative one. In other words, even if both of the inputs are predicted as not causing a change of mind, we adjust the loss so that the positive sample should be ranked higher than the negative one.

Since the positive class is substantially less prevalent than the negative (most utterances do not change minds), we only need as many negative samples as there are positive ones to construct the positive-negative pairs. To do this, we devise the following algorithm:

- For each positive input (an utterance causing a change of mind) in a dialogue, we select a random negative input from the same dialogue.
- When selecting a random negative input, we consider those that are with a distance greater than 2 utterances from the nearest change point. This allows us to select safer negative inputs, as opposed to those that might carry a partial signal of the cause of change of mind.
- For every training epoch we change the random seed for the selection of the negative sample while keeping the same positive samples.

Using this algorithm, the positive samples are consistent throughout the training, while we vary the negative ones.

Having this training procedure, we consider **RankNet** loss (Burges et al., 2005), presented on Equation 4, which is a modified logistic function on probabilities from Baum and Wilczek (1987). This loss provides a probabilistic ranking cost function, which relies only on the difference between the positive and the negative samples.

$$C(pos, neg) = 1 - \frac{e^{(pos-neg)}}{1 + e^{(pos-neg)}} \quad (4)$$

The inspiration for this type of training was drawn from a different area in machine learning research - recommender systems. There, a single user will interact with a limited number of items from the pool of available ones. For even fewer of those, the user would have provided positive feedback. Therefore there will be items that the user would like to see more of, but they have not provided positive feedback, hence having incomplete annotation. When detecting a cause for a change of mind, we observe similarly incomplete annotation - not every time someone changed their mind, they have expressed it in the conversation explicitly, and thus we cannot label which utterance would be the cause for that change of mind. However, similarly to recommender systems, positive feedback while rarer is a strong indication of when a change of mind occurs. Therefore, in this work, we propose to approach the task of predicting conversational turning points using a learning to rank training objective, rather than as standard classification.

5 Evaluation

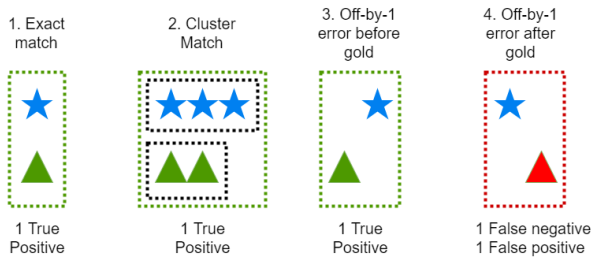


Figure 3: Four scenarios for change of mind evaluation. Blue stars denote the gold labels, while the triangles show the predicted values. With green borders and triangles, we show where the predicted and gold values match, and with red where we have an inaccuracy.

In this work, we devise a novel way for evaluating what caused a change of mind. We propose 3 key properties that our evaluation method should exhibit (with corresponding examples on Figure 3):

- The method should reward exact matches, i.e. when the gold and the predicted cause of change of mind align perfectly. (Scenario 1 on Figure 3)
- In cases where we observe a cluster of causes of changes of minds, we would like our method to (i) give full credit if we predict at least 1 of the gold utterances in the cluster, and (ii) if we predict all of them, to not "inflate" the score, giving full credit for each. An example of that is presented in Figure 3, scenario 2 – the gold and the predicted labels are clustered and aligned. Given that we have alignment between the two clusters, the method should count this as 1 true positive.
- In order to provide a more relaxed evaluation, our method should allow for small-margin errors. Given the length of each dialogue, we set the margin to 1. That said, we should count a true positive for off-by-one errors before the cause of change of mind (scenario 3), but the method should not allow for off-by-one errors after the gold label (scenario 4).

Given these desiderata, we consider how previous work evaluates change point detection methods. One approach (Killick et al., 2012) is to evaluate such methods as a regular machine learning model - the predicted and gold events should match exactly, in order to count the change point as true positive. This would cover scenario 1 (exact match), will count 2 true positives and 1 false negative for scenario 2 (cluster match) and will count the off-by-one predictions as errors. Some approaches (Martin et al., 2004) for change point detection evaluation recognise that nearly predicting a change point is good enough in practice, thus allowing for off-by-one errors. In Figure 3, both scenarios 3 and 4 are concerned with off-by-one errors, and previous work would categorise both of these as true positive, which would be incorrect (as we are not allowing off-by-one errors after the gold label).

Taking into consideration the desired properties of our evaluation method, as well as the limitations of previous research, in this work we use the following evaluation procedure:

1. We identify all clusters in the predicted and in the gold sequences, by grouping instances that are consecutive.
2. We perform alignment to identify which clusters overlap. We consider 2 clusters aligned if they overlap by at least 1 element.
3. We identify all matches between the gold and predicted pairs. If we encounter a gold label or cluster of labels, we check the prediction at the current and the previous time steps. If there is a match, we consider this pair a true positive.
4. After we iterate through the gold-prediction sequences, we mark every gold utterance that was not matched to a prediction as a false negative. Likewise, every predicted utterance that did not match a gold label, is considered a false positive.

Given this training procedure, we are able to have a list of true positive, false positive and false negative cases for our test set, allowing us to calculate class measures such as area under the precision-recall curve, and break-even precision-recall point.

6 Experimental Setup

Using the gold and the weakly supervised sets introduced in section 3, we train all of our models with the following setup. All models are trained on the 220 dialogues from the weakly supervised set. The 50 gold annotated dialogues are split into test and validation sets, of 40 and 10 dialogues respectively. The validation sets are exclusively used for model selection of the text-based models.

To train the linguistic and learning-to-rank models, we used a similar training setup. Both models embed the input using the BART embedding layer (Lewis et al., 2019). Following, we added two fully-connected layers with size of 1024 and 0.3 dropout between each of the layers. Finally, we use a simple sigmoid function to perform the final classification. Both models are trained using a batch size of 32 using the Adadelta optimizer (Zeiler, 2012) and were trained for 100 epochs, saving the iteration that has the best area under the precision-recall curve.

7 Results

On Table 2 we compare all of the models introduced in section 4, together with two baseline models. As a naïve baseline, we predict that every utterance in the conversation will lead to a change of mind. Also, we use off-the-shelf text classification methods, to provide a basic baseline for a linguistic model, namely a bag-of-words approach,

Model	Micro AUC	Macro AUC	Cutoff
Baseline: All positive	0.07	0.07	N/A
Baseline: Bag of Words	0.19	0.20	0.21
[1] Hazard Function	0.16	0.17	0.16
[2] Sequence Length	0.17	0.17	0.20
[3] BOCP (Adams and MacKay, 2007)	0.18	0.21	0.22
[4] [2] + [3]	0.21	0.23	0.26
[5] Linguistic Model	0.20	0.20	0.23
[6] Linguistic Model + [4]	0.22	0.22	0.24
[7] Linguistic Model (Learning to Rank)	0.23	**0.26**	0.24
[8] Linguistic Model (Learning to Rank) + [4]	**0.25**	**0.26**	**0.30**

Table 2: Evaluation of different methods for detecting what causes a change of mind in group discussions

paired with a Random Forest classifier (Ho, 1995).

We use 3 evaluation measures to compare the models - micro (utterance level) and macro (dialogue level) averaged area under the precision-recall curve, and the precision-recall cutoff point - the point where the precision and recall are equal. The reason to use these evaluation measures is three-fold. First, since change point detection typically deals with very imbalanced data, we need measures that are robust when the class of interest is under-represented. When dealing with heavily skewed data, Davis and Goadrich (2006) argue that the area under the ROC curve gives an overly optimistic estimate of the performance, and thus area under the precision-recall curve is a more appropriate measure. Secondly, while evaluating our models, we noticed that different models have different precision and recall characteristics. For example, some of our models had very high precision, or very high recall, whilst producing comparable F-measures. In order to give a fairer comparison of the overall model performance, we report the micro- and macro- average area under the precision-recall curve. Finally, while area-under-the-curve gives a good estimation of performance, it doesn't give a lot of intuition of how the model will perform in terms of precision and recall when used in a practical setting. Thus, we also report the precision-recall break-even point to show the relative predictive power of each model.

In Table 2 we show that all of the methods outperform the "all positive" baseline. That said, using the hazard function and the sequence length probability by themselves are the worst performing methods. Better performance is achieved by using a more sophisticated language-agnostic modelling, the Bayesian online change point detection (BOCP) (result 3). This approach takes into account the hazard function as well as a proxy for conversation performance, thus allowing for better modelling. While these approaches are reasonable, they are unable to capture language such as the arguments being made, which may cause lower performance. Interestingly, the bag-of-words model performs similarly to the significantly larger neural linguistic model which is trained on top of BART (Lewis et al., 2019) (result 5).

The best performing stand-alone model is achieved by training the linguistic model in a learning-to-rank setup (result 7), achieving Micro and Macro averaged AUC of 0.23 and 0.26 respectively.

Further, we experimented with combining language-independent and language-agnostic models. In the context of this paper, we incorporated a simplistic combination - if either of the combined models predicts a conversational turning point we consider this as a positive signal. Analysing the results, we observe that incorporating the sequence length and the BOCP (Adams and MacKay, 2007) with all linguistic models can yield a substantial improvement. By combining the sequence length with the Neural Learning-to-Rank model, we improve the performance to 0.25 micro AUC and 0.30 P-R break-even point.

In summary, while the neural models provide good stand-alone performance, they don't capture all of the information required for a prediction of a conversational turning point. Namely, a substantial signal is carried by a positional information of where you are in a dialogue (captured by the sequence length probability), as well as patterns in how people discuss solutions (Bayesian online change-point detection).

8 Qualitative Study

In order to gain some understanding of how each of the methods works, we qualitatively evaluate

models' predictions. Full conversation and model predictions are presented in appendix A. We use LIME (Ribeiro et al., 2016) to find out which words are indicative for the positive predictions. We incorporated LIME's to explain the prediction of the positive (cause of change of mind) class. The way method works is by first randomly perturbing features from the input, and then by learning linear models on the neighbourhood data to explain the label of interest. Using this workflow, we extracted common words for each of the methods and for the rest of the section we present some of the findings.

If we consider a pair of utterances that contain group interaction in the form of user mention: "*utt1: <MENTION> any ideas ? utt2: but then again most people get this wrong then it cant be as easy as we think surely*". Here both the bag-of-words baseline and the neural linguistic model classified the second utterance as a cause of change of mind. Interestingly, the models gave weight to different features. The bag-of-words identified words such as "easy", "people" and "wrong" as important, which are part of an argument. On the other hand, the neural linguistic model put by far the highest weight on the participant mention, which is not related to the task at hand, but rather to the group dynamics. This observation is also supported by previous research (Niculae and Danescu-Niculescu-Mizil, 2016; Woolley et al., 2010), which argues that group dynamics play important role in collaboration.

Looking into the cases where one of the models predicted a cause of change of mind one utterance before the cause (as we allow for off by 1 errors), we consider the following pair of utterances: "*then yeah we d have to make sure two vowels or two even numbers appear <SEP> so i think you'd just need to turn over <CARD> and <CARD>*". Here the neural learning-to-rank model, predicted a cause of change of mind, and some of the words with the highest weight were "odd", "turn", and "need". We hypothesise that the model learned to recognise argument markers as suggestive for a cause of future change of mind. Similarly, in the example "<CARD> is not an even we know tat <SEP> that*" the learning-to-rank model put higher weights on the words "even", <CARD> and "know".

Generally, the qualitative analysis shows that our best model (learning-to-rank) learnt to recognise argument cues as indicative of a conversational turning point. The model identified words that are related to the task such as card mentions or specific terms of the Wason card selection task. That said, this could be a drawback - it is unclear how such models would perform for a different task, where the vocabulary is substantially different.

9 Conclusions

In this work, we investigated methods for detecting the utterances that make someone change their mind, in the context of a recently introduced dataset containing group discussions of people collaborating to solve a task. We demonstrate that the best performance is achieved by combining a text-based model with a language-agnostic ones (such as positional information). In future work, we want to leverage the proposed approach to develop a system that can generate utterances that cause a change of mind in order to enhance group decision-making.

Acknowledgements

The authors would like to acknowledge the support of the Isaac Newton Trust and Cambridge University Press in creating the DeliData dataset (Karadzhov et al., 2021). Georgi Karadzhov is supported by EPSRC doctoral training scholarship. Tom Stafford and Andreas Vlachos are supported by the EPSRC grant no. EP/T023414/1: Opening Up Minds.

References

Ryan Prescott Adams and David JC MacKay. 2007. Bayesian online changepoint detection. *stat*, 1050:19.

Eric Baum and Frank Wilczek. 1987. Supervised learning of probability distributions by neural networks. In *Neural information processing systems*.

GJJ Burg and CKI Williams. 2020. An evaluation of change point detection algorithms. *arXiv preprint arXiv:2003.06222*.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96.

Jie Chen and Arjun K Gupta. 1997. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical association*, 92(438):739–747.

Shauna Concannon, Patrick GT Healey, and Matthew Purver. 2015. Shifting opinions: An experiment on

agreement and disagreement in dialogue. *SEMDIAL 2015 goDIAL*, page 15.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240.

Christine De Kock and Andreas Vlachos. 2021. I beg to differ: A study of constructive disagreement in online conversations. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2017–2027.

Abdoulaye Diamoutene, Farid Noureddine, Bernard Kamsu-Foguem, and Diakarya Barro. 2021. Reliability analysis with proportional hazard model in aeronautics. *International Journal of Aeronautical and Space Sciences*, pages 1–13.

Peter J Green. 1995. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

Bjørn G Hallsson and Klemens Kappel. 2020. Disagreement and the division of epistemic labor. *Synthese*, 197(7):2823–2847.

James Hensman, Neil D Lawrence, and Magnus Rattray. 2013. Hierarchical bayesian modelling of gene expression time series across irregularly sampled replicates and clusters. *BMC bioinformatics*, 14(1):1–12.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. 2017. Analyzing the semantic types of claims and premises in an online persuasive forum. In *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark. Association for Computational Linguistics.

Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Georgi Karadzhov, Tom Stafford, and Andreas Vlachos. 2021. Delidata: A dataset for deliberation in multi-party problem solving. *arXiv preprint arXiv:2108.05271*.

Elena A Khapalova, Venkata K Jandhyala, Stergios B Fotopoulos, and James E Overland. 2018. Assessing change-points in surface air temperature over alaska. *Frontiers in Environmental Science*, page 121.

Rebecca Killick, Paul Fearnhead, and Idris A Eckley. 2012. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Tze Leung Lai. 1995. Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(4):613–644.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Boris Maciejovsky, Matthias Sutter, David V Budescu, and Patrick Bernau. 2013. Teams make you smarter: How exposure to teams improves individual decisions in probability and reasoning tasks. *Management Science*, 59(6):1255–1270.

David R Martin, Charless C Fowlkes, and Jitendra Malik. 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549.

Hugo Mercier and Dan Sperber. 2011. Why do humans reason? arguments for an argumentative theory. *Behavioral and brain sciences*, 34(2):57–74.

Joaquin Navajas, Tamara Niella, Gerry Garbulsky, Bahador Bahrami, and Mariano Sigman. 2018. Aggregated knowledge from a small number of debates outperforms the wisdom of large crowds. *Nature Human Behaviour*, 2(2):126–132.

Vlad Niculae and Cristian Danescu-Niculescu-Mizil. 2016. Conversational markers of constructive discussions. In *Proceedings of NAACL-HLT*, pages 568–578.

Mikhail Nikulin, Noureddine Saaidia, and Ramzan Tahir. 2011. Reliability analysis of redundant systems by simulation for data with unimodal hazard rate functions. *Journal | MESA*, 2:277–286.

Kyong Joo Oh and Ingoo Han. 2001. An intelligent clustering forecasting system based on change-point detection and artificial neural networks: Application to financial economics. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 8–pp. IEEE.

Ewan S Page. 1954. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115.

Laura K. Read and Richard M. Vogel. 2016. Hazard function analysis for flood planning under nonstationarity. *Water Resources Research*, 52:4116 – 4131.

Jaxk Reeves, Jien Chen, Xiaolan L Wang, Robert Lund, and Qi Qi Lu. 2007. A review and comparison of changepoint detection techniques for climate data. *Journal of applied meteorology and climatology*, 46(6):900–915.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Adrian FM Smith. 1975. A bayesian approach to inference about a change-point in a sequence of random variables. *Biometrika*, 62(2):407–416.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing*, 167:107299.

Ryan D Turner, Steven Bottone, and Clay J Stanek. 2013. Online variational approximations to non-exponential family change point models: with application to radar tracking. *Advances in Neural Information Processing Systems*, 26.

Yao Wang, Chunguo Wu, Zhaohua Ji, Binghong Wang, and Yanchun Liang. 2011. Non-parametric change-point method for differential gene expression detection. *PloS one*, 6(5):e20060.

Peter C Wason. 1968. Reasoning about a rule. *Quarterly journal of experimental psychology*, 20(3):273–281.

Anita Williams Woolley, Christopher F Chabris, Alex Pentland, Nada Hashmi, and Thomas W Malone. 2010. Evidence for a collective intelligence factor in the performance of human groups. *science*, 330(6004):686–688.

Matthew D. Zeiler. 2012. Adadelta: An adaptive learning rate method. *ArXiv*, abs/1212.5701.

Jichuan Zeng, Jing Li, Yulan He, Cuiyun Gao, Michael Lyu, and Irwin King. 2020. What changed your mind: The roles of dynamic topics and discourse in argumentation process. In *Proceedings of The Web Conference 2020*, pages 1502–1513.

A Dialogue example and model predictions

Utterance	Gold	OCP	Hazard	SeqLen	Ling	BoW	L2R
What did you guys say was the answer ?	1	0	0	0	1	0	0
<CARD> is not an even we know tat	0	0	0	0	0	0	0
that *	0	0	0	0	0	0	1
i put <CARD> and <CARD> , you ?	1	0	0	0	0	0	0
<CARD> , <CARD> and <CARD>	0	0	0	0	0	1	0
Why did you think it was n't <CARD> ?	1	1	0	0	0	0	0
i chose all 4 cards so clearly mine was n't the one	0	0	0	0	0	0	0
Urm i m thinking	0	0	0	0	0	0	0
It might be right , we need to discuss	0	0	0	0	0	0	0
what do they exactly mean by turn	0	0	0	0	0	0	0
turn over ?	0	0	0	0	0	0	0
yeah	0	0	0	0	0	0	0
I assumed so	0	0	0	0	0	0	0
So what reasoning did you guys use for the cards you picked	0	0	0	0	0	0	0
they said most peope get this wrong so i m just wondering if they are trying to be cheeky by rotating them	0	0	0	0	0	0	0
why did you guys put your answers down ?	1	0	0	0	0	0	1
No , I think it means turning them over like onto the other side	0	0	0	0	0	0	0
Okay , I thought we need <CARD> because we need to see if there is a vowel on the other side	0	0	0	0	0	0	1
The same for <CARD> but the other way around	0	0	0	0	0	0	0
yeah makes sense	1	0	0	0	0	0	0
And <CARD> to see if the ' All ' section of the statement is correct	0	0	0	0	1	0	1
<MENTION> any ideas ?	0	0	0	0	0	0	0
but then again most people get this wrong then it cant be as easy as we think surely	1	0	0	0	1	1	0
Probably not	0	0	0	0	0	1	0
So do we think we should flip <CARD> ?	1	0	0	0	0	0	0
then yeah we d have to make sure two vowels or two even numbers appear	0	0	0	0	0	0	1
so i think you d just need to turn over <CARD> and <CARD>	0	0	0	1	0	0	1
Why not <CARD> ?	1	0	0	0	0	0	0
yeah and <CARD> like you said	0	0	0	0	0	1	0
i m happy with that if you guys are	0	0	0	1	1	0	0
I am	0	0	0	0	1	1	0
yeah m happy with that	0	0	0	0	0	0	0
i m *	0	0	0	0	0	0	0
So <CARD> , <CARD> and <CARD> ?	1	0	0	0	1	0	1
<CARD> , <CARD> & <CARD>	0	0	0	0	0	1	1

Dialogue Term Extraction using Transfer Learning and Topological Data Analysis

Renato Vukovic, Michael Heck, Benjamin Ruppik
Carel van Niekerk, Marcus Zibrowius, Milica Gašić
Heinrich Heine University Düsseldorf, Germany
{renato.vukovic, heckmi, ruppik, niekerk, marcus.zibrowius, gasic}@hhu.de

Abstract

Goal oriented dialogue systems were originally designed as a natural language interface to a fixed data-set of entities that users might inquire about, further described by domain, slots and values. As we move towards adaptable dialogue systems where knowledge about domains, slots and values may change, there is an increasing need to automatically extract these terms from raw dialogues or related non-dialogue data on a large scale. In this paper, we take an important step in this direction by exploring different features that can enable systems to discover realizations of domains, slots and values in dialogues in a purely data-driven fashion. The features that we examine stem from word embeddings, language modelling features, as well as topological features of the word embedding space. To examine the utility of each feature set, we train a seed model based on the widely used MultiWOZ data-set. Then, we apply this model to a different corpus, the Schema-Guided Dialogue data-set. Our method outperforms the previously proposed approach that relies solely on word embeddings. We also demonstrate that each of the features is responsible for discovering different kinds of content. We believe our results warrant further research towards ontology induction, and continued harnessing of topological data analysis for dialogue and natural language processing research.

1 Introduction

Dialogue systems are becoming increasingly popular as natural language interfaces to complex services. Goal-oriented dialogue systems, which we see as the main area of application of the results presented here, are intended to be capable of conversing with a user to solve one or more tasks. They need to provide factual information and plan ahead over the course of multiple turns of dialogue. Thus, they differ fundamentally from chat-based dialogue systems, which aim to engage the user in interesting conversation by offering entertainment. Chat-based systems have been successfully trained using fully end-to-end approaches founded on large pre-trained models (Adiwardana et al., 2020; Lin et al., 2020; Zhang et al., 2020; Thoppilan et al., 2022). In contrast, state-of-the-art goal-oriented dialogue systems continue to rely on a pre-defined *ontology*: a database comprising domains (i.e., general topics for interaction), slots (constructs belonging to a particular topic), and values (concrete instantiations of such constructs) (Ultes et al., 2017; Zhu et al., 2020; Kulhánek et al., 2021; Peng et al., 2021; Lee, 2021; He et al., 2022).

Consequently, state-of-the-art goal-oriented dialogue systems still have a high reliance on manual labour. Firstly, the underlying ontology needs to be manually designed for each domain of conversation (Milward and Beveridge, 2003). Secondly, the dialogue system needs to learn from a certain amount of dialogue data labelled with concepts from that ontology in order to recognize and understand these concepts in context (Young et al., 2013). This manual annotation is again challenging, time-consuming and expensive (Budzianowski et al., 2018). There is thus a strong need for methods that can automate *ontology construction* from raw data. Moreover, ontology construction from raw dialogue data would have two-fold benefits: the dialogue data would be labelled automatically

Proceedings of the SIGdial 2022 Conference, pages 564–581
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

as the ontology is constructed, thus rendering any human involvement unnecessary.

In this work, we concentrate exclusively on the first step of ontology construction: *term extraction*. The terms relate to regions of importance in the raw text. The subsequent steps of ontology construction, which we do not consider here, usually involve some form of clustering to boil down the extracted terms to a smaller number of concepts before they are finally organized into a full ontology.

Traditionally, term extraction begins by extracting terms based on frequency, in a way that aims to maximize recall (Nakagawa and Mori, 2002; Wermter and Hahn, 2006). As frequency alone is a fairly primitive feature, this first step has close to zero precision and typically results in far too many terms. This makes further substantial filtering necessary within the term extraction step (Frantzi and Ananiadou, 1999). Filtering typically relies on heuristics or pre-existing natural language processing (NLP) models that have been trained on unrelated data, e.g., semantic parsers (Bourigault and Jacquemin, 1999; Aubin and Hamon, 2006). Heuristics as well as NLP models require substantial amounts of linguistic expertise to be created.

In this work, we take a purely data-driven approach toward dialogue term detection to circumvent these limitations. The high dimensional data spaces arising from word embeddings are hard to understand and visualize. Topological data analysis (TDA) is a collection of mathematical tools which provides measurements of the geometry of high-dimensional point clouds at various scales. The major advantage of topological features is their invariance under small deformations and rotations, as opposed to the coordinates of the embedding vectors. This leads to characteristics that are very generalizable and not dependent on the exact data set used for training. The utility of TDA for NLP and dialogue modelling in particular are still underexplored. We believe that information that can be gathered using topological methods has considerable predictive power concerning term extraction, which to the best of our knowledge we exploit with this work for the very first time.

Starting from the approach of Qiu et al. (2022), we train a BIO-tagging (Ramshaw and Marcus, 1995) model on the widely used MultiWOZ (Budzianowski et al., 2018) data-set as the seed set by fine-tuning general purpose large pre-trained language models. Our BIO-tagger accepts various features as input, all of which uniquely contribute to solving the task. We measure the zero-shot transfer ability of our proposed models on the Schema-Guided Dialogue (Rastogi et al., 2020) data-set, another well-established large-scale corpus for dialogue modelling. Our contributions are as follows:

- We present novel features to solve the term extraction task. Our experimental results show significant improvements over a strong baseline, a recently proposed model that only takes contextual word embeddings as input.
- We demonstrate the suitability of masked language modelling scores to predict relevant terms.
- We exhibit the suitability of a range of topological features of neighbourhoods of word vectors to predict terms of relevance, including terms that are not present in the original seed training set.
- We make our code publicly available.[1]

Our proposed method for term extraction leverages semantics as well as information gained from topological data analysis. No element of our approach requires linguistic knowledge, nor do we rely on any heuristics. Our models are either trained from scratch using a seed data-set, or leverage the predictive power of pre-trained and then fine-tuned large general purpose language models. These models learn via self-supervision on large corpora, and our additional training only requires a moderate amount of labelled seed data.

2 Related Work

It is normally assumed that the *ontology* is provided and built independently of the dialogue system. For instance, in information seeking dialogue systems, this would be a structured representation of the database. Approaches to ontology learning from texts generally involve enriching a small ontology with new concepts and new relationships using text mining methods such as linguistic techniques and lexico-syntactic patterns (Pantel and Pennacchiotti, 2006; Aguado De Cea et al., 2008), clustering techniques (Agirre et al., 2000; Witschel, 2005), statistical techniques (Sugiura et al., 2003) and association rules (Bodenreider et al., 2005; Gulla et al., 2009). The majority of these methods require some form of human intervention. The potential of machine learning in this area has been demonstrated

[1] http://doi.org/10.5281/zenodo.6858565

in the Never-Ending Language Learning (NELL) project (Mitchell et al., 2018). NELL learns factual knowledge from years of self-supervised experience in harvesting the web, using previously learned knowledge to improve subsequent learning.

In the pipeline of knowledge base construction, term extraction is typically the first step. One example of a term extractor is presented in (Sclano and Velardi, 2007). It uses a part-of-speech (POS) tagger to select nouns, verbs and adjectives to which a number of heuristic frequency-based probabilistic models are applied to select term candidates. WordNet (Fellbaum, 1998) is employed to handle misspellings. A number of more recent methods for knowledge base construction start with a similar approach as Sclano and Velardi (2007). In (Romero and Razniewski, 2020) we can also see heavy reliance on frequency, the use of dependency parsers in (Nguyen et al., 2021), as well as rules based on lexical and numerical features and the use of WordNet as in (Chu et al., 2019).

A notable example of dialogue ontology induction is presented in (Hudeček et al., 2021), where a rule-based semantic parser is used as a starting point to propose an initial set of concepts. A more data-driven approach is presented by Qiu et al. (2022) who proposed training a BIO-tagger on fine-tuned contextual embeddings to induce slots. The approach is validated on MultiWOZ via leave-one-out domain experiments. We take this work as a starting point. In very recent work, Yu et al. (2022) propose ontology induction using language modelling attention maps and regularized probabilistic context free grammar to detect regions of interest in text, followed by clustering. This work is complementary to ours, and it would be interesting to explore its combination with our proposal.

The 'Beyond domain APIs' track of the 9th dialog system technology challenge (DSTC9) (Gunasekara et al., 2020) aimed to remove friction in task-oriented dialogue systems where users might issue a request that is out of a system's scope. While DSTC9 aimed to integrate non-dialogue data into dialogue, none of the challenge submissions attempted ontology construction or expansion.

Topological data analysis remains largely underutilized in natural language processing. One notable exception is the work presented by Jakubowski et al. (2020). It shows that the Wasserstein norm of degree zero persistence of punctured neighbourhoods in a static word embedding correlates with the polysemy of a word. Tymochko et al. (2021) apply persistent homology to word embedding point clouds with the goal of distinguishing fraudulent from genuine scientific publications. Their best performing model utilizes persistence features derived from time-delay embeddings of term frequency data. Kushnareva et al. (2021) compute persistent homology of a filtered graph constructed from the attention maps of a pre-trained language model and harness the features for an artificial text detection task.

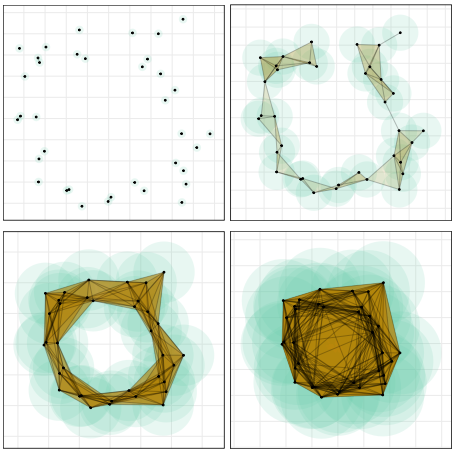
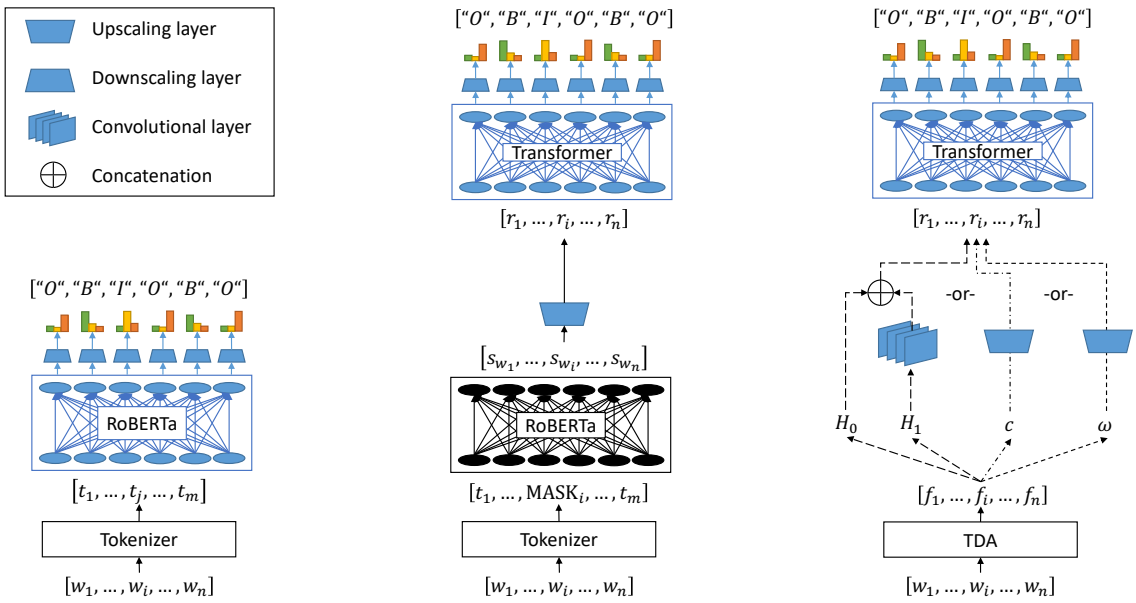


Figure 1: Illustration of the Vietoris-Rips complex VR_ε for four different values of ε.

3 Background on TDA

Topological data analysis (TDA) is an emerging toolkit of mathematical methods for analysing the 'shape' of data. In our case, we study point clouds resulting from word vector embeddings, but these general methods apply equally well to spaces of sensor data, images, or audio. *Topology* measures important features of a geometric space which are invariant under certain structure preserving transformations such as scaling, rotation, stretching and bending. *Homology* quantifies the presence or absence of d-dimensional *holes* in a geometric space: In dimension $d = 0$ the homology group H_0 computes the connected components of a space, while in dimension $d = 1$ the group H_1 describes the non-fillable closed loops in the space.

Consider a discrete point cloud $P \subset \mathbb{R}^M$ equipped with a distance such as the Euclidean metric or the cosine distance. To apply topological tools to P, we need to turn P into a geometric space. One such 'geometrization' is the *Vietoris-Rips complex* VR_ε, which produces, for each non-



(a) Baseline model (subsection 4.1) (b) MLM model (subsection 4.2) (c) TDA models (subsection 4.3)

Figure 2: Our three main architectures for dialogue term detection. Their main distinction is the type of features expected as input. Blue denotes trainable model components. For illustration purposes, here $n = 6$.

negative filtration parameter ε, a *simplicial complex*, a certain higher-dimensional generalization of a graph. To construct VR_ε, we consider a collection of higher-dimensional balls of radius ε centred at the data points. As ε increases, the balls grow and merge as in Figure 1. Their overlaps determine the vertices, edges, triangles and higher-dimensional pieces of the complex VR_ε.

The motivation for varying ε is to measure the 'scale' or 'resolution' of different topological features. The filtration parameters ε at which different k-dimensional holes appear and disappear in VR_ε are summarized in a multiset of points in the plane, visually represented as a *persistence diagram* as in Figure 4. Each dot in the diagram corresponds to a feature. Its horizontal coordinate is the birth time, its vertical coordinate the death time of the feature. The farther a dot is away from the diagonal, the longer the corresponding feature persists across the range of the parameter ε, and thus the more likely it is to reflect a large-scale topological property of the point cloud P. For an overview of persistent homology from a computational perspective, see Edelsbrunner and Harer (2010).

4 Dialogue Term Detection

4.1 Term Tagging

Our ultimate goal is to extract terms describing domains, slots and values from raw dialogues. In order to achieve this, we adopt the BIO-tagging mechanism presented by Qiu et al. (2022). In the seed corpus, the spans where concepts occur are tagged with labels 'B' (beginning of concept), 'I' (inside of concept) and 'O' (outside of concept), without distinguishing between different concepts. The baseline model is trained on RoBERTa (Liu et al., 2019) embeddings as features, and shows modest generalization capabilities when tested in leave-one-out domain experiments.

We investigate two fundamentally different feature sets to increase the generalization capability of models fine-tuned for BIO-tagging. For each feature, we use a specific input projection and train a transformer followed by a token-level classification head. This architecture is illustrated in Figure 2. As the models extract different terms depending on the feature type they are trained on, we use the union of the predictions of all the TDA models, respectively, of all the models, to obtain the final set of terms. One may also build a combined model using all features as joint input, however due to the nature of the training this would maximize accuracy and not recall.

4.2 MLM Model

The first feature set we consider stems from context-level information captured by large pretrained masked language models (MLM)

like BERT (Devlin et al., 2019) and RoBERTa. Our hypothesis is that, based on how confident an MLM is in predicting a certain word, we can infer the meaningfulness of said word. We introduce the *masked language modelling score* (MLM score) $s(w_i) = 1 - p_{\text{MLM}}(w_i \mid [w_1, \ldots, w_{i-1}, \text{MASK}, w_{i+1}, \ldots, w_n])$ as the probability that the word w_i is not predicted by the MLM for the MASK token on position i, based on the context $[w_1, \ldots, w_{i-1}, \text{MASK}, w_{i+1}, \ldots, w_n]$. Thus, meaningful words should have a high MLM score, as illustrated in Appendix C. The total MLM score s_{w_i} of a word is the average of all scores of all appearances of the word in the data-set.

4.3 TDA Models

Topological features allow us to address the following problem observed in transfer learning: Tagging models trained directly on the word embedding vectors derived from one dialogue data-set do not generalize well to the embeddings of a different data-set. We propose the investigation of *topological features* of neighbourhoods of word vectors. Such topological features capture geometric properties that are invariant under distance-preserving transformations of the data points, and are more generalizable and stable under perturbation than the word-vectors and language model features themselves. Our hypothesis is that these features reflect properties of words that are data-set independent.

The simplest topological feature we examine is a *codensity vector* that measures the data density in neighbourhoods of various sizes of a given word vector. A second, far more sophisticated feature that we utilize is *persistence*. As explained in section 3, persistence detects geometric features of a data-set at different scales. While degree zero persistence is closely related to density measures, higher degree persistence captures more refined information. Finally, we also investigate the *Wasserstein norm* as a two-dimensional summary of persistence. We now describe these topological descriptors with more mathematical rigour.

Word embedding neighbourhoods Our topological tagging models use descriptors derived from neighbourhoods of words in the embedding space. The neighbourhoods are defined relative to a point cloud $X \subset \mathbb{R}^M$ constructed from the word vectors of an ambient vocabulary embedding. For a given centre $w \in \mathbb{R}^M$, let $\mathcal{N}_n(w) \subset X \cup \{w\}$ denote the subset of the n nearest neighbours of w with respect to cosine distance (including w itself). In our experiments, we employ neighbourhoods of size $n = 50$. See Figure 3 or Appendix A for examples.

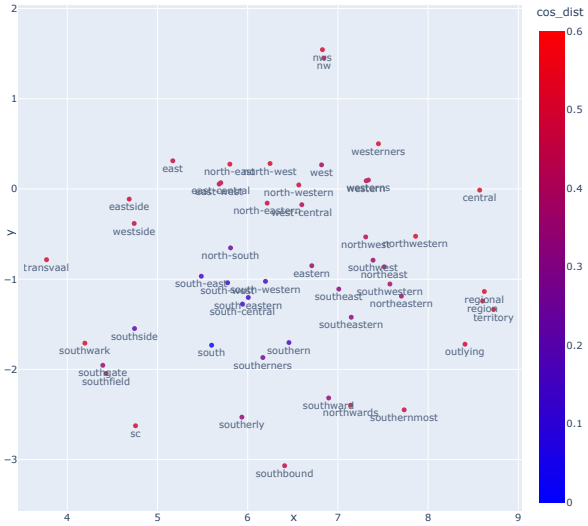


Figure 3: 2-dim. t-SNE projection of the neighbourhood $\mathcal{N}_{50}(w)$ of $w =$ 'south'. Colours indicate cosine distance from w in the original 384-dim. embedding space.

The ambient point cloud X in the embedding space needs to be independent of the specific dialogue vocabulary, so that the resulting persistence features of the neighbourhoods remain comparable. Our vocabulary consists of the 50,000 most common words in the English language, extracted from Grave et al. (2018). The embeddings for the point cloud X are created from the SentenceTransformers (Reimers and Gurevych, 2019) `paraphrase-MiniLM-L6-v2` model. These dense embeddings of dimension $M = 384$ can be meaningfully compared with cosine similarity. Note that even though we are using a contextualized model for creating the embeddings, we obtain a 'static' point cloud X containing the 50,000 vocabulary vectors. For building the neighbourhood $\mathcal{N}_n(w)$ of a word w not contained in the ambient vocabulary X, we first produce w's SentenceTransformers embedding.

Codensity The *k-codensity* in a point w of a point cloud $P \subset \mathbb{R}^M$ is defined as the distance from w to the kth nearest point in P. Thus, points with many neighbours at a close distance have a small codensity, which corresponds to a large density of the point cloud around the point w. We construct a 6-dimensional vector c containing the k-codensity of $\mathcal{N}_{50}(w)$ at w for $k \in \{1, 2, 5, 10, 20, 40\}$, with the intention of quantify-

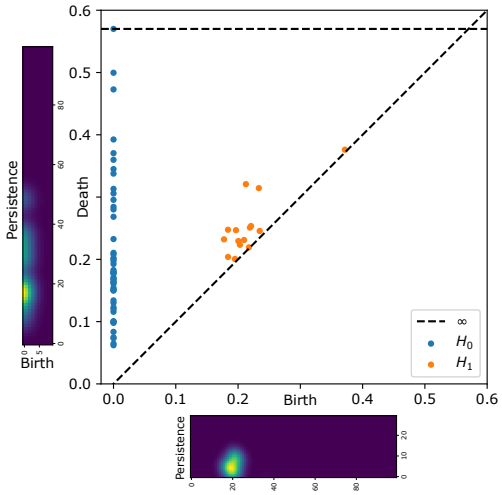


Figure 4: Persistence diagram of $\mathcal{N}_{50}(w = \text{'south'})$ for H_0 (blue dots) and H_1 (orange dots) and corresponding persistence images (left: H_0, bottom: H_1).

ing the neighbourhood density at various scales.

Persistence We produce the persistence diagram (PD) of the sub-point-cloud $\mathcal{N}_{n=50}(w) \subset \mathbb{R}^{384}$ with filtration parameter in the range $[0, 1]$ using cosine distances. Practically, we apply Ripser (Bauer, 2021) and its Python interface (Tralie et al., 2018) for computations of H_0 and H_1 with $\mathbb{F}_2$-coefficients. We restrict to 0- and 1-dimensional homology to keep the computational costs reasonable. The resulting persistence diagram is a multiset of points in the unit square $[0, 1]^2$, as in Figure 4.

Before we can pass the persistence diagrams into the tagging model, we have to apply a *vectorization* step, i.e., map the persistence diagrams into a space which is suitable for training machine learning classifiers. For this we use *persistence images* (Adams et al., 2017), a short overview of the construction and our choice of parameters is given in Appendix B. Figure 4 contains an example of the persistence images for the 'south' neighbourhood.

Wasserstein norm The *Wasserstein distance* is a commonly applied measure of similarity of persistence diagrams (Cohen-Steiner et al., 2010). In our case, it is a rough numerical estimate of the similarity of the shapes of neighbourhoods. The *Wasserstein norm* $\|D\|$ is the Wasserstein distance from D to the empty diagram. For constructing the input features of the Wasserstein models, we compute the order-1 Wasserstein distances with Euclidean ground metric using the GUDHI library (The GUDHI Project, 2022) separately for the H_0 and H_1 persistence diagrams, leading to a 2-dimensional Wasserstein input vector ω.

4.4 Training & Inference

The MLM score model (Figure 2b) and the TDA models (Figure 2c) use the following input projections of the respective input features: The 100-dimensional H_0 persistence image vector and 30×100-dimensional H_1 persistence image are passed into the model independently and concatenated after downscaling H_1 to dimension 396 via a convolutional layer with kernel size 35×25. Then they are input to a transformer with hidden dimension $h = 496$ and 8 attention heads. The transformer output is the input for a token-level classification head after passing through a dropout layer. The 6-dimensional codensity vector c, the 2-dimensional Wasserstein norm vector ω and the single-dimensional MLM score s are all upscaled to hidden dimension $h = 128$ via a 2-layer fully connected neural network to expand the representation space, before being put into three separate transformers with hidden dimension $h = 128$ and 16 attention heads. The transformer sequence output passes through a dropout layer into the token-level classification head. The token-level classification head consists of a dropout layer, a feed-forward layer with hidden dimension h, another dropout, $\tanh$ for activation and an output projection to dimension 3 corresponding to the three possible BIO tags. The classification head is based on the implementation in the HuggingFace library (Wolf et al., 2019), where the dropout rate for all layers is 0.1.

We utilize RoBERTa encoders in two of our models (see Figure 2), once to obtain MLM scores with fixed parameters, and once to obtain contextual semantic embeddings after fine-tuning on the BIO-tagging task. We train each model on MultiWOZ with cross-entropy loss and a learning rate of $4\text{e}{-5}$ using the AdamW optimizer (Loshchilov and Hutter, 2019), warm-up for 10% of total training steps and linear decay afterwards. We train for 15 epochs, with training stopping early if the loss on the validation set stays within a range of $\delta = 0.005$ and batch size 128 on one NVIDIA Tesla T4 GPU. For the much smaller training data in the leave-one-out experiments, the batch size is decreased to 32.

5 Experiments

We conducted experiments to answer the following questions: (1) Is it possible to train a model on

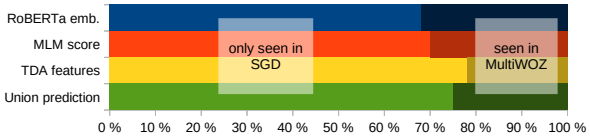


Figure 5: Percentage of extracted terms which were already seen during training or are only seen on SGD during test time.

the seed data-set that achieves a high recall rate on the unseen ontology? (2) Which of the proposed features is most valuable for that purpose? (3) What kind of concepts is the model able to find?

Note that we are mainly focusing on recall as evaluation measure, while retaining the F1-score of the baseline model. Improvements in precision can be achieved with further post-processing, such as clustering (Qiu et al., 2022; Yu et al., 2022).

5.1 Data-sets

We use two well-established data-sets for modelling task-oriented dialogues. MultiWOZ (Budzianowski et al., 2018; Eric et al., 2020) is a corpus of human-to-human dialogues that were collected in a Wizard-of-Oz fashion. Each conversation has one or more goals that revolve around seeking information about or booking tourism-related entities. The data-set consists of over 10,000 dialogues covering 6 domains. There are 30 unique domain-slot pairs that take approximately 4,500 unique values. Value occurrences are annotated with span labels. MultiWOZ is the seed set for training all of our term extraction models.

The Schema-Guided Dialogue (SGD) data-set (Rastogi et al., 2020) is considerably larger than MultiWOZ, with dialogues spanning across 20 domains that represent a wide variety of services. The number of unique values is almost four times larger than in MultiWOZ. This means that any model trained on the significantly more narrow MultiWOZ seed data would need to be able to generalize extremely well to achieve reasonable term extraction performance on SGD. Therefore, SGD is an ideal data-set for our zero-shot experiments.

5.2 Set-up

In order to investigate the models' ability to extract terms in an unseen domain, we design two experiments. First, we conduct a leave-one-out domain experiment on MultiWOZ, similar to the approach taken by Qiu et al. (2022), with two important differences. We focus mainly on recall as

Approach	Measure	Taxi	Rest.	Hotel	Attr.	Train
RoBERTa embeddings	F1	0.87	0.81	0.68	0.91	0.84
	Recall	0.87	0.89	0.95	0.94	0.92
	Precision	0.87	0.76	0.53	0.89	0.77
MLM score	F1	0.44	0.47	0.32	0.42	0.57
	Recall	0.43	0.48	0.69	0.53	0.72
	Precision	0.46	0.46	0.21	0.35	0.47
Persistence image vectors	F1	0.72	0.61	0.41	0.63	0.65
	Recall	0.79	0.69	0.87	0.65	0.92
	Precision	0.67	0.54	0.27	0.61	0.50
Codensity	F1	0.57	0.46	0.38	0.51	0.62
	Recall	0.51	0.48	0.64	0.59	0.76
	Precision	0.64	0.44	0.27	0.45	0.52
Wasserstein norm	F1	0.57	0.50	0.45	0.46	0.48
	Recall	0.58	0.53	0.46	0.51	0.69
	Precision	0.57	0.47	0.45	0.43	0.37
TDA features	F1	0.65	0.53	0.33	0.52	0.47
	Recall	0.84	0.81	0.89	0.84	0.94
	Precision	0.53	0.39	0.20	0.37	0.31
Union prediction	F1	0.65	0.53	0.26	0.49	0.44
	Recall	**0.95**	**0.92**	**0.97**	**0.98**	**0.98**
	Precision	0.50	0.37	0.15	0.33	0.28

Table 1: Leave-one-out results on MultiWOZ.

the adequate evaluation measure for term extraction, and we do not allow partial matches of the tagged term. When designing the matching function, we were guided by the tolerance threshold of a picklist-based dialogue state tracker. For example, the term extractor is allowed to match 'an expensive' with the golden term 'expensive', as having a non-content word in the term would make no difference to the tracker. However, matching 'Pizza Hut' with the golden term 'Pizza Hut Cherry Hinton' is considered a false positive, as 'Pizza Hut' would not be precise enough for the tracker to distinguish entities. Note that such matches were considered by Qiu et al. (2022) as true positives, so our matching function is stricter. For both training and testing we limit ourselves to user utterances, as the system utterances may contain API calls, which is already structured data.

For the second experiment, we train our models on the training portion of the MultiWOZ data-set and test it on the SGD data-set. We then examine the overlap in true positives between models using different features. We also analyse the models' abilities to extract terms referring to different domains and slots, highlighting easy and difficult terms.

5.3 Results

Leave-one-out domain We remove one of the five MultiWOZ domains in training and only test

Approach	F1 ↑	Rec. ↑	Prec. ↑	L2 ↓	Tags
RoBERTa emb.	0.45	0.35	0.63	0.29	2757
MLM score	0.34	0.34	0.35	0.35	4933
PI vectors	0.47	0.46	0.48	**0.20**	4775
Codensity	0.37	0.34	0.42	0.52	4054
Wasserst. n.	0.42	0.40	0.44	0.62	4536
TDA features	0.48	0.63	0.39	-	8189
Union pred.	**0.48**	**0.74**	0.36	-	10398

Table 2: Dialogue term extraction results on SGD with models trained on MultiWOZ together with the total number of tagged terms per model. There are 5008 target terms in SGD. L2-norm is used as uncertainty measure for the single models.

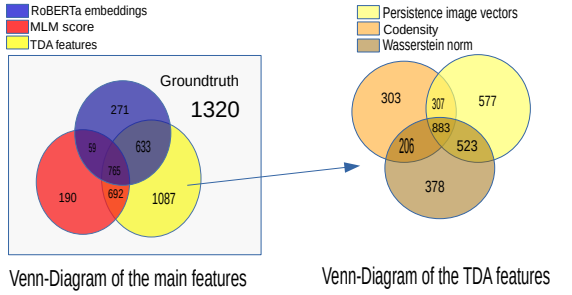


Figure 6: Venn-Diagram of SGD terms found in each of the three models using RoBERTa, MLM score, TDA features, as well as analysis of term overlap of the models trained on different TDA features.

on it, so the model has not seen any dialogues in the left-out domain. We only utilize single domain dialogues in the training and test set. Results in Table 1 show that the recall increases for each unseen domain experiment when adding the predictions by the models trained on persistence and language modelling features to form the union prediction.

Unseen ontology The results in Table 2 show that adding the predictions of the new feature models improve both recall and F1-score significantly for term extraction on the unseen SGD ontology compared to the language model only baseline, without the need to fine-tune the embeddings on the token classification task with any SGD data. In Figure 5 the percentage of completely new terms found in the predictions of each model is shown. The TDA feature model predictions contain mostly unseen terms. Confidence scores would be critical in a subsequent automatic ontology construction. We compare the L2-norm of the model's predictions to the ground truth label, showing that the model trained on persistence image vectors from MultiWOZ has the highest confidence score on the unseen SGD data.

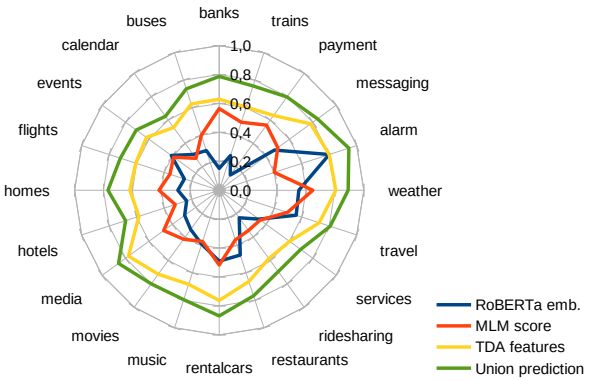


Figure 7: Recall per domain on SGD by our models compared with the baseline fine-tuned RoBERTa model.

Overlap Figure 6 shows that the sets of extracted terms differ significantly by model. Therefore, the union of predictions is useful for capturing as many relevant terms as possible. The MLM score model already adds more terms to the fine-tuned language model. The topological features, however, by far supply the biggest portion of new terms. Among the different TDA features, the persistence images yield the largest number of additional terms.

Domain and slot coverage Figure 7 demonstrates that the different models find various amounts of terms depending on the domain. The recall of the TDA models is the highest across all domains, while RoBERTa is only able to outperform the MLM score model in terms of recall in 5 out of 20 domains, e.g., in 'music' and 'restaurants', which contain many multi-word terms.

Examples False negatives tend to be long multi-word terms, as exemplified in Table 3. False positives predominantly include typos and incomplete terms. Predictions by RoBERTa contain 2.0 words on average. In contrast, the MLM score model and TDA feature model term predictions have an average length of 1.6 and 1.8 words, respectively. We give an illustrative instance of terms extracted by the different models from an example utterance in Table 4.

6 Discussion and Future Outlook

Our novel term extraction approach based on topological data analysis and masked language modelling scores significantly outperforms the word-embedding-based baseline on the recall rate both in leave-one-out experiments and when applied to a completely different corpus. Importantly, our re-

Seen in MultiWOZ	Only seen in SGD	False negatives	False positives
Lebanese; Hotel Indigo London-Paddington; LAX International Airport; The Queen's Gate Hotel; Hair salon	Delta Aesthetics; McDonald's; 3455 Homestead Road; receiver; Pescatore	Little Hong Kong; Yankees vs. Rangers; Dr. Eugene H. Burton III; 341 7th street; La Quinta Inn by Wyndham Dacramento Downtown	Especillay by; Bears vs; Angeles and; Polk Street; theater please; resrevation; neaarby

Table 3: Example predictions of the Union model on SGD (typos are reproduced as they appear in the data-set). Examples for each of our other models can be found in Appendix E.

	i ' d like to find a steakhouse that ' s not very costly to eat at .
RoBERTa embeddings	steakhouse not
MLM score	' steakhouse that
TDA features	steakhouse costly

Table 4: Example of a normalized, tokenized utterance together with terms extracted by the different models. Unconnected boxes indicate separate terms, i.e., here the MLM score model assigned a B tag to 'steakhouse' and a B tag to 'that'. More example utterances can be found in Appendix E.

sults demonstrate a strong ability of topological data analysis to extract domain independent features that can be used to analyse unseen data-sets. This finding warrants further investigation.

Our approach still produces a significant number of false positives. The next step in the ontology construction pipeline, clustering, could be deployed to significantly reduce that number, as has already been demonstrated by Yu et al. (2022). We believe that their approach and our approach could be combined, but that goes beyond the scope of this work.

However, ultimately, precision is only of secondary importance. In a typical goal oriented system, we have a dialogue state tracker tracking concepts through conversation. Whether or not the tracker is tracking some irrelevant terms does not impact the overall performance of a dialogue system. All that matters is that the tracker does track every term that actually is a concept. Of course, the computational complexity of the tracker increases linearly with the number of tracked terms (Heck et al., 2020; van Niekerk et al., 2020; Lee et al., 2021). But, as can be seen from Table 2, our method merely doubles the number of terms, so the computational price tag is low. With this in mind, it is also conceivable that the tracker itself could be utilized to increase the precision. This would be an interesting direction for further research.

Some simpler options for improvement are more immediate: Here, we utilize SentenceTransformers only to provide static embeddings for each word, but of course a similar analysis can be applied to contextualized word embeddings, at the expense of higher computational complexity. Further, persistence images (subsection 4.3) could be replaced by features tailored to downstream tasks, such as features obtained from the novel Persformer model (Reinauer et al., 2021).

7 Conclusion

To the best of our knowledge, we present the first application of topological features in dialogue term extraction. Our results show that these features distinguish content from non-content words, in a way that can be generalized from a training domain to unseen domains. We believe that these findings are only the tip of the iceberg, and warrant further investigation of topological features in NLP in general. In addition, we have shown that masked language modelling scores are useful for term extraction as well. In combination, the features we investigate allow us to make a significant step towards automatic ontology construction from raw data.

Acknowledgements

RV is supported by funds from the European Research Council (ERC) provided under the Horizon 2020 research and innovation programme (Grant agreement No. STG2018 804636) as part of the DYMO project. CVN and MH are supported by funding provided by the Alexander von Humboldt Foundation in the framework of the Sofja Kovalevskaja Award endowed by the Federal Ministry of Education and Research. Google Cloud provided computational infrastructure. We want to thank the anonymous reviewers whose comments improved the exposition of our paper.

References

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. 2017. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35.

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a Human-like Open-Domain Chatbot. *arXiv:2001.09977*.

Eneko Agirre, Olatz Ansa, Eduard Hovy, and David Martínez. 2000. Enriching very large ontologies using the WWW. In *Proceedings of the First International Conference on Ontology Learning*, volume 31, pages 25–30.

Guadalupe Aguado De Cea, Asunción Gómez-Pérez, Elena Montiel-Ponsoda, and Mari Carmen Suárez-Figueroa. 2008. Natural language-based approach for helping in the reuse of ontology design patterns. In *Proceedings of the International Conference on Knowledge Engineering (ICKE)*.

Sophie Aubin and Thierry Hamon. 2006. Improving term extraction with terminological resources. In *Advances in Natural Language Processing*, pages 380–387, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ulrich Bauer. 2021. Ripser: Efficient computation of Vietoris-Rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423.

Olivier Bodenreider, Marc Aubry, and Anita Burgun. 2005. Non-lexical approaches to identifying associative relations in the gene ontology. *Pacific Symposium on Biocomputing*, 10(91):102.

Didier Bourigault and Christian Jacquemin. 1999. Term extraction + term clustering: An integrated platform for computer-aided terminology. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 15–22, Bergen, Norway. Association for Computational Linguistics.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Cuong Xuan Chu, Simon Razniewski, and Gerhard Weikum. 2019. TiFi: Taxonomy induction for fictional domains. In *The World Wide Web Conference*, WWW '19, page 2673–2679, New York, NY, USA. Association for Computing Machinery.

David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. 2010. Lipschitz functions have L_p-stable persistence. *Foundations of Computational Mathematics. The Journal of the Society for the Foundations of Computational Mathematics*, 10(2):127–139.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Herbert Edelsbrunner and John L. Harer. 2010. *Computational topology, An introduction*. American Mathematical Society, Providence, RI.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tür. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC)*, pages 422–428, Marseille, France. European Language Resources Association.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA.

Katerina T Frantzi and Sophia Ananiadou. 1999. The C-value/NC-value domain-independent method for multi-word term extraction. *Journal of Natural Language Processing*, 6(3):145–179.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jon Atle Gulla, Terje Brasethvik, and Gøran Sveia Kvarv. 2009. Association rules and cosine similarities in ontology relationship learning. In *Enterprise Information Systems*, pages 201–212, Berlin, Heidelberg. Springer Berlin Heidelberg.

R. Chulaka Gunasekara, Seokhwan Kim, Luis Fernando D'Haro, Abhinav Rastogi, Yun-Nung Chen, Mihail Eric, Behnam Hedayatnia, Karthik Gopalakrishnan, Yang Liu, Chao-Wei Huang, Dilek Hakkani-Tür, Jinchao Li, Qi Zhu, Lingxiao Luo, Lars Liden, Kaili Huang, Shahin Shayandeh, Runze Liang, Baolin Peng, Zheng Zhang, Swadheen Shukla, Minlie Huang, Jianfeng Gao, Shikib Mehri, Yulan Feng, Carla Gordon, Seyed Hossein Alavi, David R. Traum, Maxine Eskénazi, Ahmad Beirami, Eunjoon Cho, Paul A. Crook, Ankita De, Alborz Geramifard, Satwik Kottur, Seungwhan Moon, Shivani Poddar,

and Rajen Subba. 2020. Overview of the ninth dialog system technology challenge: DSTC9. *CoRR*, abs/2011.06486.

Wanwei He, Yinpei Dai, Yinhe Zheng, Yuchuan Wu, Zheng Cao, Dermot Liu, Peng Jiang, Min Yang, Fei Huang, Luo Si, et al. 2022. Galaxy: A generative pre-trained model for task-oriented dialog with semi-supervised learning and explicit policy injection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36.10, pages 10749–10757.

Michael Heck, Carel van Niekerk, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Marco Moresi, and Milica Gašić. 2020. TripPy: A Triple Copy Strategy for Value Independent Neural Dialog State Tracking. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 35–44. Association for Computational Linguistics.

Vojtěch Hudeček, Ondřej Dušek, and Zhou Yu. 2021. Discovering dialogue slots with weak supervision. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2430–2442, Online. Association for Computational Linguistics.

Alexander Jakubowski, Milica Gašić, and Marcus Zibrowius. 2020. Topology of word embeddings: Singularities reflect polysemy. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 103–113, Barcelona, Spain (Online). Association for Computational Linguistics.

Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. 2021. AuGPT: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 198–210, Online. Association for Computational Linguistics.

Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. Artificial text detection via examining the topology of attention maps. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 635–649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chia-Hsuan Lee, Hao Cheng, and Mari Ostendorf. 2021. Dialogue State Tracking with a Language Model using Schema-Driven Prompting. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4937–4949, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yohan Lee. 2021. Improving end-to-end task-oriented dialog system with a simple auxiliary task. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1296–1303, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaojiang Lin, Peng Xu, Genta Indra Winata, Farhad Bin Siddique, Zihan Liu, Jamin Shin, and Pascale Fung. 2020. Caire: An end-to-end empathetic chatbot. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13622–13623.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692v1.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

David Milward and Martin Beveridge. 2003. Ontology-based dialogue systems. In *Proceedings of the 3rd Workshop on Knowledge and reasoning in practical dialogue systems (IJCAI)*, pages 9–18. Citeseer.

T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM*, 61(5):103–115.

Hiroshi Nakagawa and Tatsunori Mori. 2002. A simple but powerful automatic term extraction method. In *COLING-02: COMPUTERM 2002: Second International Workshop on Computational Terminology*.

Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2021. Advanced semantics for commonsense knowledge extraction. In *Proceedings of the Web Conference 2021*, WWW '21, page 2636–2647, New York, NY, USA. Association for Computing Machinery.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 113–120, Sydney, Australia. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2021. Soloist: Building task bots at scale with transfer learning and machine teaching. *Transactions of the Association for Computational Linguistics*, 9:807–824.

Liang Qiu, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. Structure extraction in task-oriented dialogues with slot clustering. *arXiv:2203.00073*.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards Scalable Multi-domain Conversational Agents: The Schema-Guided Dialogue Dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34.05, pages 8689–8696.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Raphael Reinauer, Matteo Caorsi, and Nicolas Berkouk. 2021. Persformer: A transformer architecture for topological machine learning. *CoRR*, abs/2112.15210.

Julien Romero and Simon Razniewski. 2020. Inside Quasimodo: Exploring construction and usage of commonsense knowledge. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3445–3448, New York, NY, USA. Association for Computing Machinery.

Nathaniel Saul and Chris Tralie. 2019. Scikit-TDA: Topological data analysis for python.

Francesco Sclano and Paola Velardi. 2007. TermExtractor: A web application to learn the shared terminology of emergent web communities. In *Enterprise Interoperability II*, pages 287–290, London. Springer London.

Naoki Sugiura, Masaki Kurematsu, Naoki Fukuta, Noriaki Izumi, and Takahira Yamaguchi. 2003. A domain ontology engineering tool with general ontologies and text corpus. In *Proceedings of the 2nd Workshop on Evaluation of Ontology based Tools (EON)*, volume 87.

The GUDHI Project. 2022. *GUDHI User and Reference Manual*, 3.5.0 edition. GUDHI Editorial Board.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. 2022. LaMDA: Language Models for Dialog Applications. *arXiv:2201.08239*.

Christopher Tralie, Nathaniel Saul, and Rann Bar-On. 2018. Ripser.py: A lean persistent homology library for python. *The Journal of Open Source Software*, 3(29):925.

Sarah Tymochko, Julien Chaput, Timothy Doster, Emilie Purvine, Jackson Warley, and Tegan Emerson. 2021. Con connections: Detecting fraud from abstracts using topological data analysis. In *Proceedings of the 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 403–408.

Stefan Ultes, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gašić, and Steve Young. 2017. PyDial: A multi-domain statistical dialogue system toolkit. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, Vancouver, Canada. Association for Computational Linguistics.

Carel van Niekerk, Michael Heck, Christian Geishauser, Hsien-chin Lin, Nurul Lubis, Marco Moresi, and Milica Gašić. 2020. Knowing What You Know: Calibrating Dialogue Belief State Distributions via Ensembles. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3096–3102, Online. Association for Computational Linguistics.

Joachim Wermter and Udo Hahn. 2006. You can't beat frequency (unless you use linguistic knowledge) – a qualitative evaluation of association measures for collocation and term extraction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 785–792, Sydney, Australia. Association for Computational Linguistics.

Hans F. Witschel. 2005. Using decision trees and text mining techniques for extending taxonomies. In *Proceedings of the Workshop on Learning and Extending Lexical Ontologies by using Machine Learning (OntoML)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. POMDP-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Dian Yu, Mingqiu Wang, Yuan Cao, Izhak Shafran, Laurent El Shafey, and Hagen Soltau. 2022. Unsupervised slot schema induction for task-oriented dialog. *arXiv:2205.04515*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. ConvLab-2: An open-source toolkit for building, evaluating, and diagnosing dialogue systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149, Online. Association for Computational Linguistics.

A Neighbourhoods and Persistence Diagrams

We produce a table with various figures of neighbourhoods, their persistence diagrams, Wasserstein norm vectors and codensity vectors in Figure 8.

B Details about the Persistence Diagram Vectorization Step

We used the scikit-tda/persim library (Saul and Tralie, 2019) in the practical implementation of persistence images.

As a first step, the $(\text{birth}, \text{death})$ coordinates of the dots in the persistence diagram are transformed into $(\text{birth}, \text{lifetime} = \text{death} - \text{birth})$ coordinates. We then place a Gaussian kernel with variance $\sigma = 0.0007$ onto each point in the $(\text{birth}, \text{lifetime})$ diagram, linearly weighted by the lifetime. We sum up the various probability distributions and then integrate the resulting function over the patches of a rasterization with a pixel size of 0.1 of the image plane. Adams et al. (2017) discuss that the performance of the resulting persistence images for downstream tasks is robust in the choices of these parameters. As usual in the Vietoris-Rips filtration, the birth of all the 0-dimensional homology classes in H_0 occur for radius $\varepsilon = 0$, and we consider the persistence features in the range $[0.0, 1.0]$. Thus, we only pass the 0th column of the generated H_0 persistence image to the model, which is a 100-dimensional vector. For the H_1 persistence image, we take the entire birth range $[0.0, 1.0]$ and persistence range $[0.0, 0.3]$ into account, so that the image has dimensions 100×30.

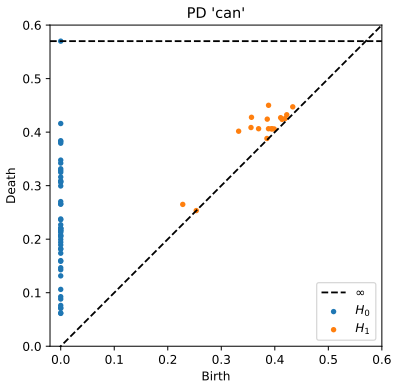
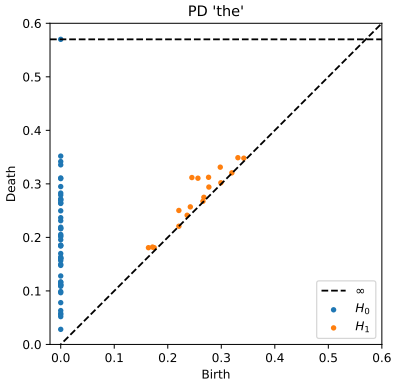
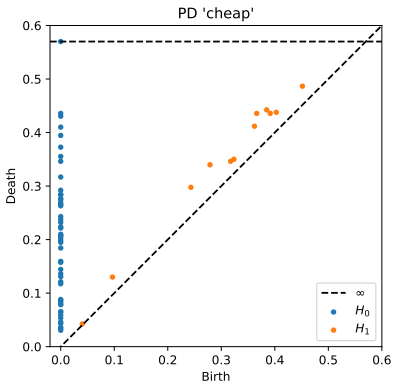
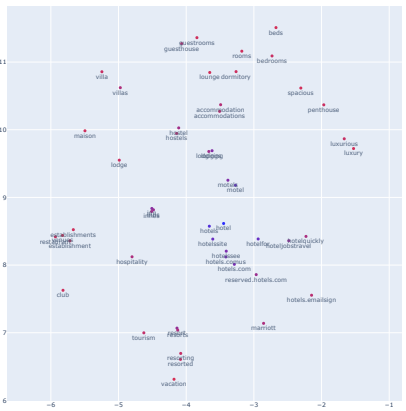
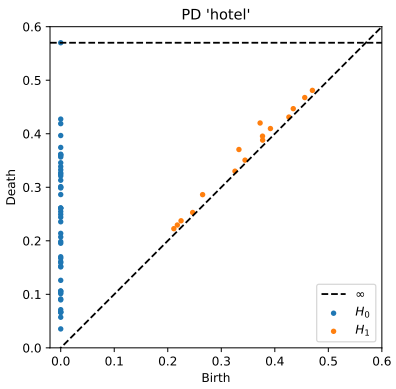
	Neighbourhood	Persistence Diagram	Wasserst. / Codens.
can			$\begin{pmatrix}7.712\\0.311\end{pmatrix}$ $\begin{pmatrix}0.181\\0.259\\0.307\\0.379\\0.402\\0.439\end{pmatrix}$
the			$\begin{pmatrix}6.490\\0.219\end{pmatrix}$ $\begin{pmatrix}0.099\\0.129\\0.213\\0.275\\0.318\\0.349\end{pmatrix}$
cheap			$\begin{pmatrix}6.823\\0.350\end{pmatrix}$ $\begin{pmatrix}0.059\\0.065\\0.195\\0.322\\0.387\\0.464\end{pmatrix}$
hotel			$\begin{pmatrix}7.902\\0.166\end{pmatrix}$ $\begin{pmatrix}0.106\\0.167\\0.217\\0.265\\0.374\\0.473\end{pmatrix}$

Figure 8: 2-dimensional t-SNE projection of the neighbourhood $\mathcal{N}_{50}(w)$; corresponding Persistence diagram; 2-dim. Wasserstein norm vector (for H_0 and H_1); 6-dim. codensity vector (for $k \in \{1, 2, 5, 10, 20, 40\}$).

C Masked Language Modelling Score Examples

In Table 5 the MLM scores on MultiWOZ and SGD of example words show that the score is high for meaningful words across data-sets.

Word	Score on MultiWOZ	Score on SGD
cheap	0.96	0.92
restaurant	0.86	0.86
the	0.59	0.63
how	0.70	0.67
not	0.45	0.50

Table 5: Masked language modelling score examples.

D Further Experimental Results

See Table 6, Table 7, Table 8 and Table 9 for further experimental results.

E Further Example Tags

See Table 10 for more utterances with the corresponding tags by the different models and Table 11 for an analysis of which terms tagged by each model were already seen in MultiWOZ.

	MultiWOZ			SGD		
Approach	F1-Score	Recall	Precision	F1-Score	Recall	Precision
RoBERTa embeddings	0.80	0.91	0.72	0.45	0.35	0.63
MLM scores	0.38	0.83	0.25	0.34	0.34	0.35
Persistence image vectors	0.53	0.87	0.38	0.47	0.46	0.48
Codensity	0.42	0.76	0.29	0.37	0.34	0.42
Wasserstein norm	0.37	0.65	0.26	0.42	0.40	0.44
TDA features together	0.33	0.89	0.20	0.48	0.63	0.39
Union prediction	0.28	**0.96**	0.17	0.48	**0.74**	0.36

Table 6: Results of all models trained on MultiWOZ and tested on MultiWOZ and SGD.

Approach	MultiWOZ	SGD
RoBERTa embeddings	816	2757
MLM score	2174	4933
Persistence image vectors	1464	4775
Codensity	1658	4054
Wasserstein norm	1631	4536
TDA features	2867	8189
Union prediction	3712	10398

Table 7: Total number of terms tagged on MultiWOZ and SGD broken down per model trained on MultiWOZ. For reference, there are 645 target terms in total in MultiWOZ and 5008 in SGD.

	MultiWOZ			SGD		
Approach	F1-Score	Recall	Precision	F1-Score	Recall	Precision
RoBERTa embeddings	0.65	0.92	0.50	0.83	0.88	0.78
MLM scores	0.32	0.76	0.21	0.37	0.33	0.44
Persistence image vectors	0.45	0.84	0.31	0.76	0.80	0.73
Codensity	0.37	0.69	0.25	0.50	0.49	0.51
Wasserstein norm	0.40	0.78	0.27	0.53	0.54	0.52
TDA features together	0.30	0.92	0.18	0.64	0.88	0.50
Union prediction	0.23	**0.98**	0.13	0.61	**0.98**	0.44

Table 8: Results of all models trained on SGD and tested on MultiWOZ and SGD.

Approach	F1-Score	Recall	Precision
RoBERTa embeddings	0.87	0.91	0.84
MLM scores	0.53	0.76	0.41
Persistence image vectors	0.75	0.87	0.66
Codensity	0.59	0.70	0.52
Wasserstein norm	0.53	0.62	0.46
TDA features together	0.57	0.92	0.41
Union prediction	0.50	**0.97**	0.33

Table 9: Results of all models trained on MultiWOZ and tested on the MultiWOZ test set only.

utterance	the curse of la llorona is a good one
RoBERTa embeddings	
MLM score	la good one
Persistence image vectors	la llorona
Codensity	la
Wasserstein norm	
utterance	i ’ m bored . get me some tickets for an activity .
RoBERTa embeddings	
MLM score	
Persistence image vectors	activity
Codensity	
Wasserstein norm	
utterance	what other therapists are there ?
RoBERTa embeddings	
MLM score	
Persistence image vectors	
Codensity	therapists
Wasserstein norm	
utterance	later on . for now i want to know the weather in there next wednesday .
RoBERTa embeddings	wednesday
MLM score	. i wednesday
Persistence image vectors	wednesday
Codensity	weather wednesday
Wasserstein norm	
utterance	do you know a place where i can get some food ?
RoBERTa embeddings	place food
MLM score	food
Persistence image vectors	place food
Codensity	place food
Wasserstein norm	food
utterance	what time does the show begin ?
RoBERTa embeddings	time
MLM score	show
Persistence image vectors	time
Codensity	time
Wasserstein norm	time

Table 10: More examples of tokenized utterances together with terms extracted by the different models.

Model	Seen in MultiWOZ	Only seen in SGD	False negatives	False positives
RoBERTa emb.	Sushi Yoshizumi; Salesforce transit center; Jojo Restaurant & Sushi Bar; bistro liaison; Eric's Restaurant; K&L Bistro	Arizona vs. LA Dodgers; El Hombre; Arcadia; 795 El Camino Real; Owls vs. Tigers; Green Chile Kitchen;	visit date; unapologetic; 134; The Motans; JT Leroy; Orchids Thai; 251 Llewellyn Avenue; 12221 San Pablo Avenue; Menara Kuala Lumpur;	Meriton; Rodeway Inn; Stewart; Embarcadero Center; Elysees; Shattuck; LAX; El; attractionin
MLM score	350 Park Street; Doubletree by Hilton Hotel San Pedro - Port of Los Angeles; 24; Show Time; Up 2U Thai Eatery; 25; 381 South Van Ness Avenue; Broken English	Olly Murs; Bret Mckenzie; football game: USC vs Utah; stage door; 1012 Oak Grove Avenue; 'Mamma Mia; John R Saunderson; Alderwood Apartments	630 Park Court; Unapologetic; visit date; The Motans; V's Barbershop Campbell; 101 South Front Street #1; 134; Orchids Thai	humid then; others?; rad; wa; outdoor; alright, I; valley; spoke; webster; a song
PI vectors	Trademark Hotel; Dorsett City; London; Center Point Road O'Hare International Airport; Maya Palenque Restaurant; Casa Loma Hotel	Claude de Martino; Nero; Toronto FC vs Crew; Written in Sand; Emmylou Harris; Helen Patricia; Palo Alto Caltrain Station; Jack Carson	Shailesh Premi; Gorgasm; 157; Dad; destination city; serves alcohol 2556 Telegraph Avenue #4; Glory Days; The Park Bistro & Bar; Arcadia Sessions at The Presidio	Maggiano; XD; sexist scum; fir; red chillies; morning instead; capitol; Robin; !!! if so; free
Codensity	Tell me you love me; dentist name; The American Hotel Atlanta Downtown - A Doubletree by Hilton; Dim Sum Club; Le Apple Boutique Hotel KLCC; 555 Center Avenue	Hyatt Place New York/Midtown-South; colder weather; 'Little Mix; Commonwealth; 3630 Balboa Street; Newton Faulkner; directed by; How deep is your love	visit date; Unapologetic; 134; The Motans; V's Barbershop Campbell; 101 South Front Street #1; Orchids Thai; 12221 San Pablo Avenue	and humid; vapour; 5:15; corect; names; flight leaving; collect; tiresome; Marriott
Wasserstein norm	Wence's Restaurant; Miss me more; restaurant reservation; 1118 East Pike Street; El Charro Mexican Food & Cantina; Murray Circle Restaurant	Broderick Roadhouse; Mets vs. Yankees; 226 Edelen Avenue; 1030; 162; Phillies vs. Cubs; 1110; Diamond Platnumz; '2664 Berryessa Road #206; Oliveto	Anaheim Intermodal Center; Sangria; Vacation Inn Phoenix; 1776 First Street; After the Wedding; Mikey Day	loacation; enoteca; salone; balances; overseas; mars; help; Angeles and; 4:15; niles; titale; frmo; Oracle park
TDA features together	1300 University Drive #6; The American Hotel Atlanta Downtown - A Doubletree by Hilton; Millennium Gloucester Hotel London Kensington	4087 Peralta Boulevard; Power; Hyang Giri; Okkervil River; event location; 320; Jordan Smith; Caffe California; Ruth Bader Ginsburg; Neil Marshall; 171; 1599 Sanchez Street	Out of Love; Alderwood Apartments; has garage; 168; GP visit; Catamaran Resort Hotel and Spa; Dodgers vs. Diamondbacks; Showplace Icon Valley Fair; West Side Story	venu; being; replaced; parking; Okland; times; comments; pond; crowd; flick; 1,710; Blacow Road; Kathmandu

Table 11: Prediction examples of the different models on SGD (typos are reproduced as they appear in the data-set).

Evaluating N-best Calibration of Natural Language Understanding for Dialogue Systems

Ranim Khojah and **Alexander Berman**
Dept. of Philosophy, Linguistics
and Theory of Science
University of Gothenburg
guskhojra@student.gu.se,
alexander.berman@gu.se

Staffan Larsson
Dept. of Philosophy, Linguistics
and Theory of Science
University of Gothenburg
and Talkamatic AB
staffan@talkamatic.se

Abstract

A Natural Language Understanding (NLU) component can be used in a dialogue system to perform intent classification, returning an N-best list of hypotheses with corresponding confidence estimates. We perform an in-depth evaluation of 5 NLUs, focusing on confidence estimation. We measure and visualize calibration for the 10 best hypotheses on model level and rank level, and also measure classification performance. The results indicate a trade-off between calibration and performance. In particular, Rasa (with Sklearn classifier) had the best calibration but the lowest performance scores, while Watson Assistant had the best performance but a poor calibration.

1 Introduction

Natural Language Understanding (NLU) is an important component in dialogue systems. One of the typical tasks of NLU is intent classification: given a user utterance, the NLU returns a list of N hypotheses (an N-best list) ranked according to confidence estimates (a real number between 0 and 1). The highest ranking hypothesis is returned by the NLU as the predicted intent. Confidence estimates are also available for lower ranked hypotheses.

In this study, we evaluate confidence estimation in 5 NLU services, namely Watson Assistant, Language Understanding Intelligent Service (LUIS), Snips.ai and Rasa (with two pipelines Rasa-Sklearn and Rasa-DIET). We measure the calibration and the performance of NLUs on rank level (results for a specfic rank) and on model level (aggregated results of all ranks). *Calibration* here refers to the correlation between confidence estimates and accuracies, i.e. how useful the confidence estimate associated with a certain hypothesis is for predicting its accuracy.

To achieve our objectives, we conduct an exploratory case study on the 5 NLUs. We train the NLUs using a subset of a multi-domain dataset proposed by Liu et al. (2021). We measure the calibration of the NLUs on model and rank levels using reliability diagrams and correlation coefficient with respect to instance-level accuracy. We also measure the performance on a model level through accuracy and F1-score.

Our evaluation aims to facilitate NLU service selection and help dialogue system developers adapt their dialogue system to specific NLU services. For example, depending on the degree of calibration in an NLU, contextual or interactive disambiguation (clarification requests) can be an option. If confidence estimates reflect true accuracy, then if two (or more) hypotheses have similar confidence estimates, this may indicate the presence of an ambiguity in the user input (from the perspective of the NLU, i.e., disregarding dialogue context) that needs to be resolved. Conversely, if confidence estimates (especially those for non-top ranks) do not reflect accuracies, then even if the top two (or more) hypotheses have similar estimates, this may not be a reliable indication of ambiguity but rather be due to noise.

Our evaluation scripts are publicly available on GitHub[1] along with the dataset, enabling replication of the study and to ease building on it.

2 Related work

Current NLUs typically use machine-learning on natural-language data (i.e., the user utterances) to extract features (e.g., keywords, word counts and word embeddings) and predict the intent of the user accordingly (Jung, 2019; Shridhar et al., 2019).

NLU services are widely used by dialogue developers and allow them to create and train NLU models for dialogue systems. However, the task of

[1] https://github.com/ranimkhojah/confidence-estimation-benchmark

Proceedings of the SIGdial 2022 Conference, pages 582–594
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

choosing the best NLU service depends on the domain and context of the dialogue system. In prior work, benchmarks and evaluations have been performed to identify the best NLU service in different domains like software engineering (Abdellatif et al., 2021), meteorology (Canonico and De Russis, 2018), question answering (Braun et al., 2017) and others (McTear et al., 2016; Stoyanchev et al., 2016; Kar and Haldar, 2016; Koetter et al., 2019). Generally, these evaluation studies have been conducted to draw the trade-off line between different NLU services in terms of the usability of their user interfaces (Gregori, 2017), technical features (e.g., language and device support) (Koetter et al., 2019) and performance (Braun et al., 2017; Liu et al., 2021).

NLU performance is usually assessed via performance measures (e.g., accuracy, F1-score, etc.) which depend only on the top hypothesis returned by the NLU, and disregarding the associated confidence estimates. For example, an NLU that predicts 3 out of 10 intents incorrectly with high confidence estimates has the same performance as an NLU that predicts 3 out of 10 intents incorrectly with a low confidence estimation.

In earlier work, various methods for visualizing and measuring confidence calibration (the extent to which confidence estimates reflect true likelihoods) have been discussed. For example, Guo et al. (2017) and Vasudevan et al. (2019) visualize calibration of neural network models through reliability diagrams. As for quantitative metrics, one proposed measurement is statistical correlation between confidence estimate and some instance-level performance metric; Dong et al. (2018) use Spearman's correlation with respect to F1 score, while Vasudevan et al. (2019) use Pearson's correlation with respect to instance-level accuracy. A second option is to aggregate across instance-level calibration scores (so called proper scoring rules); examples include Brier score (Brier et al., 1950) and negative log-likelihood (Quinonero-Candela et al., 2005). A third approach involves partitioning confidence estimates into bins, assessing correlation for individual bins, and then aggregating across bin-level calibration results; one popular example of such an approach is Expected Calibration Error (ECE) (Naeini et al., 2015), which has been extended by Nixon et al. (2019) to assess calibration of all predictions rather than only the top one.

In this study, we apply some of the previously proposed calibration assessment methods – namely reliability diagrams and correlation with instance-level performance – to NLUs. In addition, we also measure calibration on rank level, enabling a more fine-grained analysis.

3 Background

When using an NLU, an utterance U is fed to the trained NLU, and the output normally includes the information in the following example:

```
{ 'utterance' : 'U' ,
  'top_intent': 'intent_1',
  'intent_ranking': {
    'intent_1': conf_1, # rank 1
    'intent_2': conf_2, # rank 2
    ... ,
    'intent_N': conf_N  # rank N
  }
}
```

The output of the NLU given an utterance U is a prediction consisting of the user utterance, the top intent and an intent ranking. The intent ranking consists of the N-best intent hypotheses along with their corresponding confidence estimates. The confidence estimates reflect how confident the NLU model is regarding each hypothesis.

Figure 1 illustrates how NLUs are used in dialogue systems, involving a scenario where a user asks a dialogue system a question within the home domain. The user utterance (which can be typed by the user in a chat or captured by a speech recognizer) is sent to an NLU service which performs intent classification on the user utterance and returns a prediction with the top intent and the intent ranking. The results are sent to a dialogue manager that decides how to steer the dialogue based on the output from the NLU and some dialogue policy. In case of a high estimated confidence for the most likely hypothesis, the dialogue manager integrates the user's intent, and information is sent to the natural-language generator that generates a response which is uttered back to the user.

A dialogue system can use confidence estimates as a basis for choosing a grounding strategy (e.g. asking a control question when confidence is low), ambiguity detection and handling (e.g. asking a clarification question if the top-ranked intents have similar confidence estimates) or re-scoring of hypotheses based on contextual information not available to the NLU but to the dialogue manager (such as dialogue state).

Different NLUs may have different ways of computing confidence estimates, possibly reflect-

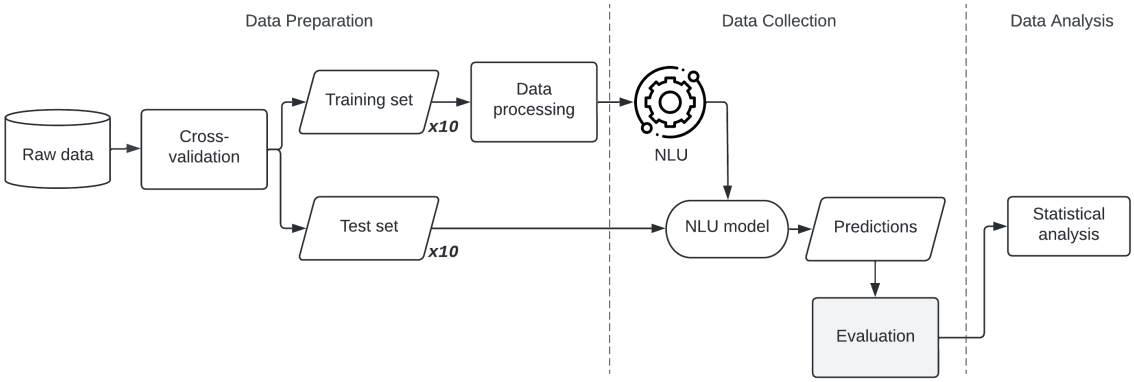


Figure 1: A dialogue system.

ing different notions of confidence. However, for the purpose of using the estimates in a dialogue system, we are interested in how well they reflect true probabilities. In section 4 we note variations in how confidence estimates are computed, but do not take these differences into account in our evaluation.

4 NLU services

NLU services can be used to construct the NLU component in a dialogue system. In this study, we chose NLU services (henceforth NLUs) based on the following criteria: i) can perform intent classification and ii) returns at least 10 top hypotheses in the output. We examine 5 NLUs: Watson Assistant (IBM, 2010), Language Understanding Intelligent Service (LUIS) (Microsoft, 2017), Snips (Snips, 2013), and Rasa (Rasa, 2016) (in two configurations).

Below, we briefly introduce the NLUs. Information about the NLUs, including the tested version, is summarized in table 1.

Watson Assistant Watson Assistant (henceforth Watson) is a cloud-based NLU developed by IBM. When parsing an utterance, Watson returns the top 10 hypotheses along with their confidence estimates. Confidence estimates are calculated independently for each intent that it has been trained on. In addition, Watson has an optional built-in "irrelevant" intent for out-of-scope (OOS) input.

LUIS LUIS (Language Understanding Intelligent Service) is provided by Microsoft and runs on the Azure cloud platform. LUIS trains an intent using provided positive examples and other intents as negative examples.

There is no limit in the number of hypotheses that LUIS returns; in other words, if the NLU is trained on N intents, then the intent ranking is of length N. A "None" intent for out-of-scope input is also supported, but requires the user to train it on example utterances.

Rasa Opensource Rasa is an open-source NLU provided by Rasa Technologies. It can run on different pipelines that are configurable which increases the flexibility of the NLU (Bocklisch et al., 2017). Rasa returns the top 10 hypotheses and their corresponding confidence estimates are normalized (they sum up to 1). Rasa does not offer a built-in out-of-scope intent.

In this study, we use with two different pipelines. The first pipeline uses the Sklearn intent classifier[2] while the second uses Dual Intent and Entity Transformer (DIET) (Bunk et al., 2020). We refer to the two pipelines above as Rasa-Sklearn and Rasa-DIET respectively.

Snips Snips is an AI voice platform for connected devices which provides an NLU for Python called Snips NLU (henceforth Snips). By default, Snips returns all hypotheses of all intents with confidence estimates, in addition to a "None" intent[3] for OOS input.

5 Dataset and data preparation

To conduct intent classification as a part of our evaluation, we build on the dataset proposed by Liu et al. (2021). The authors collect and annotate

[2] https://rasa.com/docs/rasa/components/#sklearnintentclassifier

[3] https://snips-nlu.readthedocs.io/en/latest/tutorial.html#the-none-intent

NLU	Packaging	Classifier Type	Version	OOS intent
Watson	Cloud-based service	Multiple-binary	Invoked in April 2022	Yes
LUIS	Cloud-based service	Multi-class	Invoked in April 2022	Yes
Snips	Open-source framework	Multi-class	v0.20.2	Yes
Rasa	Open-source framework	Multi-class	v2.4.3	No

Table 1: Summary of studied NLUs. (OOS = out of scope.)

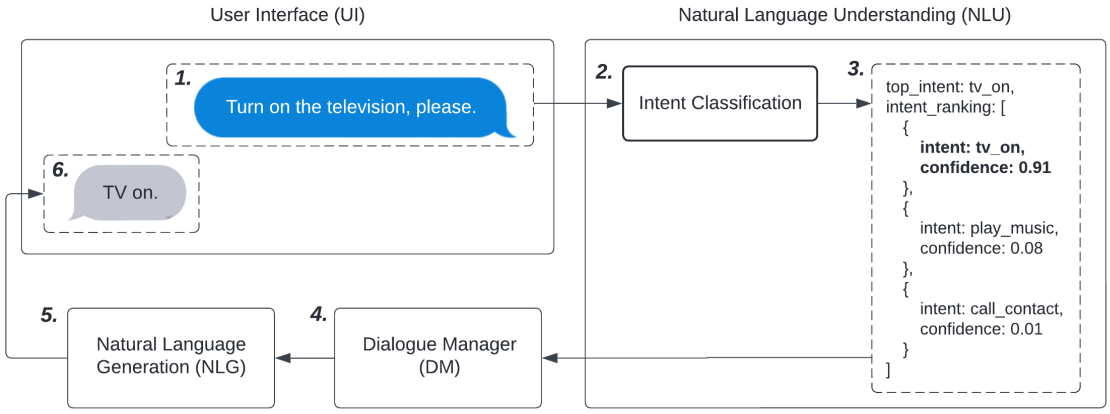


Figure 2: The evaluation process followed for each NLU to obtain the results; this process was repeated 5 times, 1 time per NLU model.

25716 user utterances for human-robot interaction and cover 64 intents, 18 scenarios and 21 domains. From this dataset, we select the 10 intents with the most examples (highest number of instances), yielding a total of 14962 utterances (see table 2).[4] We perform repeated random sub-sampling (Dubitzky et al., 2007) with 10 iterations to generate 10 random datasets; each dataset is divided with a 2:1 ratio into a training and testing sets respectively. (A breakdown by domain and/or scenario could also have been interesting, but was ruled out due to data sparsity.)

When analyzing the outputs from the NLUs, we exclude hypotheses with the OOS ("None"/"irrelevant") class in the intent ranking in order to ensure that all NLUs have the same intent ranking length and make their results comparable. (See section 8 for a discussion about OOS handling.)

6 Evaluation of confidence estimation

An overview of our study's execution is illustrated in figure 2. The evaluation is performed at two levels: rank and model. On rank level, the results are obtained for each rank across the NLUs, whereas on model level, the results of all ranks are aggregated.

The evaluation focuses on the calibration and performance of the NLUs. Calibration is measured using reliability diagrams and Spearman's correlation coefficient with respect to instance-level accuracy. The latter is measured through accuracy and F1-score. Evaluation is conducted for each split and results are averaged across splits.

6.1 Confidence calibration

Confidence calibration is the extent to which a model is able to produce confidence estimates that reflect the accuracy (true likelihood) of the respective intent hypotheses (Guo et al., 2017). For example, in a well-calibrated model, hypotheses with a confidence estimate of 0.7 are correct in 70% of the cases.

Reliability diagrams are visualizations of a model's calibration (Guo et al., 2017). They plot true likelihood (accuracy) of predictions as a function of confidence estimate. Hence, a perfectly-calibrated model is visualized as the identity func-

[4] Liu et al. (2021) provide user utterances in different forms: original (raw), with entity annotations, and normalized. In our study, we use the original user utterances.

Intent	Size	Example
query	5981	what's the time in australia
set	1748	wake me up at 9am on Friday
music	1205	start playing music from favourites
quirky	1088	I am not tired I am actually happy
factoid	1052	tell me comics of charlie chaplin
remove	986	cancel my 7am alarm
negate	939	you don't understand it right
sendemail	694	send a group mail to lookafter
explain	684	could you clarify me on it further
repeat	585	please let's start over
Total	14962 examples	

Table 2: Selected intents for the case study with their respective size (i.e. number of utterances) and one example utterance.

tion, and any deviation indicates miscalibration.

Reliability diagrams are plotted by partitioning predictions into bins, each of which represents a confidence range. In our study, we use 10 uniformly distributed bins, i.e. [0.0-0.1], [0.1-0.2], ...[0.9-1.0]. For each bin, mean confidence estimate and accuracy is calculated and plotted as a point.

Spearman's correlation coefficient In order to numerically measure the degree of calibration, we assess the correlation between confidence estimates (scores in the range 0-1) and instance-level accuracies (1 for correct classifications, 0 for incorrect classifications). More specifically, we measure the extent to which an increase in confidence estimate is associated with an increase in instance-level accuracy – in other words, the monotonicity of the relationship between confidence estimate and accuracy. The degree of monotonicity is measured using Spearman's correlation coefficient (Xiao et al., 2016).[5]

Given two variables (X and Y) of size N ($x_1, x_2, ..x_n$ and $y_1, y_2...y_n$ respectively), Spearman's correlation coefficient (ρ) is calculated through the formula:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where n is the number of samples, and d is the pairwise differences of the elements of the variables x_i and y_i.

[5]We choose Spearman's correlation rather than Pearson's correlation since our data is not normally distributed.

A perfectly-calibrated model has a Spearman's correlation coefficient of 1, while a correlation coefficient of 0 conveys a lack of correlation between confidence and accuracy.

Note that other approaches to numerically estimating calibration have been discussed in the literature, e.g. negative log-likelihood (Quinonero-Candela et al., 2005), Brier score (Brier et al., 1950) and expected calibration error (Nixon et al., 2019). Different measurement approaches have different advantages and weaknesses (Ashukha et al., 2020), and no gold standard seems to exist. In this study, we have opted for Spearman's correlation due to the fact that monotonicity in the relation between confidence estimate and accuracy is an important characteristic of good calibration. Spearman's correlation has been previously used to evaluate confidence scores for neural semantic parsers (Dong et al., 2018).

6.2 Performance

Since performance only considers the first rank, it can only be computed on a model level. To measure the performance, we use F1-score and accuracy. We use F1-score since it considers false positives and false negatives through precision and recall. Another reason is the unbalanced distribution of the example utterances across intents. We also include the accuracy since in this particular multi-domain dataset, false negatives have no major risks.

7 Results and analysis

In this section we present our results (averaged across the 10 splits). Our collected data are visual (reliability diagrams and calibration profiles) and numeric (Spearman's correlation, accuracy and F1-score). For our numeric results, we provide the average along with the standard deviation (SD), whereas for the visual results we provide the standard deviation in Appendix B to avoid cluttered diagrams.

7.1 Reliability diagrams

Calibration of the NLUs is visualized through reliability diagrams on model level (figure 3) and rank level (figures 4, 5, 6, 7). In the rank-level reliability diagrams, ranks 4-10 have been merged due to data sparsity.

Model-level results: On a model level (figure 3), all NLUs show a generally monotonic relationship between confidence and accuracy, except for Watson's lower ranges. In particular, Rasa-Sklearn is the closest to the gold standard, and is thus the best calibrated NLU according to this analysis. Moreover, Snips underestimates the true likelihood of predictions, while LUIS is over-confident. We observe a discrepancy in Watson's first 2 bins in the reliability diagram (figure 3) – a sudden underestimation followed by a drop that indicates an extreme overestimation.[6]

Rank-level results: On the first rank (figure 4), the NLUs are fairly well-calibrated in general. On ranks 2 (figure 5) and 3 (figure 6), the degree of calibration decreases (in comparison with the previous rank), for three of the NLUs (Watson, LUIS and Snips – all over-confident), while for the Rasa NLUs the trend seems inverted.

7.2 Calibration score and profile

The calculated Spearman's correlations between the confidence estimates and instance-level accuracy (table 3) show that Rasa-Sklearn has the highest Spearman's correlation with a score of ∼0.51, and is followed by LUIS, Rasa-DIET, Watson, and Snips with the lowest Spearman's correlation of ∼0.507. The difference between LUIS and Rasa-DIET is not significant, while the differences between each other pairs of NLUs are significantly different with a large effect size. (The entire list of t-test results is presented in table 6 in Appendix C.)

[6] As shown by figure 10 in Appendix A, Watson's first two bins are small in comparison with the other NLUs.

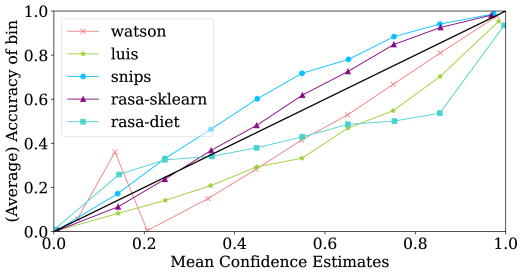


Figure 3: Model-level reliability diagram. The x-axis shows the mean confidence estimates in each bin, while the y-axis shows the mean accuracy of the confidence estimates in each bin (averaged across splits). The black diagonal line plots the identity function representing a gold standard of a perfectly-calibrated model.

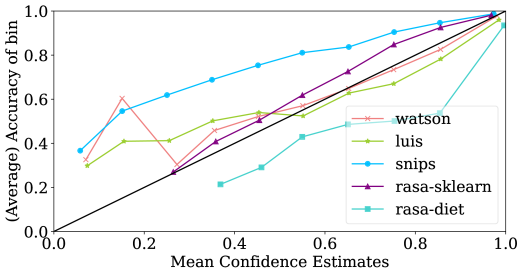


Figure 4: Rank-level reliability diagram on rank 1.

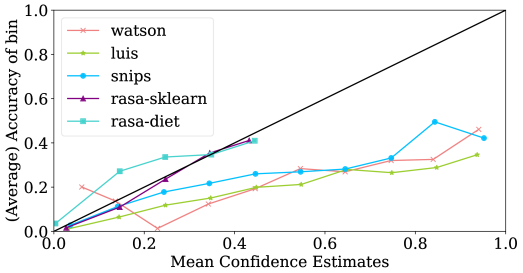


Figure 5: Rank-level reliability diagram on rank 2.

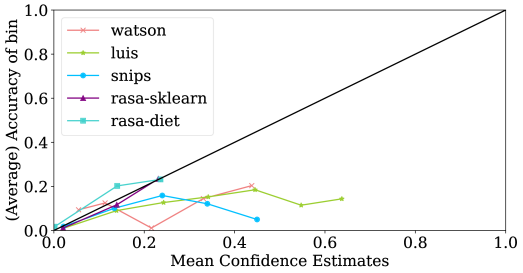


Figure 6: Rank-level reliability diagram on rank 3.

The model-level reliability diagram appears to resonate with the model-level calibration where

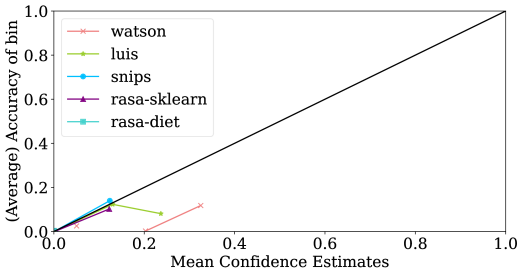


Figure 7: Rank-level reliability diagram on ranks 4-10.

Rasa-Sklearn shows the best calibration in the reliability diagram as well as the strongest monotonicity.

Figure 8 shows the *calibration profile* for each NLU – the Spearman's correlation coefficient as a function of rank. A perfect calibration profile (where calibration is perfect on each rank) would correspond to a straight line along the top of the diagram. In contrast, we can observe that all NLUs have noticeably lower Spearman's correlation for lower ranks. The decrease in Spearman's correlation for lower ranks may indicate that lower ranks are worse calibrated than higher ranks. However, there are reasons to treat these results with some caution.

We can note that the Spearman's correlation is generally lower on a rank level than on a model level. This can be explained by the fact that ranks extend across smaller ranges of confidence estimates (see model-level histogram in Appendix A), which increases variation in one of the correlated variables. Thus, it appears that a higher Spearman's correlation coefficient may be due to a larger variation in the confidence estimates. This may also explain that while figure 8 suggests a decrease in the calibration for lower ranks, the rank-level reliability diagrams show that Rasa-Sklearn and Rasa-DIET have better calibration in lower ranks. Still, on a model level, we take monotonicity to be a characteristic of well-calibrated NLUs. The stronger the monotonicity, the more one can trust an NLU's ranking of hypotheses in a prediction.

7.3 Performance

We measure the performance of the NLUs in intent classification by evaluating accuracy and F1-score. Performance is only evaluated on a model level since it considers the top hypothesis of the NLU's prediction. Our results of the accuracy and F1-scores are averaged across 10 splits for each NLU.

Accuracy: The results in table 4 show that Watson has the highest (∼0.92) and Rasa-Sklearn the lowest (∼0.87) accuracy. The accuracy scores of LUIS and Snips are not significantly different from each other, while all other differences between NLUs are statistically significant with a large effect size.

F1-score: The results in table 5 show that Watson has the highest (∼0.92) and Rasa-Sklearn the lowest (∼0.79) F1-score. All pairwise differences between the NLUs are significant with a large effect size. (The entire pairwise t-test results for accuracy and F1-score are included in table 7 in Appendix C.)

Our performance results are consistent with earlier work comparing Watson, LUIS and Rasa-Sklearn (Liu et al., 2021) that use the complete version of our dataset, and with Abdellatif et al. (2021) who use two datasets from the software engineering domain. However, the results differ from those in Braun et al. (2017) who use Telegram chatbot and StackExchange corpora in a question-answering domain and that has Watson as the worst performing NLU, and Rasa and LUIS on top.

A natural question at this point is whether calibration and performance are correlated. Figure 9 plots calibration (model-level Spearman's correlation) against model-level accuracy and F1 score. Judging from this, calibration and performance are not correlated, indicating a trade-off between calibration and performance (as previously reported for neural networks by Guo et al., 2017).

8 Discussion

In this study, we did not find support for any correlation between calibration and performance (judged by looking only at the top hypothesis). A consequence of this is that when it comes to choosing an NLU for a dialogue system, there is likely to be a trade-off between performance (good for getting the right interpretation) and calibration (good for detecting input that is ambiguous from the NLU perspective).

Differences in degree of calibration across ranks has been observed for all NLUs. Specifically, several of the NLUs are better calibrated for higher-ranking hypotheses than for lower-ranking ones.

NLU	Watson	LUIS	Snips	Rasa-Sklearn	Rasa-DIET
Mean	0.50838	0.50935	0.50669	**0.51024**	0.50906
Median	0.50851	0.50934	0.506491	**0.51026**	0.50888
***p*-value**	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
SD	0.00075	0.00055	0.00064	0.00046	0.00074

Table 3: Model-level calibration scores (Spearman's correlation coefficient ρ)

NLU	Watson	LUIS	Snips	Rasa-Sklearn	Rasa-DIET
Mean	**0.92287**	0.88726	0.88991	0.87263	0.90376
Median	**0.91997**	0.890405	0.89060	0.87866	0.89973
SD	0.00225	0.00417	0.00414	0.00386	0.003860

Table 4: (Averaged) accuracy scores of NLUs

NLU	Watson	LUIS	Snips	Rasa-Sklearn	Rasa-DIET
Mean	**0.92144**	0.88890	0.89029	0.79020	0.81890
Median	**0.91972**	0.89300	0.89166	0.79561	0.81716
SD	0.00234	0.00373	0.00407	0.00358	0.00331

Table 5: (Averaged) F1-scores of NLUs

For dialogue system developers, we may interpret this as indicating that it may be useful to look at the top two or three hypotheses when trying to detect ambiguity in input utterances. Looking at hypotheses ranked lower than 4 is likely to not be very informative. Fortunately, ambiguities are much more frequently 2-way (i.e. there are two possible interpretations of an input) or 3-way than 4-way or more.

It is worth stressing that one of the studied NLUs (Watson) is a multiple-binary classifier (it treats intents independently), while the others are multi-class (they treat intents as mutually exclusive). In this study, we do not investigate whether one type of classifier is more appropriate than another – presumably, both types have benefits and disadvantages. Nevertheless, since our dataset assumes a single correct class for a given utterance[7], our analysis may indirectly favour multi-class classifiers.

When interpreting our results, one should also consider that different NLUs handle out-of-scope (OOS) input differently. Specifically, among the studied NLUs only Rasa does not include an OOS intent. Our exclusion of out-of-scope intents from the intent rankings returned by the NLUs does not rule out the possibility that different OOS handling may have affected the result. A more level-playing field would have required all NLUs to either not consider OOS at all, or for all of them to be trained on the same OOS examples. Unfortunately, since Snips' OOS handling cannot be configured, neither of these options were available. (Larson et al. (2019) evaluated OOS detection for NLUs, but without considering confidence calibration.)

9 Conclusions and future work

We took established calibration measurement approaches and applied them to intent classification of publicly available NLUs. We also extended the chosen measurements with a rank-level analysis. Our findings show that the best calibrated NLU is Rasa-Sklearn and the least calibrated NLU is Snips, while Watson takes the lead as the best performing NLU and Rasa-Sklearn as the worst performing NLU. The results indicate a trade-off between confidence calibration and performance. We also showed differences in degree of calibration across ranks and discussed their implication

[7]Utterances in Liu et al.'s (2021) dataset, on which we build, are labelled with a single correct intent. There are cases of identical utterances for two different intents, but they are very rare (9 out of 25576 unique utterances).

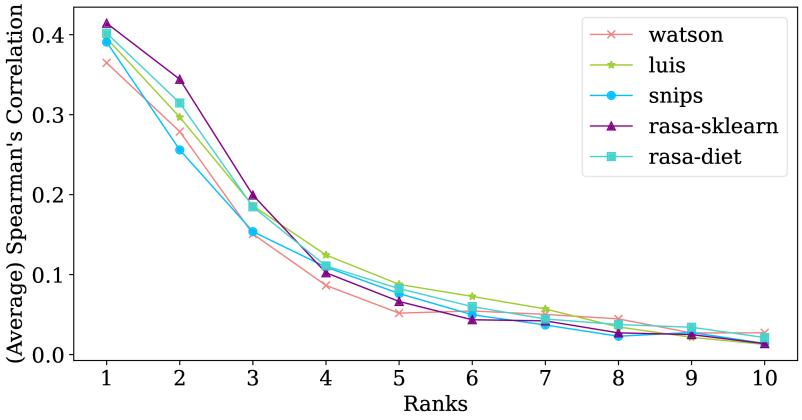
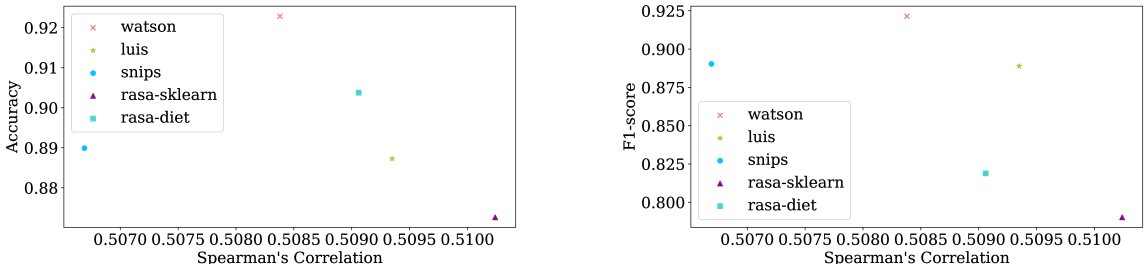


Figure 8: Calibration profiles for all NLUs (Spearman's correlation for ranks 1-10)



(a) Accuracy vs. Calibration

(b) F1-score vs. Calibration

Figure 9: (Model-level) accuracy (a) and F1-score (b) vs. calibration

for dialogue system development.

In future work, it would be interesting to extend the investigation with qualitative analyses of how differences in confidence estimation play out in concrete examples. It could also be valuable to find a better way of assessing how well the NLUs capture genuine ambiguity – something which is difficult with a dataset that assumes a single correct intent for a given utterance.

Acknowledgements

This work was supported by a grant from the Swedish Research Council (VR project 2014-39) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

References

Ahmad Abdellatif, Khaled Badran, Diego Costa, and Emad Shihab. 2021. A comparison of natural language understanding platforms for chatbots in software engineering. *IEEE Transactions on Software Engineering*.

Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. 2020. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.

Daniel Braun, Adrian Hernandez Mendez, Florian Matthes, and Manfred Langen. 2017. Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185.

Glenn W Brier et al. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. 2020. DIET: Lightweight language understanding for dialogue systems. *arXiv preprint arXiv:2004.09936*.

Massimo Canonico and Luigi De Russis. 2018. A comparison and critique of natural language understanding tools. *Cloud Computing*, 2018:120.

Li Dong, Chris Quirk, and Mirella Lapata. 2018. Confidence modeling for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753.

Werner Dubitzky, Martin Granzow, and Daniel P Berrar. 2007. *Fundamentals of data mining in genomics and proteomics*. Springer Science & Business Media.

Eric Gregori. 2017. Evaluation of modern tools for an OMSCS advisor chatbot. *SMARTech: smartech.gatech.edu*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

IBM. 2010. IBM Watson. Online available at: https://www.ibm.com/watson. Accessed on: 2022-04-14.

Sangkeun Jung. 2019. Semantic vector learning for natural language understanding. *Computer Speech & Language*, 56:130–145.

Rohan Kar and Rishin Haldar. 2016. Applying chatbots to the internet of things: Opportunities and architectural elements. *arXiv preprint arXiv:1611.03799*.

Falko Koetter, Matthias Blohm, Monika Kochanowski, Joscha Goetzer, Daniel Graziotin, and Stefan Wagner. 2019. Motivations, classification and model trial of conversational agents for insurance companies. In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,*, pages 19–30. INSTICC, SciTePress.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction*, pages 165–183. Springer.

M McTear, Z Callejas, and D Griol. 2016. The conversational interface: Talking to smart devices: Springer international publishing. *Doi: https://doi.org/10.1007/978-3-319-32967-3*.

Microsoft. 2017. LUIS (Language Understanding) - Cognitive Services. Online available at: https://www.luis.ai/home. Accessed on: 2022-04-14.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019. Measuring calibration in deep learning. *CVPR Workshops*, 2(7).

Joaquin Quinonero-Candela, Carl Edward Rasmussen, Fabian Sinz, Olivier Bousquet, and Bernhard Schölkopf. 2005. Evaluating predictive uncertainty challenge. In *Machine Learning Challenges Workshop*, pages 1–27. Springer.

Rasa. 2016. Rasa: Open source conversational AI. Online available at: https://rasa.com/. Accessed on: 2022-04-14.

Kumar Shridhar, Ayushman Dash, Amit Sahu, Gustav Grund Pihlgren, Pedro Alonso, Vinaychandran Pondenkandath, György Kovács, Foteini Simistira, and Marcus Liwicki. 2019. Subword semantic hashing for intent classification on small datasets. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–6. IEEE.

Snips. 2013. Snips.ai. Online available at: https://snips.ai/. Accessed on: 2022-04-14.

Svetlana Stoyanchev, Pierre Lison, and Srinivas Bangalore. 2016. Rapid prototyping of form-driven dialogue systems using an open-source framework. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–219.

Vishal Thanvantri Vasudevan, Abhinav Sethy, and Alireza Roshan Ghias. 2019. Towards better confidence estimation for neural models. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7335–7339. IEEE.

Chengwei Xiao, Jiaqi Ye, Rui Máximo Esteves, and Chunming Rong. 2016. Using Spearman's correlation coefficients for exploratory data analysis on big dataset. *Concurrency and Computation: Practice and Experience*, 28(14):3866–3878.

A Histograms of bin sizes

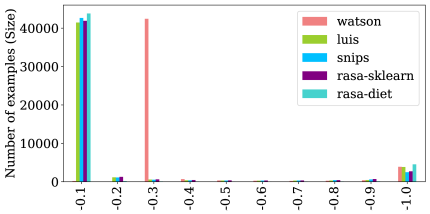


Figure 10: Model-level histogram

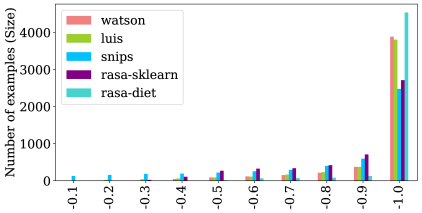


Figure 11: Rank-level (rank 1)

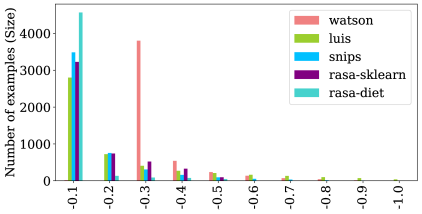


Figure 12: Rank-level (rank 2

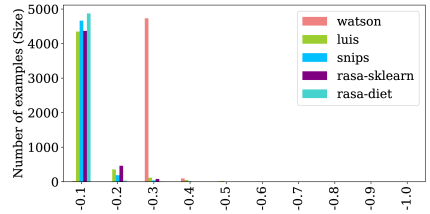


Figure 13: Rank-level (rank 3)

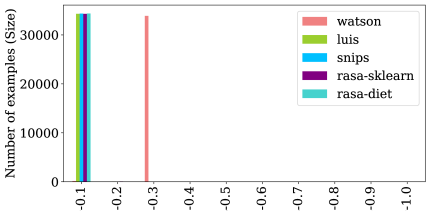


Figure 14: Rank-level (ranks 4-10)

B Reliability diagrams with standard deviation

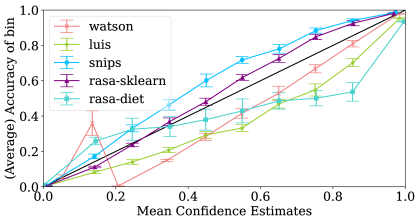


Figure 15: Model level

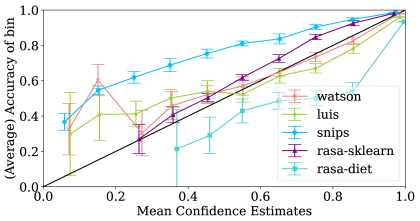


Figure 16: Rank level (rank 1)

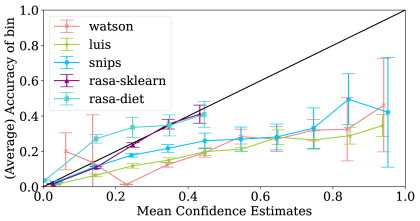


Figure 17: Rank level (rank 2)

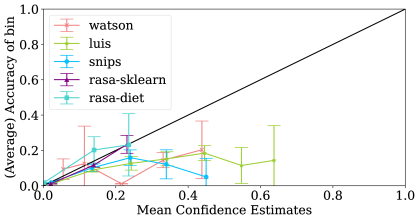


Figure 18: Rank level (rank 3)

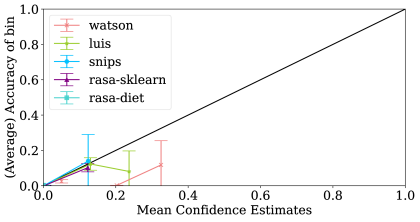


Figure 19: Rank level (ranks 4-10)

C T-test calculations

Pairwise Comp.	t-Statistic	*p*-value	df	Effect Size	SSD (*p*<.05)
(Watson, LUIS)	-3.1645	0.01147	9	L	Yes
(Watson, Snips)	4.9025	0.00084	9	L	Yes
(Watson, Rasa-Sklearn)	-5.4977	0.0003813	9	L	Yes
(Watson, Rasa-DIET)	-2.9555	0.01608	9	L	Yes
(LUIS, Snips)	25.569	<0.00001	9	L	Yes
(LUIS, Rasa-Sklearn)	-3.8306	0.00402	9	L	Yes
(LUIS, Rasa-DIET)	-78.645	0.2895	9	S	No
(Snips, Rasa-Sklearn)	-16.545	<0.00001	9	L	Yes
(Snips, Rasa-DIET)	-7.8118	<0.00001	9	L	Yes
(Rasa-DIET, Rasa-Sklearn)	-4.1319	0.002552	9	L	Yes

Table 6: T-test for pairwise NLUs' Spearman's correlation scores on a model level

Table 7: T-test for pairwise NLUs' performance

Pairwise Comp.	t Statistics	*p*-value	df	Effect Size	SSD (*p*<.05)
Accuracy					
(Watson, LUIS)	18.462	<0.00001	9	L	Yes
(Watson, Snips)	29.325	<0.00001	9	L	Yes
(Watson, Rasa-Sklearn)	25.059	<0.00001	9	L	Yes
(Watson, Rasa-DIET)	12.82	<0.00001	9	L	Yes
(LUIS, Snips)	-0.62904	0.545	9	N	No
(LUIS, Rasa-Sklearn)	11.672	<0.00001	9	L	Yes
(LUIS, Rasa-DIET)	-7.2468	<0.00001	9	L	Yes
(Snips, Rasa-Sklearn)	13.889	<0.00001	9	L	Yes
(Snips, Rasa-DIET)	-7.7684	<0.00001	9	L	Yes
(Rasa-DIET, Rasa-Sklearn)	18.968	<0.00001	9	L	Yes
F1-score					
(Watson, LUIS)	15.437	<0.00001	9	L	Yes
(Watson, Snips)	25.432	<0.00001	9	L	Yes
(Watson, Rasa-Sklearn)	79.213	<0.00001	9	L	Yes
(Watson, Rasa-DIET)	73.47	<0.00001	9	L	Yes
(LUIS, Snips)	1.1095	0.296	9	S	No
(LUIS, Rasa-Sklearn)	95.383	<0.00001	9	L	Yes
(LUIS, Rasa-DIET)	49.549	<0.00001	9	L	Yes
(Snips, Rasa-Sklearn)	135.47	<0.00001	9	L	Yes
(Snips, Rasa-DIET)	88.435	<0.00001	9	L	Yes
(Rasa-DIET, Rasa-Sklearn)	18.098	<0.00001	9	L	Yes

Pairwise Comp.	t Statistics	*p*-value	df	Effect Size	SSD (*p*<.05)
		Rank 1			
(Watson, LUIS)	-7.6715	<0.00001	9	L	Yes
(Watson, Snips)	-9.7613	<0.00001	9	L	Yes
(Watson, Rasa-Sklearn)	-11.441	<0.00001	9	L	Yes
(Watson, Rasa-DIET)	-10.782	<0.00001	9	L	Yes
(LUIS, Snips)	1.2402	0.2463	9	S	No
(LUIS, Rasa-Sklearn)	-4.45	0.0016	9	L	Yes
(LUIS, Rasa-DIET)	-1.8668	0.09477	9	M	No
(Snips, Rasa-Sklearn)	-5.7598	0.0002729	9	L	Yes
(Snips, Rasa-DIET)	-3.0576	0.01362	9	L	Yes
(Rasa-DIET, Rasa-Sklearn)	-6.754	<0.00001	9	L	Yes
		Rank 2			
(Watson, LUIS)	-3.2206	0.01048	9	L	Yes
(Watson, Snips)	6.4881	0.000113	9	L	Yes
(Watson, Rasa-Sklearn)	-17.398	<0.00001	9	L	Yes
(Watson, Rasa-DIET)	-8.6273	<0.00001	9	L	Yes
(LUIS, Snips)	9.9936	<0.00001	9	L	Yes
(LUIS, Rasa-Sklearn)	-9.7455	<0.00001	9	L	Yes
(LUIS, Rasa-DIET)	-3.7508	<0.00001	9	L	Yes
(Snips, Rasa-Sklearn)	-17.882	<0.00001	9	L	Yes
(Snips, Rasa-DIET)	-12.898	<0.00001	9	L	Yes
(Rasa-DIET, Rasa-Sklearn)	-11.323	<0.00001	9	L	Yes
		Rank 3			
(Watson, LUIS)	-6.7607	<0.00001	9	L	Yes
(Watson, Snips)	-0.6851	0.5105	9	S	No
(Watson, Rasa-Sklearn)	-13.616	<0.00001	9	L	Yes
(Watson, Rasa-DIET)	-6.2648	0.000147	9	L	Yes
(LUIS, Snips)	7.0407	<0.00001	9	L	Yes
(LUIS, Rasa-Sklearn)	-6.3356	0.0001352	9	L	Yes
(LUIS, Rasa-DIET)	0.46202	0.655	9	N	No
(Snips, Rasa-Sklearn)	-11.323	-7.0872	9	L	Yes
(Snips, Rasa-DIET)	-7.0872	<0.00001	9	L	Yes
(Rasa-DIET, Rasa-Sklearn)	-4.6652	0.001177	9	L	Yes
		Rank 4-10			
(Watson, LUIS)	-5.9362	<0.00001	49	L	Yes
(Watson, Snips)	-0.72951	0.4692	49	N	No
(Watson, Rasa-Sklearn)	0.078179	0.938	49	N	No
(Watson, Rasa-DIET)	-3.3111	0.00175	49	S	Yes
(LUIS, Snips)	9.1052	<0.00001	49	L	Yes
(LUIS, Rasa-Sklearn)	8.087	<0.00001	49	L	Yes
(LUIS, Rasa-DIET)	3.9641	0.0002393	49	M	Yes
(Snips, Rasa-Sklearn)	1.2524	0.2164	49	N	No
(Snips, Rasa-DIET)	-4.1725	0.0001228	49	M	Yes
(Rasa-DIET, Rasa-Sklearn)	5.2551	<0.00001	49	M	Yes

Table 8: T-test for pairwise NLUs' Spearman's correlation scores on a rank level

LAD: Language Models as Data for Zero-Shot Dialog

Shikib Mehri[♠] **Yasemin Altun**[♢] **Maxine Eskenazi**[♠]
[♠] Carnegie Mellon University [♢] Google
amehri@cs.cmu.edu,altun@google.com,max@cs.cmu.edu

Abstract

To facilitate zero-shot generalization in task-oriented dialog, this paper proposes *Language Models as Data* (LAD). LAD is a paradigm for creating *diverse* and *accurate* synthetic data which conveys the necessary structural constraints and can be used to train a downstream neural dialog model. LAD leverages GPT-3 to induce linguistic diversity. LAD achieves significant performance gains in zero-shot settings on intent prediction (**+15%**), slot filling (**+31.4 F-1**) and next action prediction (**+11 F-1**). Furthermore, an interactive human evaluation shows that training with LAD is competitive with training on human dialogs. LAD is open-sourced, with the code and data available at **https://github.com/Shikib/lad**.

1 Introduction

A long-standing goal of dialog research is to develop mechanisms for flexibly adapting dialog systems to new domains and tasks (Rastogi et al., 2020; Mosig et al., 2020). While the advent of large-scale pre-training (Devlin et al., 2018; Liu et al., 2019b; Zhang et al., 2019) has brought about significant progress in few-shot and zero-shot generalization across many different problems in Natural Language Processing (Brown et al., 2020; Wei et al., 2021), zero-shot generalization in **task-oriented dialog** remains elusive. A likely reason for this discrepancy is that dialog models require significant data because they need to learn task-specific **structural constraints**, such as the domain ontology and the dialog policy. While large language models (e.g., GPT-3) exhibit strong language understanding and generation abilities (Brown et al., 2020), they have no *a priori* knowledge of the structural constraints implied by a specific (unseen) problem setting (e.g., relevant intents, dialog policy, etc.). As such, in order to adapt a pre-trained LM for task-oriented dialog, it is necessary to *impose **structural constraints** on the **unstructured** representation space of a pre-trained model*. Fine-tuning moderately-sized language models (LMs) (e.g., BERT) with well-motivated inductive biases (Mitchell, 1980) facilitates sample-efficient learning of the structural constraints (Peng et al., 2020; Henderson and Vulić, 2020; Mehri and Eskenazi, 2021b). However, fine-tuning can be impractical (e.g., in academic settings) with large LMs (e.g., GPT-3) due to the cost, computational power and immutable architectures. To this end, this paper aims to address the following: *'How can we leverage the strong language understanding and generation abilities of large LMs to facilitate **zero-shot generalization** in task-oriented dialog?'*

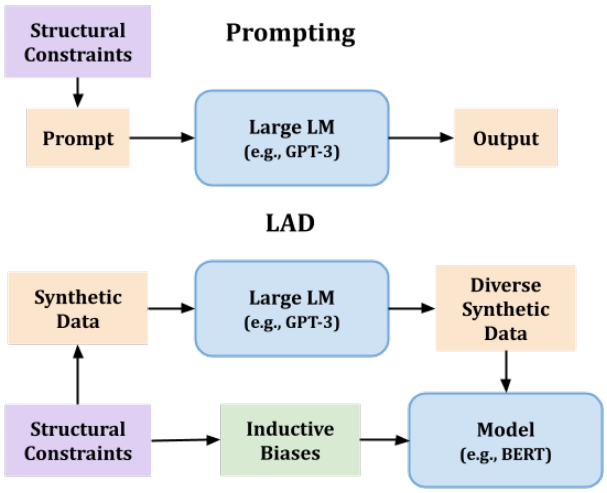


Figure 1: Prompting must convey the structural constraints through a natural language prompt. In contrast, LAD uses large LMs to induce diversity in a synthetic dataset. As such, LAD conveys structural constraints through both the synthetic data and the inductive biases in the downstream problem-specific models.

Given the in-context meta-learning abilities of large LMs (Brown et al., 2020), prior work has explored prompt-engineering or prompt-tuning (Reynolds and McDonell, 2021; Lester et al., 2021; Madotto et al., 2021). Well-designed prompts can convey the necessary structural constraints. How-

Proceedings of the SIGdial 2022 Conference, pages 595–604
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

ever, it is challenging to express complex constraints (e.g., a dialog policy) in natural language. Prompting also precludes inductive biases in the model (architecture, training algorithm, etc.) and over-relies on the meta-learning abilities of large LMs. As such, there is a tradeoff between prompting large LMs (i.e., generalizable NLU and NLG) and fine-tuning smaller LMs (i.e., problem-specific inductive biases, efficiency). A potential interpretation for the strength of large LMs is that they learn the distributional structure of language (Harris, 1954) by observing web-scale data (Sinha et al., 2021). Motivated by this interpretation, this paper proposes *Language Models as Data* (**LAD**).

LAD is a novel paradigm in which large LMs are used in a zero-shot domain-agnostic manner to induce *linguistic diversity* in synthetic data. Given a *minimal expression*[1] of the structural constraints (henceforth referred to as a **schema**), LAD (1) creates a seed synthetic dataset using domain-agnostic algorithms, (2) leverages large LMs to *reformulate* utterances, and (3) validates the resulting data to ensure adherence to the schema. The resulting synthetic data, which is sufficiently **diverse** and expresses the necessary **structural constraints**, can be used to train neural dialog models. In contrast to prompting, LAD facilitates zero-shot generalization by (1) leveraging the sophisticated abilities of large LMs (knowledge of the distributional structure of language) to induce *linguistic diversity* in the synthetic data while (2) maintaining inductive biases (motivated by the structural constraints) in the problem-specific model architectures.

The challenge of creating synthetic data that is indistinguishable from human-annotated data, both in its expression of structural constraints and in its diversity, is highly impractical (Lin et al., 2021; Feng et al., 2021). Instead, the goal of this work is to create synthetic data that is *sufficient* to train a sample-efficient and robust model. Therefore, the claim of this paper is that LAD can create synthetic data, conditioned on a minimal expression of structural constraints (i.e., a schema), that can be used to train robust and sample-efficient neural models and induce performance gains in zero-shot settings.

To validate this claim, LAD is applied to three problems in dialog: intent prediction, slot filling and next action prediction. Next action prediction is particularly difficult in zero-shot settings since the structural constraints include the *dialog policy*. LAD demonstrates significant gains across five datasets (**+10** to **+30** improvements on F-1 and accuracy) in zero-shot settings when evaluating on human-annotated corpora. To further validate the efficacy of LAD, an interactive evaluation with humans (over 1600 dialogs) is performed. The results of this interactive evaluation suggest that LAD can yield performance comparable to training on human dialogs. The claim of this paper is validated empirically across multiple datasets. LAD is shown to generate diverse and accurate synthetic data, which is subsequently used to train neural dialog models and facilitate zero-shot generalization.

[1]A minimal expression can be defined as the *smallest* amount of data necessary to express a structural constraint. For example, one utterance to define an intent class.

2 Definitions

Zero-shot generalization can be conceptualized as *imposing **structural constraints** on the unstructured representation space of a pre-trained model, using a given **schema** (i.e., minimal expression).* We begin with a neural network, $\mathcal{M}$, with general language understanding abilities and limited knowledge of task-oriented dialog (e.g., BERT (Devlin et al., 2018)). The necessary **structural constraints** that $\mathcal{M}$ must learn are implied by the target dialog setting, i.e., the problem (e.g., next action prediction), the domain (e.g., restaurants) and the task (e.g., restaurant reservation). These structural constraints conceptually define the desired properties for the representations of $\mathcal{M}$, i.e., *what must be learned* by $\mathcal{M}$. In a full-shot setting, the constraints are conveyed by a human-annotated dataset and thereby learned through supervised learning. In contrast, the goal in zero-shot generalization is to learn these structural constraints from a minimal expression, i.e., a **schema**. The following sections formally define structural constraints and schemas.

Throughout this paper, **zero-shot** refers to a setting wherein the only *human-annotated* data is the schema. Since a schema is a minimal expression of the necessary structural constraints, we argue that it is impossible to use less data, without making assumptions about the prior knowledge of a pre-trained model. Such assumptions would limit the generality of a method for zero-shot generalization.

2.1 Structural Constraints

To effectively adapt a model, particularly in zero-shot settings, it is imperative to define *what the model must learn*. Structural constraints conceptualize the desired properties for the representations

of a model $\mathcal{M}$. Understanding these structural constraints allows us to design an effective paradigm to facilitate zero-shot generalization. Concretely, knowledge of the structural constraints influences (1) the inductive biases (Mitchell, 1980) in the model architecture, (2) the design of the schema, and (3) the algorithms used to create synthetic data.

Intent prediction, for example, is the problem of classifying an utterance $u \in \mathcal{U}$ to an intent $i \in \mathcal{I}$. An intent prediction model $\mathcal{M}_I$ must learn to produce similar representations $\mathcal{M}_I(u)$ for all utterances that have the same intent. Learning this structural constraint is equivalent to transforming the unstructured representation space of $\mathcal{M}$ to the structured output space (i.e., the intent classes).

In the problem of **slot filling**, for a given utterance $u = \{w_1, w_2, \ldots, w_n\}$ and a slot key $s \in \mathcal{S}$, we must predict the corresponding slot value for s. The value will either be a contiguous span from u, $w_{i:i+k}$, or none. A slot filling model $\mathcal{M}_S$ must learn two sets of structural constraints. First, the representation of u (or the contextual representation of $w \in u$) must follow the structural constraints of intent prediction. Second, each slot value representation $\mathcal{M}_S(w_{i:i+k})$ should be similar to other values for the slot s. These two constraints impose structure on both the utterance-level and the span-level representations of $\mathcal{M}_S$.

The structural constraints of intent prediction and slot filling are straightforward and are often learned by a linear layer in supervised settings (Casanueva et al., 2020; Mehri et al., 2020). The constraints for the problem of **next action prediction** are more complex. Next action prediction is the problem of predicting the next system action $a \in \mathcal{A}$ conditioned on the dialog history $u_1, u_2, \ldots, u_n$ according to some dialog policy. Given the intents and slots in the dialog history, $\mathcal{I}_D = \{i_1, i_2, \ldots, i_m\}$ and $\mathcal{S}_D = \{s_1, s_2, \ldots, s_k\}$, the dialog policy can be expressed as a function of these intents and slots, $a = \text{policy}(\mathcal{I}_D, \mathcal{S}_D)$. As such a next action prediction model $\mathcal{M}_A$ must learn (1) the structural constraints of intent prediction, (2) of slot filling and (3) the mapping defined by the policy function. The complexity of third constraint led to the schema-guided paradigm (Mehri and Eskenazi, 2021b), wherein the policy is explicitly expressed rather than being learned implicitly.

2.2 Schema

While structural constraints conceptualize what a model $\mathcal{M}$ must learn, the schema is a minimal expression of these constraints. Imagine that our objective is to train a human (i.e., $\mathcal{M}$ with human-level language understanding and reasoning abilities) to perform task-oriented dialog. Structural constraints define *what* the human must learn. The schema is the *minimum* amount of information needed, for the human to learn the necessary structural constraints, without prior knowledge.

For **intent prediction**, we define the schema to be a single utterance u for each intent $i \in \mathcal{I}$. **Slot filling** similarly relies on one utterance u for each slot type $s \in \mathcal{S}$. However, this one utterance only conveys the first structural constraint of slot filling. To ensure that $\mathcal{M}_S$ can learn meaningful span-level representations, the schema for slot filling also includes multiple[2] examples of values for each slot.

Next action prediction has three constraints. The first two constraints are equivalent to those of intent prediction and slot filling. As such, the schema includes both (1) one utterance for each intent and (2) a set of slot values for each slot type. To express the structural constraints of the dialog policy, we leverage the graph-based representations of the task-specific dialog policy proposed by Mosig et al. (2020) and Mehri and Eskenazi (2021b).

3 LAD: Language Models as Data

Despite exhibiting strong language understanding and generation abilities (Brown et al., 2020), large LMs have no *a priori* knowledge of the structural constraints of task-oriented dialog. Furthermore, imposing the necessary structural constraints on large LMs is impractical due to (1) the difficulty of fine-tuning (cost, computation, immutable architectures) and (2) the limitations of natural language prompts. As such, *Language Models as Data* (**LAD**) uses GPT-3 (Brown et al., 2020) to generate **diverse** synthetic data that express the necessary task-specific **structural constraints** and can therefore be used to train neural dialog models.

LAD is a framework for inducing zero-shot generalization in task-oriented dialog by creating *diverse* and *accurate* synthetic data. LAD, visualized in Figure 2, is a three step process: (§3.1) domain-agnostic algorithms generate a *seed dataset* from a schema, (§3.2) GPT-3 *reformulates* utterances in

[2]While the number of slot value examples could potentially reduced to 1, up to 20 are used in this paper.

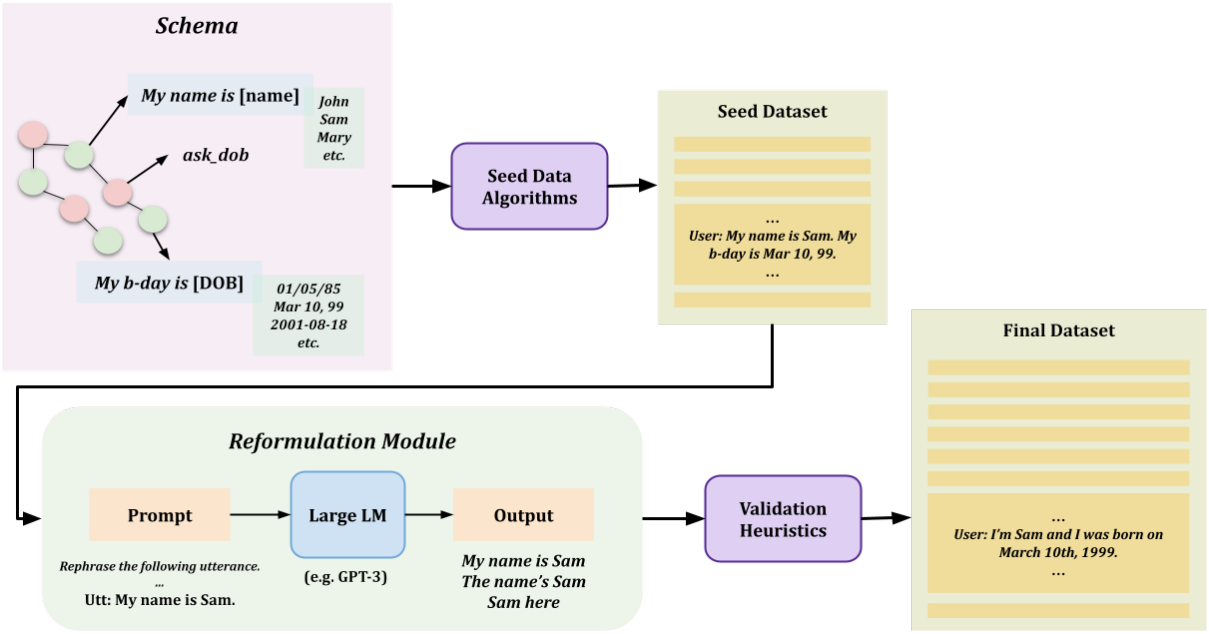


Figure 2: Visualization of LAD. (1) Domain-agnostic algorithms use the schema to create a seed dataset which conveys the necessary structural constraints. (2) Large LMs reformulate individual utterances to add linguistic diversity. (3) Validation heuristics are used to ensure adherence to the schema.

order to induce linguistic diversity, (§3.3) heuristics are used to *validate* the reformulated data to ensure adherence to the schema. LAD facilitates zero-shot generalization by explicitly leveraging the strengths of large LMs (knowledge of the distributional structure of language) without sacrificing the inductive biases (motivated by structural constraints) in the downstream neural dialog models.

3.1 Seed Data Creation

LAD begins by creating seed synthetic data from a given schema. This is a domain-agnostic process that aims to generate synthetic data which accurately convey the necessary structural constraints.

For **intent prediction**, the schema consists of one utterance for each intent class (sampled from the original corpus) and is used as the seed dataset. For **slot filling**, the schema consists of one manually-written template utterance and multiple slot values for each slot type. To construct the seed data: (1) begin with the utterance templates from the schema (e.g., *'My first name is {first_name}'*), (2) exhaustively combine template utterances to ensure coverage of slot type combinations, and (3) fill slot values by sampling from the schema.

The relative complexity of the structural constraints for **next action prediction**, particularly the dialog policy, necessitates a more sophisticated algorithm for generating the seed data. In order to avoid over-fitting and to ensure that the structural constraints are effectively learned by the model, it is imperative that the synthetic data produced by LAD be diverse and realistic. While linguistic diversity is induced through the reformulation with GPT-3, the synthetic dialogs created for next action prediction must also exhibit diversity of *user behavior*. The dialog policy expressed by the schema deterministically defines the *system* behavior. However, users should be able to deviate from the policy, e.g. by providing information out of turn. To account for this, Algorithm 1 in the Appendix generates a dialog by traversing the dialog policy graph and randomly combining multiple template utterances (e.g., *'**System:** What is your name? **User:** My name is John. My phone number is...'*).

3.2 Reformulation

To ensure that downstream neural dialog models can effectively learn the structural constraints, it is imperative that the synthetic data is sufficiently diverse. The seed synthetic data is formulaic and artificial: (1) there is a single template utterance for each user action and (2) when multiple user actions are combined they are simply concatenated. As such, the goal of the reformulation step is two-fold: (1) to induce linguistic diversity and (2) to rephrase concatenations of disjoint template utterances (*'My name is Sarah. I want to plan a party. The day*

should be Sunday') into a natural utterance (*'I'm Sarah and I'd like to plan a party for Sunday.'*).

To reformulate utterances in a domain-agnostic manner, LAD leverages the in-context meta-learning abilities of GPT-3 (Brown et al., 2020). Through manual experimentation in the OpenAI Playground[3], an appropriate prompt is constructed. The prompt begins with an instruction (*'Given a set of sentences, generate 5 natural utterances that convey the same meaning.'*) and includes six examples (details can be found in the Appendix).

Rather than producing a single reformulation of the input, the chosen prompt instructs GPT-3 to generate **five** utterances. Through the examples provided in the prompt, GPT-3 learns that it should produce five *diverse* reformulations. As such, linguistic diversity is induced through both the decoding algorithm and the six examples in the prompt.

3.2.1 Scalability

The cost of the GPT-3 API is approximately $0.05 USD per reformulation. In order to generate a substantial amount of synthetic data without incurring significant costs, the reformulation step of LAD must be performed in a scalable manner. The seed utterances are grouped by their intents and slot keys (e.g., 'name;date;time', 'name;date', 'date;time'). A subset of utterances in each group is reformulated. These reformulated utterances are used as templates and the slot values are randomly replaced. In this manner, the cost scales with respect to the number of distinct intent/slot combinations rather than the desired size of the synthetic dataset.

3.3 Validation

The seed data will always adhere to the schema and therefore *accurately* convey the necessary structural constraints. However, the reformulated utterances may not be accurate. GPT-3 may modify the intended meaning of an input utterance, for example by ignoring certain slot values. To ensure that the structural constraints are accurately expressed in the final dataset, the reformulation step of LAD filters out erroneous reformulations. For slot filling and next action prediction, this is done by ensuring that all of the slot values present in the original utterance (from the seed dataset) are also present in the reformulated utterances (produced by GPT-3).

[3]https://beta.openai.com/playground

Original Dataset	Seed	LAD	Cost (USD)
Intent Prediction			
HWU64 (8955)	64	800	$19
CLINC150 (15000)	150	1664	$43
Banking77 (8633)	77	848	$25
Slot Filling			
Restaurant8k (8633)	85	32000	$89
Next Action Prediction			
STAR (1200)	24000	22327	$226

Table 1: Statistics for the synthetic datasets created by LAD. This table lists the size of the original dataset, the seed dataset and the final synthetic dataset produced by LAD. The last column indicates the approximate cost of using GPT-3 for each of the datasets.

3.4 Dataset Statistics

LAD is evaluated on five different datasets. For intent prediction, Banking77 (Casanueva et al., 2020), CLINC150 (Larson et al., 2019), and HWU64 (Liu et al., 2019a) are used. For slot filling, Restaurant8k (Coope et al., 2020) is used. For next action prediction, STAR (Mosig et al., 2020) is used. Given a human-annotated corpus, a schema is created to express the necessary constraints. LAD is then leveraged to create a synthetic dataset conditioned on the schema. Table 1 describes the size and creation cost of each of the synthetic datasets.

4 Experiments

This paper claims that LAD can use a schema to create a sufficiently diverse and accurate synthetic dataset, which can be used to train neural dialog models and facilitate performance gains in zero-shot settings. To validate this claim, experiments are carried out on intent prediction, slot filling and next action prediction across five datasets.

For each problem, an appropriate model from prior work is identified. The chosen models (1) exhibit strong zero-shot and few-shot generalizability, and (2) are open-source. Though LAD is not guaranteed to produce *perfectly* accurate and diverse data, the inductive biases in the chosen models make them more robust to potential errors and limitations in the synthetic data.

4.1 Intent Prediction

CONVBERT+*Example-Driven+Observers* (CBEO) (Mehri and Eric, 2021) is used for intent prediction. CBEO learns to predict utterance

Model (Training Data)	BANKING77	CLINC150	HWU64
CBEO (ONE-SHOT)	31.36	53.96	43.12
CBEO (ONE-SHOT + LAD)	**51.17**	**68.11**	**65.50**
CBEO (FULL-SHOT)	93.83	97.31	93.03

Table 2: Experimental results on intent prediction. We report the accuracy of training CBEO on (1) one utterance/intent (i.e., the seed data) and (2) the synthetic data produced by LAD. For reference, we also show the results reported by Mehri and Eric (2021) obtained with full human-annotated training datasets.

Model	F-1
Zero-Shot Results	
CONVEX (HENDERSON AND VULIĆ, 2020)	5.2
COACH+TR (LIU ET AL., 2020)	10.7
GENSF (MEHRI AND ESKENAZI, 2021A)	19.5
GENSF + LAD	**50.9**
Non Zero-Shot Results	
GENSF (64 UTTERANCES)	72.2
GENSF (8633 UTTERANCES)	96.1

Table 3: Experimental results on the Restaurant8k corpus. We compare GENSF + LAD with zero-shot results reported by prior work. For reference, we also show the performance of models (reported by prior work) when trained in few-shot and full-shot settings.

Model	F-1
Zero-Shot Results	
BERT+S (MOSIG ET AL., 2020)	28.12
SAM (MEHRI AND ESKENAZI, 2021B)	53.31
SAM + LAD	**64.36**
Full-Shot Results	
SAM (MEHRI AND ESKENAZI, 2021B)	70.38

Table 4: Experimental results on the STAR corpus. SAM + LAD is compared with zero-shot results reported by prior work. For reference, the performance of SAM when trained on the full corpus is also shown.

intents by explicitly comparing to a set of examples. Predicting intents through an explicit non-parametric comparison to examples is an inductive bias that facilitates sample-efficient learning of the structural constraints.

The experimental results shown in Table 2 demonstrate that the synthetic data produced by LAD significantly increase performance on one-shot[4] intent prediction. LAD facilitates **15%+** accuracy improvement across all three intent prediction datasets. For intent prediction, LAD does not use any heuristics during the creation of the seed data or during the validation step. As such, these improvements can be attributed to the reformulation step, which leverages the prompt-driven generation abilities of GPT-3 (Brown et al., 2020).

4.2 Slot Filling

For slot filling, experiments are carried out with GENSF (Mehri and Eskenazi, 2021a) which currently has SoTA results on the Restaurant8k corpus (Coope et al., 2020), in both zero-shot and full-shot settings. GENSF reformulates slot filling as response generation in order to better leverage the capabilities of DialoGPT (Zhang et al., 2019).

As shown in Table 3, GENSF + LAD achieves a **+31.4** F-1 improvement over GENSF on the test set of Restaurant8k, without observing any examples from the corpus. GENSF + LAD learns to detect slots in the restaurant domain given only the schema, which consists of (1) a single manually written utterance for each slot type and (2) a collection of up to 20 slot values for each slot type. This significant performance improvement in zero-shot generalization validates the claim of this paper for the problem of slot filling. LAD is able to create synthetic data which effectively teaches GENSF the necessary structural constraints.

However, as shown by Mehri and Eskenazi (2021a), GENSF achieves a 72.2 F-1 score by only observing 64 human-written examples. Despite the relative success of LAD in zero-shot settings, there remains significant room for improvement.

[4]This setting is characterized as one-shot since the utterances in the schema are sampled from the respective dataset.

Model (Training Data)	COMPLETE %	ASKS ALL %	AVOIDS REDUNDANCY %
SAM (ZERO-SHOT)	98.02	76.15	78.90
SAM (FULL-SHOT)	98.31	75.69	80.65
SAM + LAD	98.52	**78.39**	79.13

Table 5: Results of the interactive human evaluation. We compare three models: (1) SAM (ZERO-SHOT), (2) SAM (FULL-SHOT) and (3) SAM + LAD. The three columns correspond to the three post-dialog questions: (1) task completion, (2) asking all necessary information and (3) avoiding redundancy. Results in boldface are statistically significant by one-tailed t-test ($p < 0.05$).

4.3 Next Action Prediction

Next action prediction is particularly challenging due to the complexity of the structural constraints. In addition to the constraints of intent prediction and slot filling, next action prediction models must also learn to follow the *dialog policy*. SAM (Mehri and Eskenazi, 2021b) learns to predict the system action by attending to a graph-based representation of the dialog policy. Explicitly attending to the dialog policy is an inductive bias that facilitates zero-shot generalization to unseen tasks.

Table 4 shows the results for three models. BERT+S (Mosig et al., 2020) trains a BERT model to attend to a rudimentary graph-based representation of the dialog policy. SAM (Mehri and Eskenazi, 2021b) improves the model architecture and introduces more expressive policy graphs. These two models are trained on the STAR corpus, which includes 24 different tasks and 24 different policy graphs. The zero-shot results are obtained by training on $n - 1$ tasks (i.e., 23) and evaluating on the remaining task, repeated 24 times. In contrast, SAM + LAD observes **no human-written dialogs** whatsoever. Instead, SAM+LAD is trained only on the synthetic dialogs produced by LAD.

In the zero-shot setting, SAM + LAD achieves an **+11.05** F-1 improvement over SAM. Furthermore, this result is only **6.02** points below the full-shot results of SAM. This significant gain further validates the claim of this paper. SAM + LAD learns the necessary structural constraints using only the synthetic data produced by LAD.

4.4 Interactive Human Evaluation

SAM + LAD achieves strong zero-shot results on the STAR corpus, especially relative to the performance of SAM (FULL-SHOT). This leads us to question the performance gap between these two models. Is the full-shot model better at next action prediction, or is it just better at modelling artifacts in the STAR corpus? STAR is known to have some degree of inconsistency with the policy graphs (Mosig et al., 2020). Furthermore, static evaluation is not necessarily reflective of the performance of a model in **real settings**. Because of variable user behavior, there may be a distribution shift between the STAR corpus and interactive settings. To this end, we perform an interactive human evaluation using Amazon Mechanical Turk (AMT).

Three models are evaluated: (1) SAM (ZERO-SHOT), (2) SAM (FULL-SHOT) and (3) SAM + LAD. Ten scenarios are defined, each of which consists of an objective (e.g, *'You want to plan a party'*) and slot values (e.g., `Name`: Kevin, `Date`: Sunday, `Num Guests`: 85). An AMT worker is instructed to interact with a dialog system according to the provided scenario. Upon completion of the dialog, three questions are answered:

1. Did the system successfully complete the dialog?
2. Did the system ask for all of the necessary information?
3. Did the system ask for information that you had already provided it?

The instructions (see Appendix) tell the worker to interact *naturally* (e.g., by providing information out of turn). Detailed instructions, including examples and counter-examples, are provided for the three post-dialog questions. Pre-screening is performed to ensure that AMT workers read and understood these instructions. During pre-screening, the worker must answer the post-dialog questions given two completed dialogs and the corresponding scenarios. Workers with a score of at least 5/6 qualify to participate in the interactive evaluation (45% of workers pass the pre-screening). Pre-screening is paid $0.75USD, regardless of the result. Each HIT (Human Intelligence Task) of the interactive evaluation includes five scenarios and pays $3.25USD (approx. 10 minutes). A post-hoc quality check is performed to remove erroneous

annotations. Simple heuristics are constructed to predict the post-dialog answers and any discrepancies with the annotations are manually verified. If an error is identified through manual validation, the annotation is removed. This form of validation is a necessary alternative to outlier detection or measures of inter-annotator agreement, since interactive dialogs are independent thereby making standard measures of data quality unsuitable.

1628 dialogs were collected, with at least 500 for each system. The results, shown in Table 5, demonstrate that the performance of all three models is fairly similar in interactive settings. For the second post-dialog question, SAM+LAD asks for all of the necessary slots **+2.7%** more often. Assuming that the number of observations is equal to the total number of turns, this result is statistically significant ($p < 0.05$) by one-tailed t-test.

Both SAM (FULL-SHOT) and SAM (ZERO-SHOT) are trained on human dialogs, though the latter does not observe data from the target task. In contrast, SAM + LAD is trained only on synthetic data produced by LAD. The comparable performance of SAM (ZERO-SHOT) and SAM (FULL-SHOT) is noteworthy and can potentially be explained by two facts: (1) the interactive dialogs are sampled from a different distribution (e.g., more informal, typos, more slots per utterance) from the STAR corpus, making the evaluation equally difficult for both systems, (2) SAM (ZERO-SHOT) has observed dialogs from the domain (e.g., seen `bank-balance` when evaluating on `bank-fraud-report`). Despite not observing any human dialogs, in interactive settings SAM + LAD attains zero-shot performance comparable to training on human dialogs from the STAR corpus. Though there remains significant room for improvement, the results of this interactive human evaluation demonstrate the efficacy of LAD. By leveraging the strengths of large language models to induce linguistic diversity, LAD produces synthetic data that effectively conveys the necessary structural constraints and facilitates zero-shot generalization, even in challenging interactive settings.

5 Related Work

5.1 User Simulators for Task-Oriented Dialog

The use of synthetic data in task-oriented dialog is a long-standing approach. Early dialog research leveraged user simulators for evaluation and optimization (Eckert et al., 1997; Scheffler and Young, 2000; Schatzmann et al., 2006). Schatzmann et al. (2007) propose a probabilistic agenda-based user simulator for bootstrapping a POMDB dialog system, demonstrating reasonable task completion rates. Georgila et al. (2006) train an n-gram user simulator which models both ASR and understanding errors. González et al. (2010) explicitly model user cooperativeness in a statistical user simulator.

Li et al. (2016) propose an agenda-based user simulator for training dialog policies with RL. Crook and Marin (2017) train a sequence-to-sequence model for user simulation. Kreyssig et al. (2018) introduce the neural user simulator (NUS), which trains a sequence-to-sequence network conditioned on user goals and the dialog history, outperforming existing methods on an interactive evaluation. Shi et al. (2019) carry out a comprehensive analysis of six different user simulators, with different dialog planning and generation methods. A key takeaway of this analysis is using agenda-based simulators to train RL systems generally results in higher performance on human evaluation. Lin et al. (2021) propose a domain-indepdendent transformer-based user simulator (TUS). The feature representations of TUS are domain-independent, thereby facilitating learning of cross-domain user behavior. TUS is trained on MultiWOZ (Budzianowski et al., 2018) and can effectively transfer to unseen domains.

LAD can be characterized as an agenda-based simulator, wherein the schema describes the ontology and the policy. The core novelty of LAD in the context of prior work is three-fold: (1) large LMs to induce linguistic diversity, (2) *zero-shot* domain-agnostic synthetic data creation, and (3) the schema as a standardized expression of structural constraints. LAD can potentially be further improved by incorporating strategies from prior work, such as modelling cooperativeness (González et al., 2010) or ASR errors (Georgila et al., 2006).

5.2 Using Large Language Models

Large language models (Brown et al., 2020; Chowdhery et al., 2022) exhibit strong language understanding, generation and reasoning abilities. Prompting is the dominant paradigm for leveraging large LMs for various downstream problems (Reynolds and McDonell, 2021; Lester et al., 2021). Madotto et al. (2021) demonstrate the efficacy of few-shot prompting for both open-domain and task-oriented dialog, with a focus on response genera-

tion and conversational parsing.

Several papers have used GPT-3 to generate synthetic data (Yoo et al., 2021; Wang et al., 2021). These approaches rely on GPT-3 to generate the labels and are not suitable for task-oriented dialog. To our knowledge, LAD is the first paper to leverage large LMs to *reformulate* utterances, in order to create synthetic data for task-oriented dialog.

6 Conclusion

In an effort to leverage the abilities of large LMs to facilitate zero-shot generalization in task-oriented dialog, this paper introduces **LAD**. LAD creates diverse and accurate synthetic data, in order to convey the necessary setting-specific structural constraints to neural dialog models. LAD achieves significant performance gains on zero-shot intent prediction, slot filling and next action prediction across five datasets. Furthermore, LAD is shown to perform competitively in interactive human evaluation, without observing human-annotated data.

7 Acknowledgements

This work was funded by a Google Research Collabs grant.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz–a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Sam Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-convert: Few-shot span extraction for dialog with pretrained conversational representations. *arXiv preprint arXiv:2005.08866*.

Paul A Crook and Alex Marin. 2017. Sequence to sequence modeling for user simulation in dialog systems. In *INTERSPEECH*, pages 1706–1710.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, pages 80–87. IEEE.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: learning and evaluation. In *Interspeech*, pages 1065–1068. Citeseer.

Meritxell González, Silvia Quarteroni, Giuseppe Riccardi, and Sebastian Varges. 2010. Cooperative user models in statistical dialog simulators. In *Proceedings of the SIGDIAL 2010 Conference*, pages 217–220.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Matthew Henderson and Ivan Vulić. 2020. Convex: Data-efficient and few-shot slot labeling. *arXiv preprint arXiv:2010.11791*.

Florian Kreyssig, Inigo Casanueva, Pawel Budzianowski, and Milica Gasic. 2018. Neural user simulation for corpus-based policy optimisation for spoken dialogue systems. *arXiv preprint arXiv:1805.06966*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Xiujun Li, Zachary C Lipton, Bhuwan Dhingra, Lihong Li, Jianfeng Gao, and Yun-Nung Chen. 2016. A user simulator for task-completion dialogues. *arXiv preprint arXiv:1612.05688*.

Hsien-chin Lin, Nurul Lubis, Songbo Hu, Carel van Niekerk, Christian Geishauser, Michael Heck, Shutong Feng, and Milica Gašić. 2021. Domain-independent user simulation with transformers for task-oriented dialogue systems. *arXiv preprint arXiv:2106.08838*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019a. Benchmarking natural language understanding services for building conversational agents. *arXiv preprint arXiv:1903.05566*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zihan Liu, Genta Indra Winata, Peng Xu, and Pascale Fung. 2020. Coach: A coarse-to-fine approach for cross-domain slot filling. *arXiv preprint arXiv:2004.11727*.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.

Shikib Mehri and Mihail Eric. 2021. Example-driven intent prediction with observers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2979–2992.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tur. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *arXiv preprint arXiv:2009.13570*.

Shikib Mehri and Maxine Eskenazi. 2021a. Gensf: Simultaneous adaptation of generative pre-trained models and slot filling. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 489–498.

Shikib Mehri and Maxine Eskenazi. 2021b. Schema-guided paradigm for zero-shot dialog. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 499–508.

Tom M Mitchell. 1980. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ....

Johannes EM Mosig, Shikib Mehri, and Thomas Kober. 2020. Star: A schema-guided dialog dataset for transfer learning. *arXiv preprint arXiv:2010.11853*.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. Soloist: Few-shot task-oriented dialog with a single pre-trained auto-regressive model. *arXiv preprint arXiv:2005.05298*.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 149–152.

Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126.

Konrad Scheffler and Steve Young. 2000. Probabilistic simulation of human-machine dialogues. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 2, pages II1217–II1220. IEEE.

Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. 2019. How to build user simulators to train rl-based dialog systems. *arXiv preprint arXiv:1909.01388*.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *arXiv preprint arXiv:2104.06644*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Improving Bot Response Contradiction Detection via Utterance Rewriting

Di Jin, Sijia Liu, Yang Liu, Dilek Hakkani-Tur
Amazon Alexa AI
{djinamzn,sijial,yangliud,hakkanit}@amazon.com

Abstract

Though chatbots based on large neural models can often produce fluent responses in open domain conversations, one salient error type is contradiction or inconsistency with the preceding conversation turns. Previous work has treated contradiction detection in bot responses as a task similar to natural language inference, e.g., detect the contradiction between a pair of bot utterances. However, utterances in conversations may contain co-references or ellipsis, and using these utterances as is may not always be sufficient for identifying contradictions. This work aims to improve the contradiction detection via rewriting all bot utterances to restore antecedents and ellipsis. We curated a new dataset for utterance rewriting and built a rewriting model on it. We empirically demonstrate that this model can produce satisfactory rewrites to make bot utterances more complete. Furthermore, using rewritten utterances improves contradiction detection performance significantly, e.g., the AUPR and joint accuracy scores (detecting contradiction along with evidence) increase by 6.5% and 4.5% (absolute increase), respectively.

1 Introduction

Latest chatbots powered by large pre-trained neural models have shown decent capabilities to maintain fluent and interesting conversations with human users (Paranjape et al., 2020; Roller et al., 2021; Bao et al., 2021; Konrád et al., 2021). However, they are still prone to various kinds of annoying mistakes (Xu et al., 2020; See and Manning, 2021; Xu et al., 2021). One such error is contradiction or inconsistency, as illustrated in Table 1.

In order to reduce contradiction errors, one approach is to develop a detection model to identify such problems after a system produces response candidates. To this end, Welleck et al. (2019) characterized the modeling of persona-related consistency as a natural language inference (NLI) problem and constructed a dialog NLI dataset based on Persona-Chat. To cover a broader range of consistency types (e.g., persona, logic, causality, etc), Nie et al. (2021) collected DECODE, a dataset containing human written dialogues with self-contradictory utterances. Besides the in-distribution human-human dialogues test set, they collected an out-of-distribution set containing dialogues between human and different chatbots. This human-bot test set can better evaluate models' performance in detecting contradiction in conversations between human and chatbots, which is the focus of this work.

We find one failure of the state-of-the-art (SOTA) contradiction detection model is due to the frequent anaphora and ellipses in chatbot utterances. One typical example is shown in Table 1, where the first bot utterance has an anaphor, "mine", and the last bot utterance misses an important entity, "Johnny Cash's concert". Such incomplete utterances would prevent detection models from fully understanding the bot utterances in the dialog, thus leading to detection errors. Therefore, we propose to first rewrite the bot utterances to recover all the missing information and then perform the contradiction detection task. To support this goal, we first collect a new dataset for incomplete utterance rewriting, which is a widely studied task (Pan et al., 2019; Su et al., 2019; Hao et al., 2021) but still lacks supporting datasets for open-domain conversations in English (Quan et al., 2019). Then we propose a rewriting model trained on this data to rewrite the anaphors to their corresponding entities and restore any missing content. We conduct experiments on the DECODE dataset (Nie et al., 2021), and demonstrate substantial performance improvement in contradiction detection when the utterance rewrite module is applied. Overall, we have made the following contributions in this work:

- We have collected a new English dataset for incomplete utterance rewriting for general open-domain conversations, and developed a rewrit-

Proceedings of the SIGdial 2022 Conference, pages 605–614
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Speakers	Original Dialogues	Rewritten Dialogues	Rewriting Type
Human:	Hi, what's your favorite singer?	—	
Bot:	**Mine** is johnny cash of course.	**My favorite singer** is Johnny Cash, of course.	Co-reference
Human:	He's amazing, I love his songs.	—	
Bot:	I never got to see johnny cash play but I wish I did.	I never got to see Johnny Cash play but I wish I did.	None
Human:	Same, I wish I could go to one of his concerts.	—	
Bot:	I have not been since last year though. I like sports.	I have not been **to Johnny Cash's concert** since last year though. I like sports.	Ellipsis

Table 1: Examples of human-bot conversations with contradictory bot utterances marked by red color. We rewrite every bot utterance to restore co-references and ellipsis (the restored parts are highlighted by bold font).

ing model for utterance restoration.

- With bot utterance rewriting, we can improve the previous best contradiction detection model by 6.5% in AUPR and 4.5% in joint accuracy that considers both contradiction and evidence labels.
- We relabeled the human-bot test set of the benchmark DECODE dataset and corrected some annotations.[1]

2 Contradiction Detection Method

2.1 Task Definition

We formalize dialogue contradiction detection as an NLI task. Given a list of utterances $x = \{u_1^H, u_1^B, ..., u_n^H, u_n^B\}$ representing a dialogue, the task is to determine if the last bot utterance u_n^B contradicts any previously conveyed information contained in the past bot utterances $\{u_1^B, ..., u_{n-1}^B\}$. Note that we are using human and bot alternating turns here (referred to as H and B), but they can be human-human conversations too. In addition to the binary label y, with 0 or 1 corresponding to the non-contradiction and the contradiction labels, respectively, we also output a set of indices $I \in \{1, ..., n-1\}$ representing the utterances in $\{u_1^B, ..., u_{n-1}^B\}$ that is actually contradicted by the last utterance u_n^B.

2.2 Detection Models

Based on the benchmark DECODE dataset, Nie et al. (2021) proposed two approaches for contradiction detection: an unstructured approach and a structured utterance-based (SUB) approach. The former one concatenates all the previous utterances in the dialogue history to form a single textual context. Then a classification model f_θ is applied to the context and the last utterance to infer the probability of contradiction. The latter SUB approach pairs every past bot utterance with the last one, and then feeds each pair to the classification model f_θ^{SUB}. The final contradiction probability is the maximum over all the outputs: $\hat{y} = max\{f_\theta^{SUB}(u_i^B, u_n^B) : i \in \{1, ..., n-1\}\}$. The supporting evidence (SE) for a contradiction decision contains the pairs having contradiction probability higher than a threshold η, i.e., $I = \{i : f_\theta^{SUB}(u_i^B, u_n^B) > \eta\}$. Nie et al. (2021) demonstrated that the latter SUB approach significantly outperforms the former one on the human-bot test set (more than 10% in accuracy). This SUB method is **the current SOTA model for contradiction detection**, which we adopted as one baseline.

2.3 Utterance Rewriting for Contradiction Detection

As discussed earlier, we noticed that many bot utterances contain co-references and ellipses and thus the baseline model fails to capture the semantic meaning or contradiction in the sentence pair. Therefore, we propose to first rewrite the bot utterances to restore co-references and ellipsis, and then feed the rewritten utterances (e.g., the dialogues on the right in Table 1) to the model. To this end, we first collect a new dataset specially for utterance rewriting and then develop a rewriting model.

Rewriting Data Collection To get parallel training data for utterance rewriting for open-domain conversations, we sub-sampled 6,000 and 4,000 dialogues from the DailyDialog (Li et al., 2017) and BST (Smith et al., 2020) datasets, respectively, as the training set. Besides, we sub-sampled 400 and 400 dialogues from DailyDialog and BST, respectively, as the test set. We only use the first six utterances in each dialog. Specifically, we use the first two utterances (from both speakers) as leading context and ask annotators to check the remaining four utterances, following Pan et al. (2019).[2] Empirically we find that the context information

[1] Code and data are released at: https://github.com/jind11/utterance-rewriting

[2] Utterance rewriting needs context to resolve co-references and ellipsis, and thus the first two utterances are not suitable for rewriting annotation.

needed to resolve co-references and ellipsis can always be found within 1-3 turns (Pan et al., 2019; Su et al., 2019). We ask annotators to identify whether an utterance is complete and can be understood without reading the context, and if not, then rewrite it to restore any missing information.

To ensure the annotation quality, we hired three in-house professional data annotators, who have been first trained via a pilot annotation session and then proceed to the official annotation phase after passing our provided qualification set. In the official annotation phase, two of them first worked independently and then the third annotator was tasked to make the adjudication over the two annotations and pick the best one or make revisions if needed. Besides, we periodically sampled 10% of the annotations from each annotator throughout the annotation process and provided feedback. The annotation is considered valid only when the accuracy of examined results surpassed 95% (we deem those rewrites that are both correct and complete as correct rewrites, and then calculate the percentage of correct rewrites as the accuracy). Overall, we have obtained 40,000 and 3,200 samples for training and testing, respectively.

Rewriting Model We treat rewriting as a sequence-to-sequence (Seq2Seq) task and adopt two pre-trained Seq2Seq models, T5 (Raffel et al., 2019) and Pegasus (Zhang et al., 2019). The input is the concatenated context utterances and the original last utterance, with special tokens inserted before each utterance to indicate its speaker.

3 Experiments

3.1 Contradiction Detection Data

We use the DECODE dataset (Nie et al., 2021) in this study. However, we found some issues with its human-bot test set: (1) Around one third of non-contradiction dialogues contain only one human and one bot utterances, which makes the detection task over-simplified, since there are no previous bot utterances. (2) Not every bot utterance has been annotated for contradiction with respect to its history. (3) Evidence is not labeled to indicate which history bot utterance contradicts the last one.

To resolve the above-mentioned issues, we curate new annotation using the dialogues in the original test set. Details of annotation procedures are provided in Section A of the Appendix. Overall, we have obtained 1,889 samples (453 positive samples and 1,436 negative ones), which we call an unbalanced set. Besides, we sub-sampled 453 negative samples and combined them with all the positive ones to form a balanced set. Table A.1 summarizes the data statistics. We will release this new test set.

3.2 Baselines

We compare the contradiction detection performance with and without rewriting bot utterances, all based on the same SUB model framework, which is the current SOTA model for contradiction detection. Another baseline we introduced is SUB-CONCAT, where each bot utterance is the concatenation of the original one with the preceding human utterance such that the missing information (coreference or ellipsis) can be recovered from the included previous utterance.

For rewriting, we compare our model against four strong baselines: one is the off-the-shelf SOTA co-reference resolution model trained on OntoNotes (named as "Co-reference") (Toshniwal et al., 2021; Wu et al., 2020), and the other three are developed based on three related datasets for rewriting, named as "CANARD" (Elgohary et al., 2019), "Gunrock" (Zhang et al., 2020), and "MuDoCo" (Tseng et al., 2021). Specifically, CANARD is a query rewriting dataset that aims to rewrite a query/question based on previous consecutive QA pairs for the conversational question answering task. The Gunrock dataset focuses on resolving ellipsis while containing a small portion of co-reference cases, and it consists of 1745 samples where all dialogues are in-house curated following the Alexa Prize competition format. The MuDoCo dataset is also for query rewriting for task-oriented dialogues covering 6 domains.

3.3 Evaluation Metrics

To evaluate incomplete utterance rewriting, we use both automatic and human evaluation. For human evaluation, we propose two metrics: (1) Correctness; (2) Completeness. The former one checks whether the rewriting part is correct and obeys the information in dialogue context, while the latter one checks whether the rewritten utterance is complete enough to be understood without reading the context. We have binary labels for both metrics and report the percentage of positive labels after human evaluation. For automatic evaluation, in addition to the widely used BLEU (Papineni et al., 2002), ROUGE-1 (R-1), and ROUGE-L (R-L) (Lin, 2004), we have added two more metrics specially for evaluating text editing models: exact match

(EM) accuracy, and the F_1 score, which was proposed in Pan et al. (2019) and focuses on n-grams that contain at least one restored word. Specifically, the n-gram restoration precision, recall, and F-score can be calculated as:

$$P_n = \frac{\{\text{restored n-grams}\} \cap \{\text{n-grams in ref}\}}{\{\text{restored n-grams}\}}$$

$$R_n = \frac{\{\text{restored n-grams}\} \cap \{\text{n-grams in ref}\}}{\{\text{n-grams in ref}\}}$$

$$F_n = 2 \cdot \frac{P_n * R_n}{P_n + R_n}$$

where "restored n-grams" refer to the n-grams in restored utterance that contain at least one restored words, and "n-grams in ref" refer to the n-grams in reference that contain at least one restored words.

For contradiction detection, we first set the threshold η to be 0.5, and report Precision/Recall/F1 for both the binary contradiction label and the support evidence labels, following Thorne et al. (2018).[3] Besides, we report Joint Accuracy, which indicates the performance when both the 2-way contradiction detection and the supporting evidence retrieval are correct. Considering that these scores are sensitive to η, we also report Area-under-Precision-Recall-Curve (AUPR) as a threshold-independent score.

3.4 Experimental Setup

For utterance rewriting, we have used three kinds of pre-trained models: T5-Base, T5-Large, and Pegasus-Large, whose parameter sizes are 220 M, 770 M, and 568 M. Each model is trained for 4 epochs with a learning rate of $5e^{-5}$, and beam search (beam size of 5) is used for generation.

For contradiction detection, following Nie et al. (2021), we used the RoBERTa-Large model whose parameter number is 330 M, which is trained for 3 epochs with a learning rate of $1e^{-5}$. We have used the Huggingface Transformer code base[4] and all experiments were run on Nvidia V100 GPUs.

4 Results and Discussion

4.1 Utterance Rewriting

We performed both automatic and human evaluation for utterance rewriting (please refer to Section 3.3 for evaluation details). Table 3 summarizes the automatic evaluation results. As can be seen, the three models perform similarly overall, with T5-Large slightly outperforming the other two. We thus adopt it as the main rewriting model in later experiments.

We also sub-sample 100 rewritten utterances by T5-Large for human evaluation. As shown in Table 4, the correctness and completeness scores for both test sets are above 85%, validating the high-quality of the rewriting model. We also report the change rate in the table that defines the percentage of the rewritten utterances that are different from the original ones (only differences in punctuation and upper/lower-case are not considered). The bottom block of Table 4 shows the percentage of utterances containing co-reference or ellipsis, or either, i.e, incomplete utterances. We see that **co-reference and ellipsis occur almost equally frequently** in incomplete utterances. Considering all the numbers together, we demonstrate that the rewriting model has covered most of those incomplete utterances.

4.2 Contradiction Detection

Table 2 compares the contradiction detection performance without rewriting and with rewriting by different rewriting models. First of all, the SUB-Concat method without rewriting does not yield any performance gain although it has included the context utterances. More importantly, after rewriting all bot utterances for both training and test sets, only our rewriting model can lead to significant improvements for all the evaluation metrics, while those baseline rewriting models either maintain or deteriorate the performance (we provided the rewriting performance of these baselines in Section B of Appendix for reference). We see that the AUPR metric has been improved by around 2.8% and 3.2% absolutely for the balanced and unbalanced sets by our model, respectively. We also implemented model ensemble where we rewrite bot utterances using our three rewriting models (T5-Base/Large and Pegasus-Large), run contradiction detection using each, and average their contradiction scores to obtain the final prediction. This further improves the detection performance over single models. Overall, we have achieved a substantial increase of 4.2% and 6.5% for AUPR and 4.5% and 3.4% for Joint-Acc. for the balanced and unbalanced sets, respectively.

4.3 Error Analysis

We conducted additional error analysis to understand the performance gains and remaining errors. We first obtained 95 false negative samples by the

[3] https://github.com/sheffieldnlp/fever-scorer

[4] https://github.com/huggingface/transformers/tree/master

Detection Method	Rewriting Model	Balanced Set				Unbalanced Set		
		P/R/F1	AUPR	SE (P/R/F1)	Joint-Acc.	P/R/F1	AUPR	Joint-Acc.
SUB-Bot only	None	89.4/70.6/78.9	89.0	90.4/62.9/74.2	69.9	73.2/70.6/71.9	75.4	81.4
SUB-Concat	None	88.1/66.9/76.0	88.1	90.0/60.0/72.0	68.1	66.3/66.9/66.6	71.6	78.7
SUB-Bot only	Co-reference	89.0/71.3/79.2	89.1	90.0/64.5/75.2	69.7	73.1/71.3/72.2	75.8	81.3
	CANARD	79.3/37.1/50.5	73.9	89.6/26.3/40.7	54.6	60.0/37.8/46.3	52.8	74.8
	Gunrock	79.2/59.8/68.2	74.0	88.5/50.2/64.0	60.0	53.0/59.8/56.2	49.0	71.9
	MuDoCo	88.1/65.3/75.0	87.4	91.6/59.5/72.1	67.9	70.6/65.3/67.9	71.2	80.2
SUB-Bot only	Ours (single)	90.9/**72.9**/80.9	91.8	93.0/**67.6**/**78.3**	73.6	73.5/**72.9**/73.2	78.6	82.8
	Ours (ensemble)	**92.9**/71.7/**81.0**	**93.2**	**93.9**/66.1/77.6	**74.4**	**80.1**/71.7/**75.7**	**81.9**	**84.8**

Table 2: Contradiction detection performance (%) on new human-bot test set. Best results for single model (T5-Large) are marked by underlines while overall best results are marked bold. 'SUB-Bot only' means feeding only bot utterances to the SUB model while SUB-Concat uses the concatenated bot utterance and the preceding human turn.

Models	BLEU	R-1	R-L	EM	F_1
T5-Base	0.653	0.822	0.801	0.213	0.402
T5-Large	0.653	0.820	0.798	0.199	0.422
Pegasus-Large	0.649	0.822	0.801	0.212	0.391
Agreement	0.714	0.840	0.837	0.323	0.309

Table 3: Automatic evaluation results for the rewriting model on the rewriting test set. Agreement is the inter-annotator agreement between two rewrites in test set.

Test Set	Correctness	Completeness	Change Rate
Rewriting	92.0	85.0	59.0
Contradiction	98.0	93.1	62.4
Test Set	**Co-reference**	**Ellipsis**	**Incomplete**
Rewriting	39.0	42.0	68.0
Contradiction	42.6	27.7	58.4

Table 4: Upper block: human evaluation of rewriting for both the rewriting and contradiction detection test sets (%); bottom block: percentage of utterances containing co-reference or ellipsis, or either (incomplete).

"SUB-Bot only" model without applying rewriting, and then manually identified 28 samples whose last bot utterances are incomplete. We then manually rewrote those incomplete bot utterances. With such manual rewriting, we are able to correctly classify 18 out of 28 samples to be positive (64.3% in accuracy), whereas, with the T5-Large rewriting model, 15 samples can be correctly predicted (53.6% in accuracy). This comparison indicates that our automatic rewriting has pushed the performance improvement close to the upper bound achieved by manual rewriting. More error analysis is provided in Section C of Appendix.

4.4 Why Utterance Rewriting Helps?

As illustrated by Table 1, in order to infer the entailment relationship between the premise (i.e. "Mine is johnny cash of course.") and hypothesis (i.e. "I have not been since last year though."), we need to resolve the anaphora and ellipses so that some key information can be restored, e.g., "Mine" is replaced by "My favorite singer" in the premise and the missing phrase of "to Johnny Cash's concert" is restored in the hypothesis. Without restoring such key information from the dialogue context, the contradiction detection model cannot fully understand the premise and hypothesis sentences, thus not being able to accurately detect contradictory cases. One could argue that we can simply concatenate the context with both premise and hypothesis respectively so that the detection model could grab the missing information itself from the context, however, the baseline method "SUB-Concat", which follows this setting, still under-performs the baseline without concatenating the context (i.e. SUB-Bot only). This indicates that when the premise and hypothesis are organized in a dialogue structure with multiple turns rather than as single-turn sentences, the NLI based detection model is not good at inferring their relationship anymore. Therefore, we need to use the utterance rewriting model to grasp the most necessary information from context and insert into the bot utterances so that we can still use the single-turn format while making up the missing information for entailment inference.

4.5 Future Work

We will keep improving the utterance rewriting model. Besides, we will showcase that utterance rewriting can also help improve other dialogue related tasks, such as task-orientated dialogue state tracking and response generation, open-domain dialogue response selection and generation, etc.

5 Conclusion

In this work, we aim to improve contradiction detection in chatbot utterances via rewriting to restore anaphora and ellipsis. To develop such an utterance rewriting model, we curated a dataset by crowd-sourcing and demonstrated that the rewriting quality is satisfactory. With such a rewriting technique, we are able to significantly improve the contradiction detection performance.

References

Siqi Bao, Huang He, Fan Wang, Hua Wu, Haifeng Wang, Wenquan Wu, Zhen Guo, Zhibin Liu, and Xinchao Xu. 2021. PLATO-2: Towards building an open-domain chatbot via curriculum learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2513–2525, Online. Association for Computational Linguistics.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924, Hong Kong, China. Association for Computational Linguistics.

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2021. RAST: Domain-robust dialogue rewriting as sequence tagging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4913–4924, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jakub Konrád, Jan Pichl, Petr Marek, Petr Lorenc, Van Duy Ta, Ondřej Kobza, Lenka Hỳlová, and Jan Šedivỳ. 2021. Alquist 4.0: Towards social intelligence using generative models and dialogue personalization. *arXiv preprint arXiv:2109.07968*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2021. I like fish, especially dolphins: Addressing contradictions in dialogue modeling. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1699–1713, Online. Association for Computational Linguistics.

Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. 2019. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1824–1833, Hong Kong, China. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ashwin Paranjape, Abigail See, Kathleen Kenealy, Haojun Li, Amelia Hardy, Peng Qi, Kaushik Ram Sadagopan, Nguyet Minh Phu, Dilara Soylu, and Christopher D Manning. 2020. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations. *arXiv preprint arXiv:2008.12348*.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. 2019. GECOR: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4547–4557, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Abigail See and Christopher Manning. 2021. Understanding and predicting user dissatisfaction in a neural generative chatbot. In *Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–12, Singapore and Online. Association for Computational Linguistics.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030, Online. Association for Computational Linguistics.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance ReWriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31, Florence, Italy. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and VERification (FEVER)

shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Shubham Toshniwal, Patrick Xia, Sam Wiseman, Karen Livescu, and Kevin Gimpel. 2021. On generalization in coreference resolution. In *Proceedings of the Fourth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 111–120, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bo-Hsiang Tseng, Shruti Bhargava, Jiarui Lu, Joel Ruben Antony Moniz, Dhivya Piraviperumal, Lin Li, and Hong Yu. 2021. CREAD: Combined resolution of ellipses and anaphora in dialogues. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3390–3406, Online. Association for Computational Linguistics.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. CorefQA: Coreference resolution as query-based span prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6953–6963, Online. Association for Computational Linguistics.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots. *arXiv preprint arXiv:2010.07079*.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2950–2968, Online. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Xiyuan Zhang, Chengxi Li, Dian Yu, Samuel Davidson, and Zhou Yu. 2020. Filling conversation ellipsis for better social dialog understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9587–9595.

A Contradiction Detection Data Collection

Considering that the original human-bot test set of the benchmark DECODE dataset is problematic, we specially curate new annotation based on those dialogues of the original test set via the following steps: (1) We first obtained 507 unique and full dialogues from the original human-bot test set[1] by merging dialogues with overlaps and removing dialogues of only one turn. We then obtained 1,889 partial dialogues for annotation by cutting each full dialogue from the beginning to each bot utterance so that we can annotate whether each bot utterance contradicts against its context. (2) In the first round of annotation, we ask three Amazon Mechanical Turk workers (from English-speaking countries, including USA, England, and Canada) to annotate both the binary label of contradiction and evidence indices that indicate which history bot utterance contradicts the last one. When setting-up the annotation interface, we have provided one line of guidance to warn annotators not to reveal any personal information during annotation. We keep those samples with three full votes as finalized samples and pass those without three equal votes to the second round. (3) In the second round, we provide the maximum set of evidence indices to another three AMT workers and let them verify and write down new annotation if they do not agree. Again, samples with three agreements are selected as finalized ones and those without are passed to authors of this work for final adjudication. Finally, among all the 1,889 samples, we have obtained 453 positive samples and 1,436 negative ones, which we call an unbalanced set. Besides, we have also sub-sampled 453 negative samples and combine them with all positive ones to form a balanced set. Table A.1 summarizes the data statistics.

Dataset	Positive	Negative	Type
Train	13,592	13,592	Human-Human
Balanced Test	453	453	Human-Bot
Unbalanced Test	453	1,436	Human-Bot

Table A.1: Statistics of the contradiction detection dataset.

B Rewriting Quality of Baselines

Table B.2 compares our rewriting model with baselines that are developed on three related datasets for utterance rewriting (CANARD, Gunrock, and MuDoCo) and we report performance on our rewriting test set. As expected, our rewriting model that is trained on our own rewriting dataset performs the best. And by combining Table 2 and Table B.2, we can see that better rewriting performance on our rewriting test set can lead to better contradiction detection performance.

Models	BLEU	R-1	R-L	EM	F_1
CANARD	0.363	0.631	0.623	0.105	0.215
Gunrock	0.352	0.626	0.612	0.008	0.223
MuDoCo	0.576	0.783	0.781	0.214	0.133
Ours	0.653	0.820	0.798	0.199	0.422

Table B.2: Automatic evaluation results for the rewriting model on the rewriting test set. Agreement is the inter-annotator agreement between two rewrites in test set.

C Qualitative Error Analysis

Among all 95 false negative samples predicted by the baseline, we find that the last bot utterances of 28 samples are incomplete and need rewriting for restoration. After automatic rewriting of all bot utterances, we can get 15 samples correct but still have 13 samples being false negatives. Now we would like to analyze the error pattern of these 13 false negatives after rewriting and we can categorize the errors into four types: numerical reasoning, logical reasoning, common sense reasoning, and hard to judge. Table C.3 provides examples for each type. Here are detailed definitions for these four types: (1) Numerical reasoning: Models need to do some numerical calculation or comparison to make a decision; (2) Logical reasoning: It requires logical reasoning for prediction; (3) Common sense reasoning: Some common sense knowledge needs to be equipped for reasoning; (4) Hard to judge: It is even hard for human to judge whether contradiction really exists or not.

Table C.4 provides several examples that are originally false negatives before rewriting bot utterances but later get correctly predicted after rewriting. As can be seen, the rewriting process can make up those critical information needed for detecting contradiction cases. Taking the first sample in Table C.4 as an example, there is ellipsis in the last two bot utterances in the original dialogue, which would lead to models' misunderstanding. After rewriting, the last two bot utterances get complete, which makes it much easier for models' decision making.

[1] https://github.com/facebookresearch/ParlAI/blob/main/projects/contradiction

Original Dialogue	Rewritten Dialogue	Error Type
human: Hi! **bot:** What do you do for a living? I'm retired, so I just sew all day. **human:** I'm a homemaker. What kind of things do you sew? **bot:** I sew a lot of quilts for my grandchildren. Do you have any kids? **human:** Aww that's sweet! Yes, I have one son. **bot:** How old is he? I have two daughters. They're a handful! **human:** He's four and yes they are a handful! How old are your daughters? **bot:** 4, 5, and 8. I love them to death, but they're exhausting!	**human:** Hi! **bot:** What do you do for a living? I'm retired, so I just sew all day. **human:** I'm a homemaker. What kind of things do you sew? **bot:** I sew a lot of quilts for my grandchildren. Do you have any kids? **human:** Aww that's sweet! Yes, I have one son. **bot:** How old is your son? I have two daughters. Your daughters are a handful! **human:** He's four and yes they are a handful! How old are your daughters? **bot:** My daughters are 4, 5 and 8. I love my daughters to death, but they're exhausting!	Numerical Reasoning
human: Hi! **bot:** Hi there! How are you? What do you do for a living? I'm a college student. **human:** I'm good, thankyou! I'm an electrician. What do you study? **bot:** It is in education. I just moved to an apartment near my job. What are you studying?	**human:** Hi! **bot:** Hi there! Are you okay? What do you do for a living? I'm a college student. **human:** I'm good, thankyou! I'm an electrician. What do you study? **bot:** I study education. I just moved to an apartment near my job. What are you studying?	Logical Reasoning
human: I've never left the US, I think I'm afraid to travel internationally though. **bot:** I can understand that. I have never been to the united states. I would love to though. **human:** Oh I thought you lived in the South United States. Where do you live? **bot:** I live on the south side of the state. It is very cold in the winter, though.	**human:** I've never left the US, I think I'm afraid to travel internationally though. **bot:** I can understand being afraid to travel internationally. I have never been to the united states. I would love to travel internationally though. **human:** Oh I thought you lived in the South United States. Where do you live? **bot:** I live on the south side of the state. It is very cold in the winter though.	Common Sense Reasoning
human: What did you do today? I have an adult son. **bot:** I went to the mall with my mom. She's my best friend. How old is your son? **human:** My son is 29. How many children do you have? **bot:** I have three kids. They're all grown up now, but they're still my best friends. **human:** Do your kids live with you? **bot:** Yes, they live with me and my husband. What do you like to do in your spare time? **human:** Have you been shopping lately? I love shopping. **bot:** No, I don't have much time for shopping. I spend most of my time with my kids.	**human:** What did you do today? I have an adult son. **bot:** I went to the mall with my mom. My mom is my best friend. How old is your son? **human:** My son is 29. How many children do you have? **bot:** I have three kids. My kids are all grown up now, but my kids are still my best friends. **human:** Do your kids live with you? **bot:** Yes, my kids live with me and my husband. What do you like to do in your spare time? **human:** Have you been shopping lately? I love shopping. **bot:** No, I don't have much time for shopping. I spend most of my time with my kids.	Hard to Judge

Table C.3: Examples that are wrongly predicted as negatives before and after automatic rewriting.

Original Dialogue	Rewritten Dialogue
human: Hi! **bot:** Hey! **human:** How are you today? **bot:** Great! And you? **human:** I am doing great as well! What do you do for a living? **bot:** I injured in an accident. **human:** When and what happened? **bot:** I had a sports injury. **human:** What kind of accident? **bot:** Broke an ankle. Very painful. **human:** What sport were you in? **bot:** I compete in rowing. **human:** How did you break an ankle from rowing? **bot:** A four wheeler accident.	**human:** Hi! **bot:** Hey! **human:** How are you today? **bot:** I am great today! And how are you? **human:** I am doing great as well! What do you do for a living? **bot:** I was injured in an accident. **human:** When and what happened? **bot:** I had a sports injury that caused me to be injured. **human:** What kind of accident? **bot:** I broke an ankle. My injury was very painful. **human:** What sport were you in? **bot:** I compete in rowing when I broke my ankle. **human:** How did you break an ankle from rowing? **bot:** I broke my ankle in a four wheeler accident.
human: Hi! **bot:** Hello, how are you? I'm a huge fan of star wars. How about you? **human:** I have never seen a Star Wars movie. **bot:** Neither have I. What do you like to do in your spare time? I like coffees.	**human:** Hi! **bot:** Hello, how are you? I'm a huge fan of star wars. How about you? **human:** I have never seen a Star Wars movie. **bot:** I have never seen a Star Wars movie. What do you like to do in your spare time? I like coffees.
human: Hi! **bot:** What do you like to do in your spare time? I like to run and play soccer. **human:** I like to work on cars in my spare time. Do you play any sports? **bot:** No, I don't. I'm not very athletic. What kind of cars do you work on?	**human:** Hi! **bot:** What do you like to do in your spare time? I like to run and play soccer in my spare time. **human:** I like to work on cars in my spare time. Do you play any sports? **bot:** No, I don't play any sports. I'm not very athletic. What kind of cars do you work on?

Table C.4: Samples that are wrongly predicted as negatives before rewriting but get corrected after rewriting.

Comparison of Lexical Alignment with a Teachable Robot in Human-Robot and Human-Human-Robot Interactions

Yuya Asano[1], Diane Litman[1,2,3], Mingzhi Yu[2], Nikki Lobczowski[3], Timothy Nokes-Malach[3,4], Adriana Kovashka[1,2], and Erin Walker[1,2,3]

[1]Intelligent Systems Program, University of Pittsburgh, USA
[2]Department of Computer Science, University of Pittsburgh, USA
[3]Learning Research and Development Center, University of Pittsburgh, USA
[4]Department of Psychology, University of Pittsburgh, USA
{yua17, dlitman, miy39, ngl13, nokes, aik85, eawalker}@pitt.edu

Abstract

Speakers build rapport in the process of aligning conversational behaviors with each other. Rapport engendered with a teachable agent while instructing domain material has been shown to promote learning. Past work on lexical alignment in the field of education suffers from limitations in both the measures used to quantify alignment and the types of interactions in which alignment with agents has been studied. In this paper, we apply alignment measures based on a data-driven notion of shared expressions (possibly composed of multiple words) and compare alignment in one-on-one human-robot (H-R) interactions with the H-R portions of collaborative human-human-robot (H-H-R) interactions. We find that students in the H-R setting align with a teachable robot more than in the H-H-R setting and that the relationship between lexical alignment and rapport is more complex than what is predicted by previous theoretical and empirical work.

1 Introduction and Related Work

Alignment is the convergence of behavior among speakers and plays an important role in designing the strategies of dialogue systems because it is associated with user engagement (Campano et al., 2015) and task success (Nenkova et al., 2008; Callejas et al., 2011; Lubold et al., 2018; Kory-Westlund and Breazeal, 2019). However, few studies have looked at how this relationship differs in multi-party versus dyadic task-oriented dialogues involving humans and a dialogue agent. This gap prevents us from inferring appropriate alignment strategies for dialogue agents across different group sizes.

Teachable agents act as peers that learners teach via dialogue. These agents have been shown to facilitate learning due to the effect of learning by teaching (Leelawong and Biswas, 2008) and the rapport the agents build with learners (Gulz et al., 2011). Inspired by theories suggesting that rapport is tied to verbal and non-verbal alignment (Lubold et al., 2019; Tickle-Degnen and Rosenthal, 1990), prior educational research has explored relationships between rapport with agents and various forms of alignment such as lexical (Rosenthal-von der Pütten et al., 2016; Lubold, 2018) and acoustic-prosodic (Lubold, 2018; Kory-Westlund and Breazeal, 2019) alignment.

While lexical alignment (the focus of this paper) in educational dialogue has been an active research area, prior studies are limited by 1) alignment measures (repetition of single words (Ai et al., 2010; Friedberg et al., 2012; Lubold, 2018) or manual annotations of semantics (Rosenthal-von der Pütten et al., 2016)) or 2) dialogue settings (they studied only dyadic interactions with an agent (Rosenthal-von der Pütten et al., 2016; Lubold, 2018; Sinclair et al., 2019), dyadic interactions between humans (Michel and Smith, 2017; Michel and Cappellini, 2019; Michel and O'Rourke, 2019; Sinclair and Schneider, 2021), or multi-party human interactions (Friedberg et al., 2012)). Multi-party interactions involving an agent remain to be explored with more sophisticated automated measures that can deal with the alignment of a sequence of words.

Therefore, we extend the past work on lexical alignment in educational dialogue in two ways. First, we view lexical alignment as initiation and repetition of shared lexical expressions, which are automatically extracted from dialogue excerpts and can consist of multiple words (Dubuisson Duplessis et al., 2021). Along with these metrics, we propose another viewpoint, *activeness*, which quantifies to what extent a speaker is involved in the establishment of shared expressions independent of their partner. Second, we investigate collaborative teaching where two learners co-teach a teachable NAO robot named Emma. We compare how individual

Proceedings of the SIGdial 2022 Conference, pages 615–622
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

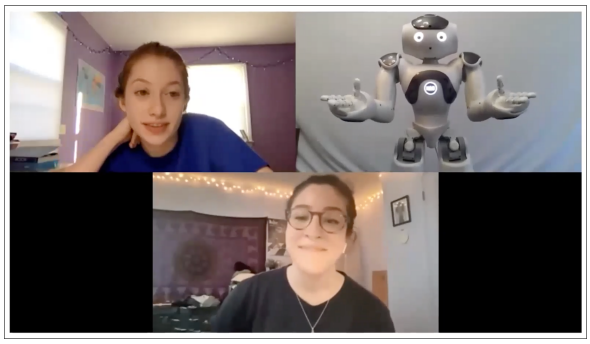


Figure 1: Screenshot of students and Emma in the H-H-R condition.

Mean (SD)	H-R (n=12)	H-H-R (n=26)
Utterances		
both speakers	174.8 (27.5)	59.1 (20.0)
student	81.3 (13.1)	27.7 (9.80)
Emma	93.5 (14.4)	31.5 (10.4)
Tokens		
both speakers	2919.0 (466.3)	1147.0 (388.1)
student	921.0 (228.7)	473.0 (227.2)
Emma	1998.0 (335.9)	673.0 (210.2)

Table 1: Descriptive statistics for H-R dialogues and the two Emma-student portions of H-H-R dialogues.

learners align with Emma and how alignment relates to rapport with her in this human-human-robot (H-H-R) setting versus in a one-on-one human-robot (H-R) setting. Although, outside of education, some researchers have also investigated H-H-R interactions (e.g., Kimoto et al., 2019), exploring alignment specifically in educational settings is useful because optimal alignment strategies differ from task to task (Dubuisson Duplessis et al., 2021). Through our comparisons, this paper provides the groundwork for designing different alignment strategies for teachable agents in the H-R and H-H-R settings.

2 Methodology

2.1 Data Collection

We recruited 40 undergraduate students from Pittsburgh, USA for an online study (due to COVID) over Zoom. Emma and the student(s) each had their own Zoom window (Figure 1) and conversed via speech. Students saw ratio word problems on a web application and taught them to Emma for 30 minutes. Each problem consisted of multiple steps, and students had to teach her step-by-step. Emma was designed to guide them by asking a question or making a statement relevant to their response even when they made a mistake. Her responses were pre-authored in Artificial Intelligence Markup Language and were selected based on pattern matching with students' utterances. All students were initially assigned to the H-H-R condition, but they were assigned to the H-R condition if their partner did not show up. We ended up with 12 students in the H-R condition and the remaining 28 in the H-H-R condition to form 14 pairs. In both conditions, students freely interacted with Emma by pressing and holding a "push to speak" button on the application. In the H-H-R condition, students were also expected to discuss the problems with their partners while teaching Emma, while, in the H-R condition, students had to keep talking to Emma without any discussions with others. An example H-H-R interaction can be found in Appendix A. We excluded one H-H-R pair from our analysis because one of the students did not talk to either Emma or the partner while working on the problems.

After teaching, learners individually answered survey questions about their perceived rapport with Emma on a six-point Likert scale, ranging from strongly disagree to strongly agree. The survey used four types of rapport measures created by Lubold (2018): general rapport measures (three items) based on the sense of connection from Gratch et al. (2007) and positivity, attention, and coordination rapport measures (four items each, twelve in total) from Sinha and Cassell (2015) and Tickle-Degnen and Rosenthal (1990). The latter twelve items had a higher Cronbach's α (.856) than the general rapport items (.839); thus, we used the average of the positivity, attention, and coordination items to create our rapport metric. The means and standard deviations of our rapport metric were 4.36 and .882 in the H-R condition, and 4.55 and .572 in the H-H-R condition. One-way ANOVA showed no effect of conditions on rapport ($F = .704, p = .407, df = 36$).

2.2 Computing Lexical Alignment

We manually transcribed all conversations, instead of using Emma's automated speech recognition, because she recorded only while students were holding the "push to speak" button. Then, because the measures of lexical alignment below are defined only for dyadic conversations, we manually identified the responder of each utterance in the H-H-R condition (see Appendix A) to split each conver-

sation into two Emma-student dialogues and one student-student dialogue. Table 1 describes the Emma-student dialogue data. Although individuals in the H-H-R condition spoke less to Emma than in the H-R condition due to the fixed experiment duration and the dialogue split, this does not affect our measures because they are normalized by the number of shared expressions or tokens.

The quantification of lexical alignment in the dialogues[1] in this paper relies on a *shared expression*, which is "a surface text pattern at the utterance level that has been produced by both speakers in a dialogue" (Dubuisson Duplessis et al., 2017). A shared expression is initiated by speaker S when used by S first and adapted (thus established as a shared expression) by the dialogue partner later. We used the alignment measures derived from shared expressions because mathematical expressions often consist of more than one token, other existing measures compute only repetition, and these measures are shown to be predictive of educational outcomes. Our ratio problems contained fractions and decimals, which cannot be expressed by one word. Indeed, the average lengths of shared expressions were $1.47 \pm .076$ and $1.44 \pm .101$ for the H-R and H-H-R conditions, respectively. Word-based measures such as counting (Nenkova et al., 2008; Friedberg et al., 2012; Wang et al., 2014), Spearman's correlation coefficient (Huffaker et al., 2006), regression models (Reitter et al., 2006; Ward and Litman, 2007), and vocabulary overlap (Campano et al., 2014) fail to represent the alignment of phrases containing more than one word. Other measures address this issue by leveraging n-grams (Michel and Smith, 2017; Duran et al., 2019) or cross-recurrence quantification analysis (Fusaroli and Tylén, 2016) but consider only repetitions in the alignment process as opposed to the measures used in this work (Dubuisson Duplessis et al., 2017, 2021). Furthermore, Sinclair and Schneider (2021) have found these measures are correlated with learning and collaboration between human students in collaborative learning.

We employed the set of *speaker-dependent* alignment measures out of the ones proposed by Dubuisson Duplessis et al. (2017, 2021)[2]: Initiated Expression (IE) and Expression Repetition (ER). IE of speaker S (IE_S) measures orientation (i.e., (a)symmetry) in the alignment process and is defined as $\frac{\text{\# expr. initiated by } S}{\text{\# expr.}}$. In a dialogue between speakers $S1$ and $S2$, the alignment process is symmetric if IE_S1 $\approx$ IE_S2 $\approx .5$ because IE_S1 + IE_S2 $= 1$. ER of speaker S (ER_S) captures the strength of repetition and is defined as $\frac{\text{\# tokens from } S \text{ in new or existing expr.}}{\text{\# tokens from } S}$.

However, IE cannot measure asymmetry or establishment independent of another speaker because, by definition, if IE_S1 increases, IE_S2 decreases. This dependence prevents us from observing increased establishment by both speakers. Therefore, we calculated Expression Initiator Difference (IED) (Sinclair and Schneider, 2021), which is given by IED $= |\text{IE\_S1} - \text{IE\_S2}|$. In addition, we propose a new measure:

Expression Establishment by Speaker S (EE_S) measures the *activeness* of S in the alignment process in terms of the establishment of new shared expressions. It is given by EE_S $= \frac{\text{\# tokens from } S \text{ used to establish new expr.}}{\text{\# tokens from } S}$.

In the example dialogue in Appendix A, there are ten shared expressions in the Emma-StudentA dialogue split: "that", "can you", "can"[3], "convert", "the", "days to", "days", "to", "hours", and "hours?". Of those, Emma started to use three expressions that Student A reused later: "that", "can you", and "can". Thus, IE_Emma $= \frac{3}{10}$ and IE_student $= \frac{7}{10}$. These are used to compute IED in the Emma-StudentA dialogue: IED $= |\frac{3}{10} - \frac{7}{10}| = \frac{2}{5}$. ER_student means the number of tokens in Student A's turns that are taken from Emma's previous turns and therefore parts of shared expressions (these tokens are italicized in Appendix A) divided by the total number of tokens Student A spoke to Emma including punctuations. Student A spoke 33 tokens to Emma and devoted four italicized tokens—"can you" and two "that"s—to shared expressions. Thus, for Student A, ER_student $= \frac{4}{33}$. Out of the four, Student A used three tokens to establish new shared expressions "that" and "can you", so EE_student $= \frac{3}{33} = \frac{1}{11}$.

2.3 Alignment Hypotheses

This study investigates the following hypotheses:

H1: Individuals in the H-H-R condition align less with Emma than in the H-R condition. Bren-

[1]The original dialogues in the H-R condition and the Emma-student dialogue splits in the H-H-R condition.

[2]We used the associated tool available at https://github.com/GuillaumeDD/dialign.

[3]Although the expression "can" is part of the longer expression "can you", it is counted as a shared expression because it appeared as a free form in Emma's last turn (Dubuisson Duplessis et al., 2017). I.e., it was not part of "can you".

nan and Clark (1996) formulated lexical alignment as the establishment of a shared conceptualization, a conceptual pact. In the H-R condition, individuals establish conceptual pacts only with Emma, but, in the H-H-R condition, individuals do so between them through discussion before talking to Emma (see the discussion between students before talking to her in Appendix A). This may mean these conceptual pacts are likely to be different from what Emma initially suggested because humans keep updating them, but Emma is not accessible to the updated conceptual pacts (in our case, Emma does not have an ability to intentionally align with humans). Therefore, individuals in the H-H-R condition may tend to use lexicons outside of shared expressions with Emma.

H2: Students feel more rapport with Emma when they align with Emma more (H2-a), she aligns with them more (H2-b), and alignment is more symmetric (H2-c). Human-human interactions show positive correlations between alignment and rapport (Lubold et al., 2019; Tickle-Degnen and Rosenthal, 1990; Sinha and Cassell, 2015). These are bi-directional; people feel a rapport when aligning with their partners and being aligned by their partners (Chartrand and van Baaren, 2009). In human-robot interactions, positive relationships between rapport and non-lexical alignments such as acoustic-prosodic (Lubold, 2018; Kory-Westlund and Breazeal, 2019) and movement (Choi et al., 2017) have also been found. We thus expect lexical alignment positively correlates with rapport in both conditions. We also anticipate a symmetric alignment process positively correlates with rapport because human-human interactions are more symmetric than human-agent ones (Dubuisson Duplessis et al., 2021) and past work increased rapport by imitating human alignment behavior.

H3: Lexical alignment is more strongly correlated with rapport with Emma in the H-R condition than in the H-H-R condition. As shown in Yu et al. (2019), Levitan et al. (2012), and Namy et al. (2002), the alignment process in H-H-R dialogues may also depend on other factors including the gender diversity of the party. Thus, in the H-H-R condition, lexical alignment alone may not be as predictive of rapport as in the H-R condition.

3 Results and Discussion

Individual alignment across H-R and H-H-R conditions (H1). We tested H1 by comparing means of ER_student and EE_student across conditions with one-way ANOVA. Table 2 partly supports H1. Individuals in the H-R condition repeated shared expressions (i.e., higher ER_student) more than in the H-H-R condition, but they were equally likely to establish shared expressions (i.e., no difference in EE_student) across conditions.

Mean (SD)	H-R (n=12)	H-H-R (n=26)
ER_student**	.594 (.052)	.462 (.077)
EE_student	.189 (.033)	.171 (.050)
ER_Emma**	.494 (.031)	.421 (.079)
EE_Emma*	.087 (.024)	.119 (.037)
IED	.155 (.111)	.158 (.127)

Table 2: Descriptive statistics of lexical alignment measures. Measures marked with * and ** are significantly different across conditions at $p < .05$ and $p < .01$ (two-tailed), respectively.

Correlations of alignment with rapport across conditions (H2 and H3). To test H2 and H3, first, we fit the regression equation with an interaction between the conditions and an alignment measure: $R = \beta_0 + \beta_1 * HHR + \beta_2 * A + \beta_3 * HHR * A$ where R is the rapport measure, A is an alignment measure, and HHR is 1 for students in the H-H-R condition; otherwise 0. Table 3 shows that β_3 is not significant for none of the alignment measures, meaning that the correlations between rapport and alignment are in the same direction regardless of the conditions.

Therefore, we used all data to compute Pearson's correlations between rapport and alignment (see Table 4). The significant negative correlation between rapport and IED supports H2-c. H2-b is not fully supported because, although EE_Emma is correlated positively with rapport, ER_Emma is not. In addition, surprisingly, we found evidence for the opposite of H2-a; EE_student has a negative correlation with rapport. Further analysis revealed IE_Emma is significantly negatively correlated with rapport ($r = -.490, p = .002$). This means students felt less rapport when they established more shared expressions relative to Emma and aligns with the findings on EE.

Finally, we compared Pearson's r between lexical alignment and rapport in the two conditions using Fisher transformation (Snedecor and Cochran, 1980) to test H3. It was not validated because there was no significant difference between the two conditions in Table 5.

Estimate of β_3 (p-value)	ER_student	EE_student	ER_Emma	EE_Emma	IED
Rapport	-0.54 (.901)	9.09 (.171)	3.10 (.652)	-0.83 (.928)	3.72 (.057)

Table 3: Coefficients of interaction terms (β_3).

Pearson's r (p-value)	ER_student	EE_student	ER_Emma	EE_Emma	IED
Rapport	-.315 (.054)	-.331* (.043)	.214 (.198)	.343* (.035)	-.573** (.000)

Table 4: Pearson's correlations between alignment measures and rapport. Correlations marked with * and ** are significant at $p < .05$ and $p < .01$ (2-tailed), respectively.

Pearson's r	H-R (n=12)	H-H-R (n=26)
ER_student	-.145	-.406
EE_student	-.457	-.285
ER_Emma	.008	.461
EE_Emma	.195	.407
IED	-.723	-.529

Table 5: Comparison of Pearson's correlations between alignment measures and rapport across conditions.

These results may be because perceived success in communication with Emma characterized by her accidental alignment leads to high rapport and low alignment by students. As Branigan et al. (2010) and Dubuisson Duplessis et al. (2017) reported, students might have (either consciously or unconsciously) expected they should establish shared expressions more than Emma due to her limited linguistic capacity. Thus, they might have started with an asymmetric alignment process. When Emma was stuck, they might have kept this strategy because they thought she did not understand them, resulting in decreased rapport. In contrast, as Emma established new shared expressions by accident, students might have thought she was following new information like humans, that she cared what they said, and that they were in sync, leading to more positivity, attention, and coordination rapport, respectively (Tickle-Degnen and Rosenthal, 1990). They may have also changed their alignment strategy to a more symmetric one (i.e., decreased alignment by students) that they usually use while interacting with humans.

3.1 Limitations

This study has several limitations. First, the limited number of participants (38 in total) might limit the detection of all correlations. Moreover, the comparison between the H-R and H-H-R conditions has low statistical power because the H-R condition had fewer than half of the participants in the H-H-R condition. It might have been biased because the assignment to the conditions was not fully random as well. Next, alignment measures may need contextual adjustments. For example, one math problem included both "three hours" and "three-fortieths of battery". Although "three" in these numbers refers to different entities, our measures saw it as a shared expression. Finally, some lexicons came from the problem prompt rather than the group conversation.

4 Conclusion

We examined relationships between lexical alignment and rapport with a teachable agent in one-on-one (H-R) and collaborative (H-H-R) teaching. Our methods expand prior literature by comparing alignment behavior in H-R and H-H-R settings and extending recent work by Dubuisson Duplessis et al. (2021) to the speaker-level act of *activeness* in the alignment process. Our results imply learners' lexical alignment with teachable agents may not always increase rapport with a teachable agent, unlike predictions from alignment theories (Lubold et al., 2019; Tickle-Degnen and Rosenthal, 1990) largely based on human-human interactions. Future work can expand our work by looking at the role of H-H portions of H-H-R interactions in their H-R portion and the effect of miscommunication as an intermediate variable on the negative correlations between rapport and learners' alignment and by extending the measures to multi-party settings without disentanglement.

Acknowledgements

We would like to thank anonymous reviewers for their thoughtful comments on this paper. This work was supported by Grant No. 2024645 from the National Science Foundation, Grant No. 220020483 from the James S. McDonnell Foundation, and a University of Pittsburgh Learning Research and Development Center internal award.

References

Hua Ai, Rohit Kumar, Dong Nguyen, Amrut Nagasunder, and Carolyn P. Rosé. 2010. Exploring the effectiveness of social capabilities and goal alignment in computer supported collaborative learning. In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems - Volume Part II*, ITS'10, page 134–143, Berlin, Heidelberg. Springer-Verlag.

Holly P. Branigan, Martin J. Pickering, Jamie Pearson, and Janet F. McLean. 2010. Linguistic alignment between people and computers. *Journal of Pragmatics*, 42(9):2355–2368. How people talk to Robots and Computers.

Susan E. Brennan and Herbert H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22:1482–1493.

Zoraida Callejas, David Griol, and Ramón López-Cózar. 2011. Predicting user mental states in spoken dialogue systems. *EURASIP Journal on Advances in Signal Processing*, 2011(1):1–21.

Sabrina Campano, Jessica Durand, and Chloé Clavel. 2014. Comparative analysis of verbal alignment in human-human and human-agent interactions. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4415–4422, Reykjavik, Iceland. European Language Resources Association (ELRA).

Sabrina Campano, Caroline Langlet, Nadine Glas, Chloé Clavel, and Catherine Pelachaud. 2015. An eca expressing appreciations. In *2015 International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 962–967.

Tanya L. Chartrand and Rick van Baaren. 2009. Chapter 5 human mimicry. volume 41 of *Advances in Experimental Social Psychology*, pages 219–274. Academic Press.

Mina Choi, Rachel Kornfield, Leila Takayama, and Bilge Mutlu. 2017. Movement matters: Effects of motion and mimicry on perception of similarity and closeness in robot-mediated communication. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, page 325–335, New York, NY, USA. Association for Computing Machinery.

Guillaume Dubuisson Duplessis, Chloé Clavel, and Frédéric Landragin. 2017. Automatic measures to characterise verbal alignment in human-agent interaction. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 71–81, Saarbrücken, Germany. Association for Computational Linguistics.

Guillaume Dubuisson Duplessis, Caroline Langlet, Chloé Clavel, and Frédéric Landragin. 2021. Towards alignment strategies in human-agent interactions based on measures of lexical repetitions. *Language Resources and Evaluation*, 55(2):353–388.

Nicholas Duran, Alexandra Paxton, and Riccardo Fusaroli. 2019. Align: Analyzing linguistic interactions with generalizable tech niques-a python library. *Psychological Methods*, 24(4):419–438.

Heather Friedberg, Diane Litman, and Susannah B. F. Paletz. 2012. Lexical entrainment and success in student engineering groups. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 404–409.

Riccardo Fusaroli and Kristian Tylén. 2016. Investigating conversational dynamics: Interactive alignment, interpersonal synergy, and collective task performance. *Cognitive science*, 40(1):145–171.

Jonathan Gratch, Ning Wang, Jillian Gerten, Edward Fast, and Robin Duffy. 2007. Creating rapport with virtual agents. In *Intelligent Virtual Agents*, pages 125–138, Berlin, Heidelberg. Springer Berlin Heidelberg.

Agneta Gulz, Magnus Haake, and Annika Silvervarg. 2011. Extending a teachable agent with a social conversation module: Effects on student experiences and learning. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, AIED'11, page 106–114, Berlin, Heidelberg. Springer-Verlag.

David Huffaker, Joseph Jorgensen, Francisco Iacobelli, Paul Tepper, and Justine Cassell. 2006. Computational measures for language similarity across time in online communities. In *Proceedings of the Analyzing Conversations in Text and Speech*, pages 15–22, New York City, New York. Association for Computational Linguistics.

Mitsuhiko Kimoto, Takamasa Iio, Michita Imai, and Masahiro Shiomi. 2019. Lexical entrainment in multi-party human–robot interaction. In *Social Robotics*, pages 165–175, Cham. Springer International Publishing.

Jacqueline M. Kory-Westlund and Cynthia Breazeal. 2019. Exploring the effects of a social robot's speech entrainment and backstory on young children's emotion, rapport, relationship, and learning. *Frontiers in Robotics and AI*, 6.

Krittaya Leelawong and Gautam Biswas. 2008. Designing learning by teaching agents: The betty's brain system. *Int. J. Artif. Intell. Ed.*, 18(3):181–208.

Rivka Levitan, Agustín Gravano, Laura Willson, Štefan Beňuš, Julia Hirschberg, and Ani Nenkova. 2012. Acoustic-prosodic entrainment and social behavior. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 11–19, Montréal, Canada. Association for Computational Linguistics.

Nichola Lubold. 2018. *Producing Acoustic-Prosodic Entrainment in a Robotic Learning Companion to Build Learner Rapport*. Ph.D. thesis, Arizona State University.

Nichola Lubold, Erin Walker, Heather Pon-Barry, and Amy Ogan. 2018. Automated pitch convergence improves learning in a social, teachable robot for middle school mathematics. In *Artificial Intelligence in Education*, pages 282–296, Cham. Springer International Publishing.

Nichola Lubold, Erin Walker, Heather Pon-Barry, and Amy Ogan. 2019. Comfort with robots influences rapport with a social, entraining teachable robot. In *Artificial Intelligence in Education*, pages 231–243, Cham. Springer International Publishing.

Marije Michel and Marco Cappellini. 2019. Alignment during synchronous video versus written chat l2 interactions: A methodological exploration. *Annual Review of Applied Linguistics*, 39:189–216.

Marije Michel and Breffni O'Rourke. 2019. What drives alignment during text chat with a peer vs. a tutor? insights from cued interviews and eye-tracking. *System*, 83:50–63. Special Edition on the Work of Professor Stephen Bax.

Marije Michel and Bryan Smith. 2017. *Measuring lexical alignment during L2 chat interaction: An eye-tracking study*, pages 244–267. Taylor and Francis.

Laura L. Namy, Lynne C. Nygaard, and Denise Sauerteig. 2002. Gender differences in vocal accommodation:: The role of perception. *Journal of Language and Social Psychology*, 21(4):422–432.

Ani Nenkova, Agustín Gravano, and Julia Hirschberg. 2008. High frequency word entrainment in spoken dialogue. In *Proceedings of ACL-08: HLT, Short Papers*, pages 169–172, Columbus, Ohio. Association for Computational Linguistics.

David Reitter, Frank Keller, and Johanna D. Moore. 2006. Computational modelling of structural priming in dialogue. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 121–124, New York City, USA. Association for Computational Linguistics.

Astrid M. Rosenthal-von der Pütten, Carolin Straßmann, and Nicole C. Krämer. 2016. Robots or agents – neither helps you more or less during second language acquisition. In *Intelligent Virtual Agents*, pages 256–268, Cham. Springer International Publishing.

Arabella Sinclair, Kate McCurdy, Christopher G Lucas, Adam Lopez, and Dragan Gaševic. 2019. Tutorbot corpus: Evidence of human-agent verbal alignment in second language learner dialogues. *International Educational Data Mining Society*.

Arabella J Sinclair and Bertrand Schneider. 2021. Linguistic and gestural coordination: Do learners converge in collaborative dialogue?. *International Educational Data Mining Society*.

Tanmay Sinha and Justine Cassell. 2015. We click, we align, we learn: Impact of influence and convergence processes on student learning and rapport building. In *Proceedings of the 1st Workshop on Modeling INTERPERsonal SynchrONy And InfLuence*, INTERPERSONAL '15, page 13–20, New York, NY, USA. Association for Computing Machinery.

G.W. Snedecor and W.G. Cochran. 1980. *Statistical Methods*, seventh edition. Iowa State University Press.

Linda Tickle-Degnen and Robert Rosenthal. 1990. The nature of rapport and its nonverbal correlates. *Psychological Inquiry*, 1(4):285–293.

Yafei Wang, David Reitter, and John Yen. 2014. A model to qualify the linguistic adaptation phenomenon in online conversation threads: Analyzing priming effect in online health community. In *Proceedings of the Fifth Workshop on Cognitive Modeling and Computational Linguistics*, pages 55–62, Baltimore, Maryland, USA. Association for Computational Linguistics.

Arthur Ward and Diane J Litman. 2007. Automatically measuring lexical and acoustic/prosodic convergence in tutorial dialog corpora.

Mingzhi Yu, Diane Litman, and Susannah Paletz. 2019. Investigating the relationship between multi-party linguistic entrainment, team characteristics and the perception of team social outcomes. In *The Thirty-Second International Flairs Conference*.

A Sample Dialogue and Lexical Alignment

Speaker	Utterance	Responder
Emma:	**Now that I know how long one battery will last, can you help me figure out how many batteries I need total?**	**Student A**
Student B:	Oh, okay.	Student A
Student A:	Okay. Next ...	Student B
Student B:	Okay.	Student A
Student A:	So one whole battery lasts three and three quarters of an hour.	Student B
Student B:	Three and three quarters of an hour. Oh my gosh. Emma, you're making this difficult on us.	Student A
Student A:	I think we need her... Oh man.	Student B
Student B	Because it's in days now. So she has to figure out...	Student A
Student A:	You have to do like dimensional analysis.	Student B
Student B	I think she... Yeah. She has to convert days to hours. I think that might be the easiest thing for her.	Student A
Student A:	Okay.	Student B
Student B:	So she has to divide...	Student A
Student A:	Wait, is she going to remember that?	Student B
Student B:	Oh, I don't know.	Student A
Student A:	Okay. I'm going to ask if she knows how to convert days to hours.	Student B
Student B:	Okay.	Student A
Student A:	**Okay, Emma. *Can you* convert the number of days to the number of hours?**	**Emma**
Emma:	**So I know how long I'll be gone in *days*, but how long *the* battery lasts is in *hours*. So first I should change *the days to hours?***	**Student A**
Student A:	**Yes, Emma. *That*'s correct.**	**Emma**
Emma:	**So I *can convert* two and three quarters *to* an improper fraction, eleven over four. And then I *can* multiply it by twenty four *hours?***	**Student A**
Student B:	She did all the work.	Student A
Student A:	**Yes, Emma. *That*'s correct.**	**Emma**

Table 6: Lexical alignment in the Emma-studentA portion of the dialogue (bolded utterances). A responder is a speaker who responded to the utterance. Speakers initiated colored but not italicized expressions and repeated the italicized ones. Contractions were tokenized as two tokens (e.g., That's to "That" and "'s"). Punctuation was treated as one token but did not constitute a shared expression by itself.

TREND: Trigger-Enhanced Relation Extraction Network for Dialogues

Po-Wei Lin Shang-Yu Su Yun-Nung Chen
National Taiwan University, Taipei, Taiwan
{r09922a24, f05921117}@csie.ntu.edu.tw y.v.chen@ieee.org

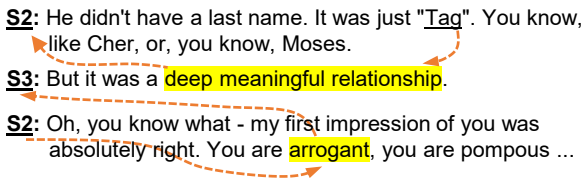
Abstract

The goal of dialogue relation extraction (DRE) is to identify the relation between two entities in a given dialogue. During conversations, speakers may expose their relations to certain entities by explicit or implicit clues, such evidences called "triggers". However, trigger annotations may not be always available for the target data, so it is challenging to leverage such information for enhancing the performance. Therefore, this paper proposes to learn how to identify triggers from the data with trigger annotations and then transfers the trigger-finding capability to other datasets for better performance. The experiments show that the proposed approach is capable of improving relation extraction performance of unseen relations and also demonstrate the transferability of our proposed trigger-finding model across different domains and datasets.[1]

1 Introduction

The goal of relation extraction (RE) is to identify the semantic relation type between two mentioned entities from a given text piece, which is one of basic and important natural language understanding (NLU) problems (Zhang et al., 2017; Zhou and Chen, 2021; Cohen et al., 2020). In this task setting, we are usually given a written sentence and a query pair containing two entities and asked to return the most possible relation type from a predefined set of relations. Dialogue relation extraction (DRE), on the other hand, aims to excavate underlying cross-sentence relation in natural human communications (Yu et al., 2020; Jia et al., 2021). The problem itself is well-motivated, because relations between entities in dialogues could potentially provide dialogue systems with additional features for better dialogue managing (Peng et al., 2018; Su et al., 2018a) or response generation (Su et al., 2018b).

[1]The source code is available at: http://github.com/MiuLab/TREND.



Arguments	Trigger	Relation
(Tag, S2)	a deep meaningful relationship	per:girl/boyfriend
(S2, S3)	arrogant	per:negative_impression

Figure 1: An example of dialogue relation extraction; the dashed arrows connect subjects, triggers, and objects. Triggers are clues of relations annotated in DialogRE.

There are two popular datasets, DialogRE (Yu et al., 2020) and DDRel (Jia et al., 2021), focusing on relation extraction in dialogues illustrated in Figure 1. In DRE, given a conversation and a query pair, we aim to identify the interpersonal relationship between the given entities, where entities can be human or other types like locations. As shown in Figure 1, the evidences of relations within the conversation flow, called **Triggers**, provide informative cues for this task. A trigger can be a short phrase or even a single word with any possible part-of-speech. In the example, the clue for knowing the speaker 2 has a negative impression on the speaker 3 comes from the sentence "*You are arrogant.*" Such hint is intuitively useful for deciding the relations. However, Albalak et al. (2022) is the only prior work that tried to explicitly leverage such signal for improving DRE, because such explanation annotations may not be always available (Kung et al., 2020).

Prior work can be divided into two main lines, one of which is graph-based methods. DHGAT (Chen et al., 2020) presents an attention-based heterogeneous graph network to model multiple types of features; GDPNet (Xue et al., 2021) constructs latent multi-view graphs to model possible relationships among tokens in a long sequence, and then refines the graphs by iterative graph convo-

Proceedings of the SIGdial 2022 Conference, pages 623–629
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

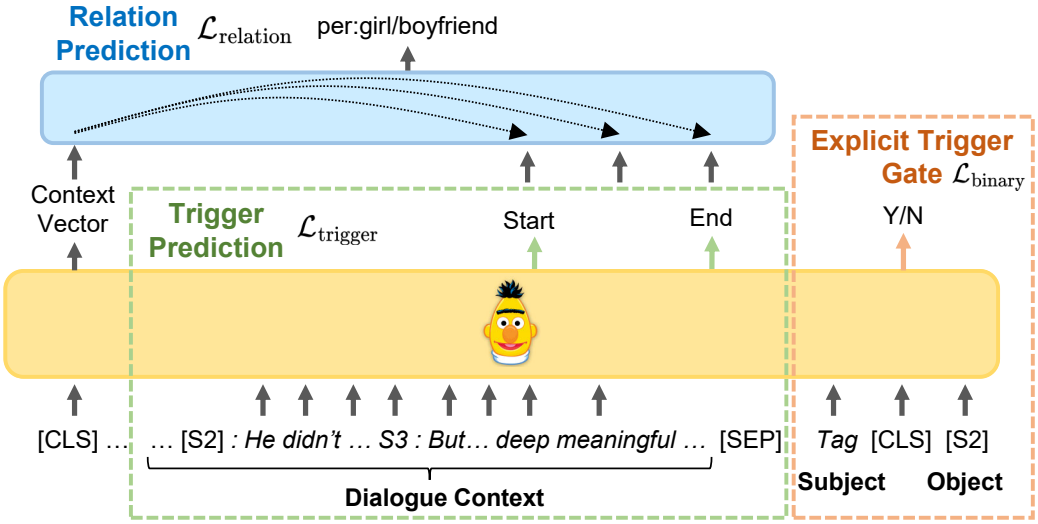


Figure 2: The proposed method contains two components: (1) a multi-tasking BERT with two fine-tuning tasks (explicit trigger classification and trigger prediction), and (2) a relation predictor with attentional feature fusion.

lution and pooling techniques. Another branch is BERT-based (Kenton and Toutanova, 2019) methods (Yu et al., 2020; Xue et al., 2022). SimepleRE (Xue et al., 2022) is a simple BERT model with an additional refinement gate for iteratively finding high-confidence prediction. LSR (Nan et al., 2020) is a latent structure refinement method for better reasoning in the document-level relation extraction task. Although it is known that using trigger information can significantly help the performance of relation extraction, only DialogRE has the annotated triggers. It is not guaranteed that utilizing the annotated triggers can generalize to other relations from other datasets, considering the discrepancy of their relation types.

Given the target data without trigger annotations, this paper proposes **TREND**, a simple multi-tasking model with an attentional relation predictor, where it learns the general capability of finding triggers and transfers it to the unseen relations for performance improvement. The experiments show that our proposed method can effectively identify the explicit triggers and generalize to unseen relations towards great flexibility and practicality.

2 Proposed Method

The core idea of this model is to identify trigger spans and accordingly leverage such signal to improve relation extraction. We hereby propose **T**rigger-enhanced **R**elation-**E**xtraction **N**etwork for **D**ialogues, **TREND**, illustrated in Figure 2.

2.1 Problem Formulation

Given a piece of dialogue context $\mathcal{D}$ composed of text tokens $\mathcal{D} = \{x_i\}$ and a query pair q containing a subject entity and an object entity $q = (s, o)$, the task aims at learning a function f that finds the most possible relations between the given entities from a predefined relation set $\mathcal{R}$, $f(\mathcal{D}, q) \rightarrow \mathcal{R}$. Note that a single query pair may contain multiple relations, and we duplicate the data samples when they have multiple relation labels by following the prior work.

2.2 TREND

The proposed model has two modules, (1) a multi-tasking BERT (Kenton and Toutanova, 2019) for encoding context and identifying triggers, and (2) a relation predictor with a feature fusion of the dialogue and the automatically identified trigger.

As illustrated in Figure 2, an input $(\mathcal{D}, q)$ will be first augmented into a BERT-style sequence. Specifically, the input format is "`[CLS]` $\mathcal{D}$ `[SEP]` s `[CLS]` o". We replace the target entity pair with their speaker tokens in $\mathcal{D}$ following Yu et al. (2020) illustrated in the figure. The first `[CLS]` encodes the dialogue contexts, and the second one is to predict whether the triggers are explicit via binary classification detailed below.

Explicit Trigger Gate Because triggers sometimes are implicit, it is difficult to identify the associated trigger spans of dialogue relations. We hereby propose to learn a binary classifier as a gate to identify if the explicit triggers exist, and empty trigger spans are inputted to relation prediction

when no explicit triggers. The binary cross entropy loss $\mathcal{L}_{\text{binary}}$ is used here.

Trigger Prediction The explicit triggers are identified by a extractive method with start-end pointer prediction (Kenton and Toutanova, 2019), which is prevalent in extractive question answering (Lee et al., 2016; Rajpurkar et al., 2016). This is a single-label classification problem of predicting the most possible positions; hence a cross entropy loss $\mathcal{L}_{\text{trigger}}$ is conducted.

Relation Prediction A learned context vector and a predicted trigger span are then fed into the relation predictor as depicted in the top part of Figure 2. The features are fused by a generic attention mechanism, where the query is the context vector $\mathbf{c}$, and the keys and the values are trigger words $\mathbf{x_i}$ encoded by BERT: $\sum \text{softmax}(\mathbf{c} \cdot \mathbf{x_i}) \cdot \mathbf{x_i}$. The merged feature is then fed into a 1-layer feed-forward network for final relation prediction using a cross entropy loss $\mathcal{L}_{\text{relation}}$.

Supervised Joint Learning Considering that only DialogRE contains the annotated trigger cues, we perform supervised joint learning for three above tasks. Three above losses are linearly combined as the learning objective for training the whole model in an end-to-end manner. The weights for adjusting the impact of each loss are tuned in the development set. We also apply schedule sampling (Bengio et al., 2015) on explicit trigger classification and trigger prediction when feeding into the relation predictor in order to mitigate the gap between the true triggers and the predicted ones.

Transfer Learning Because annotated triggers may not be available, this paper focuses on transferring the trigger-finding capability to another target dataset, DDRel, which does not contain trigger annotations and its relation types differ a lot from DialogRE. We replace the final feed-forward layer with a new one, since relation numbers may different in two datasets. Then we fine-tune the whole model using a single loss about relation prediction, $\mathcal{L}_{\text{relation}}$, where we assume the trigger-finding capability can be better transferred cross different datasets/relations.

3 Experiments

We focus on evaluating the performance of DRE on the dataset without trigger labels in order to investigate if the trigger-finding capability can be transferred across datasets/relations.

Model	F1
BERT	60.6
GDPNet	64.3
SimpleRE (single entity pair)	60.4
D-REX$_{\text{BERT}}$	59.2
TUCORE-GCN$_{\text{BERT}}$	65.5
TREND$_{\text{BERT-Base}}$	**66.8**
TREND$_{\text{BERT-Large}}$	**67.8**
SimpleRE (multiple entity pairs)	66.7
SocAoG (multiple entity pairs)	*69.1*
TREND$_{\text{BERT-Base}}$ (ground-truth triggers)	75.3

Table 1: The model performance on DialogRE.

3.1 Setting

The DRE datasets used in our experiments are DialogRE (v2) with trigger annotations (Yu et al., 2020) and DDRel (Jia et al., 2021) without trigger annotations. Text normalization like lemmatization and expanding contractions is applied to data preprocessing. In all experiments, we use mini-batch `adam` with a learning rate $3e{-}5$ as the optimizer on Nvidia Tesla V100. The ratio of teacher forcing and other hyper-parameters are selected by grid search in (0,1] with a step 0.1. The training takes 30 epochs without early stop. The detailed implementation can be found in Appendix A.

The following BERT-based methods are performed for fair comparison: 1) BERT, 2) GDPNet (Xue et al., 2021), 3) SimpleRE (Xue et al., 2022), 4) D-REX$_{\text{BERT}}$ (Albalak et al., 2022), and 5) TUCORE-GCN$_{\text{BERT}}$ (Lee and Choi, 2021). Other approaches that take multiple entity pairs for global consideration cannot directly be compared with TREND but reported as reference.

3.2 Results of Supervised Joint Learning

The performance of our TREND model jointly trained on the trigger-available DialogRE dataset is presented in Table 1, where it is obvious that our TREND achieves the best performance in the fair setting. Unlike SimpleRE and GDPNet that need to iteratively refine the latent features or latent graphs, relation prediction in the proposed TREND is straight-forward, making training and inference efficient and robust. Furthermore, D-REX (Albalak et al., 2022) also leverages triggers for relation prediction but performs significantly worse than our simple TREND models in the same setting. Our trained binary gate has about 85% accuracy while the trigger prediction has no more than 50% of exact match. Although our model cannot per-

Model	4-class		6-class		13-class	
	Acc	Macro-F	Acc	Macro-F	Acc	Macro-F
BERT	47.1 / 58.1	44.5 / 52.0	41.9 / 42.3	39.4 / 38.0	39.4 / 39.7	20.4 / 24.1
TUCORE-GCN$_{\text{BERT}}$	43.8 / 60.3	41.9 / 56.6	36.9 / **52.6**	38.7 / 54.2	29.5 / 44.9	20.5 / **36.9**
TREND$_{\text{BERT-Base}}$	51.5 / **65.4**	**46.5 / 61.2**	40.3 / **52.6**	**43.0 / 55.0**	**40.5 / 46.2**	21.2 / 34.7
w/o binary gate	52.5 / 53.8	45.3 / 49.7	37.0 / 43.6	41.8 / 45.9	36.6 / 43.6	**26.4** / 36.3
TREND$_{\text{BERT-Large}}$	**51.6** / 60.3	**46.5** / 54.0	**42.5** / 46.2	**43.0** / 48.2	34.4 / 43.6	19.9 / 36.3
w/o binary gate	41.5 / 47.4	40.3 / 44.9	39.0 / 42.3	43.1 / 42.9	38.5 / 34.6	17.3 / 21.1

Table 2: The DDRel performance in session-level/pair-level settings and different granularity settings (4,6,13-class).

fectly extract the triggers, the predicted spans can still facilitate relation prediction in our proposed TREND. It demonstrates that our TREND model is capable of identifying potential triggers and utilizing such cues for predicting relations. Note that TREND$_{\text{BERT-Large}}$ is for reference, indicating that a larger model has the potential of further improving the performance. The upper-bound of our proposed TREND$_{\text{BERT-Base}}$ is 75.3 shown in the last row of Table 1, where the ground truth triggers are inputted in the relation predictor. This higher score suggests that our TREND model still has a room for improvement and the proposed model design is well validated.

3.3 Results of Transfer Learning

Due to the lack of trigger annotations in DDRel, our TREND model focuses on transferring the trigger-finding capability learned from DialogRE to DDRel. We compare our proposed TREND with two models, which are not designed for transferring across different relation extraction datasets, so they are directly trained on the DDRel data. Table 2 presents the performance achieved on DDRel evaluated in session-level and pair-level settings, where session-level relation extraction is given a *partical* dialogue the entity pair is involved in and pair-level is based on a *full* dialogue (Jia et al., 2021).[2] All scores are much lower than ones in DialogRE due to the higher difficulty of this dataset. The obtained improvement compared with the BERT baseline is larger when the longer dialogue contexts as the input; that is, pair-level improvement is more than session-level one. The probable reason is that extracting key evidences for predicting relations is more important to overcome information overload.

Furthermore, we report the performance of the current state-of-the-art (SOTA) relation extraction model, TUCORE-GCN, on the DDRel dataset.[3] It can be found that our proposed method can effectively transfer the capability of capturing triggers from DialogRE to DDRel, and outperform TUCORE-GCN in most cases, achieving a new SOTA performance in DDRel.

Suprisingly, TREND$_{\text{BERT-Large}}$ does not outperform TREND$_{\text{BERT-Base}}$, implying that TREND$_{\text{BERT-Base}}$ already has enough good capability of capturing triggers and can generalize to another dataset (DDRel) and a new relation set.

3.4 Ablation Study

Because our trigger finding module contains a binary classifier deciding the existence of explicit triggers and a trigger predictor extracting trigger spans, we examine the effectiveness of the binary gate. By removing the binary gate, the performance is consistently degraded shown in Table 2, further demonstrating the effectiveness of the designed trigger-finding module in our TREND model.

3.5 Generalization of Unseen Relations

To further investigate if our trigger-finding capability can generalize to different relations, we categorize all relations into seen and unseen relations based on the relation similarity between the two datasets shown in Table 3, and show the session-level performance in Table 4. It can be seen that our proposed TREND is capable of transferring trigger-finding capability from DialogRE to DDRel, even DDRel does not contain trigger annotations. More importantly, our learned trigger-finding capability is demonstrated general to diverse relations, because TREND achieves better results for not only seen but also unseen relations whose triggers never appear in the DialogRE data. We qualitatively analyze the predicted triggers of unseen relations, where TREND extracts a dirty word ("fxxk") and a

[2] A session only contains multiple turns in a dialogue, so session-level results are worse than pair-level ones.

[3] The numbers are obtained based on the released code in Lee and Choi (2021).

DDRel Relation	DialogRE Relation
Workplace Superior-Subordinate	per:boss
Workplace Superior-Subordinate	per:subordinate
Friends	per:friends
Lovers	per:girl/boyfriend
Neighbors	per:neighbor
Roommates	per:roommate
Child-Parent	per:children
Child-Other Family Elder	per:other family
Siblings	per:siblings
Spouse	per:spouse
Colleague/Partners	per:works
Courtship	-
Opponents	-
Professional Contact	-

Table 3: Relation ontology mapping between DDRel and DialogRE datasets.

DDRel Relation	Seen	Unseen
BERT	23.77	9.94
TUCORE-GCN	23.39	10.81
TREND	28.30	13.13

Table 4: F1 results of DDRel seen and unseen relations.

word "client" as triggers for unseen relations "opponent" and "professional contact" in DDRel respectively. The full samples can be found in Table 6. It shows the effectiveness and generalizability of our proposed TREND model towards practical usage.

3.6 Qualitative Study

The predicted triggers and relation for DialogRE and DDRel datasets are presented in Table 5 and Table 6 respectively. Note that the triggers are not annotated in DDRel. It can be found that TREND can extract explicit cues as triggers not only for the seen relations, which are similar to relations in DialogRE, but also unseen ones.

4 Conclusion

This paper proposes TREND, a multi-tasking model with the generalizable trigger-finding capability, to improve dialogue relation extraction. TREND is a simple, flexible, end-to-end model based on BERT with three components: (1) an explicit trigger gate for trigger existence, (2) an extractive trigger predictor, and (3) a relation predictor with an attentional feature fusion. The experiments demonstrate that TREND can successfully transfer the learned trigger-finding capability across different datasets and diverse relations for better dialogue relation extraction performance, showing the great potential of improving explainability without rationale annotations.

S1: What's up? S2: Monica and I are **engaged**. S1: Oh my God. Congratulations. S2: Thanks.		
Argument	**Relation**	**Trigger**
(S2, Monica)	girl/boyfriend	engaged

Table 5: A predicted result of TREND on DialogRE.

S1: That's all. S2: That's all?! S1: You don't see it, do you, **father**?" S2: No. Fellow wants to sell a house ...		
Argument	**Relation (Seen)**	**Trigger**
(S1, S2)	Child-Parent	father
S1: **Fuck** me! S2: Want a drink? Okay... I'm not good at this sort of thing, but we don't have a lot of time, so I'll just go ahead and get started.		
Argument	**Relation (Unseen)**	**Trigger**
(S1, S2)	Opponent	fuck
S1: I'm Joe Galvin, I'm representing Deborah Ann Kaye, case against St. S2: I told the guy I didn't want to talk to... S1: I'll just take a minute. Deborah Ann Kaye. You know what I'm talking about. S2: No. S1: He's the Assistant Chief of Anesthesiology, Massachusetts Commonwealth. He says your doctors, Towler and Marx, put my girl in the hospital for life. And we can prove that. What we don't know is why. I want someone who was in the O.R. S2: I've got nothing to say to you. S1: You know what happened. S2: Nothing happened. S1: Then why aren't you testifying for their side? I can subpoena you, you know. I can get you up there on the stand. S2: And ask me what? S1: Who put my **client** in the hospital for life. S2: I didn't do it, Mister. S1: Who are you protecting, then? S2: Who says that I'm protecting anyone? S1: I do. Who is it? The Doctors. What do you owe them? S2: I don't owe them a goddamn thing. S1: Then why don't you testify? S2: You know, you're pushy, fella... S1: You think I'm pushy now, wait 'til I get you on the stand... S2: Well, maybe you better do that, then.		
Argument	**Relation (Unseen)**	**Trigger**
(S1, S2)	Professional Contact	client

Table 6: Predicted results of TREND on DDRel.

Acknowledgements

We thank reviewers for their insightful comments and Ze-Song Xu for running baselines. This work was financially supported from Google and the Young Scholar Fellowship Program by Ministry of Science and Technology (MOST) in Taiwan, under Grants 111-2628-E-002-016 and 111-2634-F-002-014.

References

Alon Albalak, Varun Embar, Yi-Lin Tuan, Lise Getoor, and William Yang Wang. 2022. D-REX: Dialogue relation extraction with explanations. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 34–46.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Hui Chen, Pengfei Hong, Wei Han, Navonil Majumder, and Soujanya Poria. 2020. Dialogue relation extraction with document-level heterogeneous graph attention networks. *arXiv preprint arXiv:2009.05092*.

Amir DN Cohen, Shachar Rosenman, and Yoav Goldberg. 2020. Relation classification as two-way span-prediction. *arXiv preprint arXiv:2010.04829*.

Qi Jia, Hongru Huang, and Kenny Q Zhu. 2021. Ddrel: A new dataset for interpersonal relation classification in dyadic dialogues. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13125–13133.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Po-Nien Kung, Tse-Hsuan Yang, Yi-Cheng Chen, Sheng-Siang Yin, and Yun-Nung Chen. 2020. Zero-shot rationalization by multi-task transfer learning from question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2187–2197.

Bongseok Lee and Yong Suk Choi. 2021. Graph based network with contextualized representations of turns in dialogue. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 443–455.

Kenton Lee, Shimi Salant, Tom Kwiatkowski, Ankur Parikh, Dipanjan Das, and Jonathan Berant. 2016. Learning recurrent span representations for extractive question answering. *arXiv preprint arXiv:1611.01436*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulić, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Kam-Fai Wong, and Shang-Yu Su. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Shang-Yu Su, Xiujun Li, Jianfeng Gao, Jingjing Liu, and Yun-Nung Chen. 2018a. Discriminative deep dyna-q: Robust planning for dialogue policy learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3813–3823.

Shang-Yu Su, Kai-Ling Lo, Yi-Ting Yeh, and Yun-Nung Chen. 2018b. Natural language generation by hierarchical decoding with linguistic patterns. In *Proceedings of The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Fuzhao Xue, Aixin Sun, Hao Zhang, and Eng Siong Chng. 2021. GDPNet: Refining latent multi-view graph for relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14194–14202.

Fuzhao Xue, Aixin Sun, Hao Zhang, Jinjie Ni, and Eng-Siong Chng. 2022. An embarrassingly simple model for dialogue relation extraction. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6707–6711. IEEE.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. 2020. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4927–4940.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *arXiv preprint arXiv:2102.01373*.

A Reproducibility

A.1 Hyperparameters

All the hyper-parameters were selected by grid search in (0,1] with step 0.1. The loss functions are linearly combined and each of them has an adjustable weight.

TREND$_{\text{BERT-Base}}$

- Loss: $0.3 \cdot \mathcal{L}_{\text{trigger}} + 1.0 \cdot \mathcal{L}_{\text{relation}} + 1.0 \cdot \mathcal{L}_{\text{binary}}$
- schedule sampling: 0.7 for trigger prediction, 0.7 for binary classification

TREND$_{\text{BERT-Large}}$

- Loss: $0.3 \cdot \mathcal{L}_{\text{trigger}} + 1.0 \cdot \mathcal{L}_{\text{relation}} + 1.0 \cdot \mathcal{L}_{\text{binary}}$
- schedule sampling: 0.5 for trigger prediction, 0.7 for binary classification

Data	Training	Inference
DialogRE	15 mins×30	5 mins
DDRel (session-level)	15 mins×30	5 mins
DDRel (pair-level)	1.5 mins×30	10 secs

Table 7: Time efficiency on three sets of experiments.

A.2 Time Efficiency

The training and inference cost in terms of time is reported in Table 7.

User Satisfaction Modeling with Domain Adaptation in Task-oriented Dialogue Systems

Yan Pan
BMW Group, Germany
Technical University of Munich, Germany
Frank_panyan@outlook.com

Mingyang Ma
BMW Group, Germany
mingyang.ma@bmw.de

Bernhard Pflugfelder
BMW Group, Germany
bernhard.pflugfelder@bmw.de

Georg Groh
Technical University of Munich, Germany
grohg@in.tum.de

Abstract

User Satisfaction Estimation (USE) is crucial in helping measure the quality of a task-oriented dialogue system. However, the complex nature of implicit responses poses challenges in detecting user satisfaction, and most datasets are limited in size or not available to the public due to user privacy policies. Unlike task-oriented dialogue, large-scale annotated chitchat with emotion labels is publicly available. Therefore, we present a novel user satisfaction model with domain adaptation (USMDA) to utilize this chitchat. We adopt a dialogue Transformer encoder to capture contextual features from the dialogue. And we reduce domain discrepancy to learn dialogue-related invariant features. Moreover, USMDA jointly learns satisfaction signals in the chitchat context with user satisfaction estimation, and user actions in task-oriented dialogue with dialogue action recognition. Experimental results on two benchmarks show that our proposed framework for the USE task outperforms existing unsupervised domain adaptation methods. To the best of our knowledge, this is the first work to study user satisfaction estimation with unsupervised domain adaptation from chitchat to task-oriented dialogue.

1 Introduction

The developed task-oriented dialogue system has achieved great success for various business situations, such as virtual assistants and information-seeking systems with domain knowledge (Deriu et al., 2021). However, a dialogue chatbot with limited model capability sometimes fails to understand queries correctly and even annoys users with the wrong response. User Satisfaction Estimation (USE) is able to detect user satisfaction and enable adjustment of the strategy of the system. Liu et al. (2021) implemented a smooth handoff from the machine to a human agent when USE estimates a negative emotion from a user. When USE detects good user feedback in the deployment environment, chatbots can utilize this information to learn and improve continuously (Hancock et al., 2019).

In recent years, the USE in dialogue systems is always considered in the classification task. Previous works (Sun et al., 2021; Deng et al., 2022) show that data-driven pre-trained models can learn good exchange-level representations from task-oriented corpora and predict correct user satisfaction. Unfortunately, most user satisfaction datasets are very limited in size (Saha et al., 2020; Shi and Yu, 2018) or not publicly available due to user privacy policies (Wang et al., 2020). Moreover, it is time-consuming and expensive to conduct human evaluation experiments or crowd-sourcing for user satisfaction in a real-world task-oriented application.

Compared to the task-oriented dataset, the chitchat corpora from social media is easy-to-get but without explicit chatting targets. The underlying difference in linguistic patterns between the chitchat and task-oriented dialogue makes it difficult to utilize the chitchat dataset in the USE task directly. Therefore, unsupervised domain adaptation from chitchat to task-oriented dialogue is valuable and challenging in user satisfaction tasks.

As shown in Figure 1, we collect two dialogue sessions from human-human chitchat and human-machine task-oriented dialogue. In the chitchat, people talk around one topic and explicitly express their intents with emotions. In task-oriented dialogue, the user and system have explicit actions where the user wants to achieve his goal, and the system uses the background knowledge following the presetting actions. But users tend to implicitly show their emotions and are comfortable with the fulfillment of their goals.

To tackle the domain difference, we propose a novel USMDA framework and implement USE

Proceedings of the SIGdial 2022 Conference, pages 630–636
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

Chitchat	Emotion	Satisfaction
Listen to me! When my time comes, I wanna be buried at sea.	Neutral	Neutral
You what?	Scared	Dissatisfied
I wanna be buried at sea, it looks like fun.	Joyful	Satisfied

Task-oriented dialogue	Action	Satisfaction
Hello! Can you help me find a hotel room?	Inform Intent	Neutral
May I suggest 1 Hotel Brooklyn Bridge? It is a well reviewed, 4 star hotel.	Offer	-
What other options do I have?	Request Alternatives	Dissatisfied

Figure 1: Two example dialogue sessions in chitchat (Zahiri and Choi, 2018) and task-oriented dialogue (Rastogi et al., 2020).

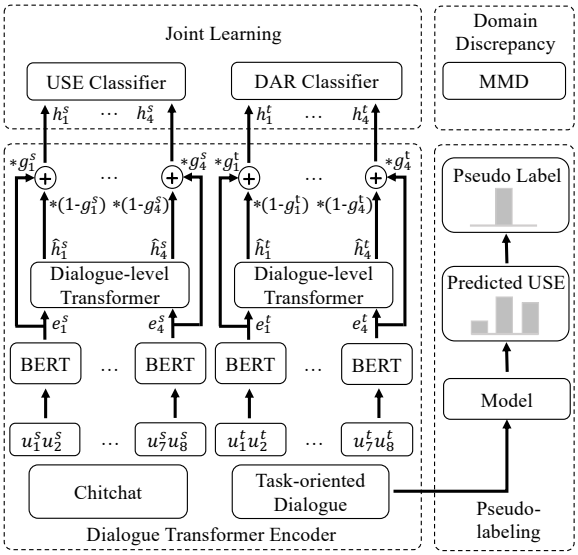


Figure 2: Overall framework architecture. The superscripts s and t denote the source chitchat data and target task-oriented dialogue data.

with unsupervised domain adaptation from chitchat to task-oriented dialogue. On the one hand, the model reduces the domain discrepancy of turn representations between chitchat and task-oriented dialogue datasets. On the other hand, the model learns satisfaction signals in context features from chitchat, and learns user actions in the task-oriented system with an additional Dialogue Action Recognition (DAR) task. Moreover, the framework utilizes the pseudo-labeling approach (Lee, 2013) to label the most confident predictions and build a stronger USE model.

To the best of our knowledge, our paper is the first attempt to explore the USE with domain adaptation from chitchat to task-oriented dialogue. In this work, we make the following contributions:

- We propose the USMDA framework to perform user satisfaction estimation with unsupervised domain adaptation from chitchat to task-oriented dialogue.
- The result shows that user actions and invariant dialogue-related features improve the performance of the USE model within an unsupervised domain adaptation setting.
- The results on two datasets demonstrate that the proposed framework in the USE task achieves better results than other domain adaptation approaches.

2 Problem Definition

We formulate the task of user satisfaction estimation with domain adaptation from chitchat to task-oriented dialogue. Given a set of chitchat and task-oriented dialogue sessions, each session contains N utterances $\{u_1, u_2, ..., u_N\}$. We split the N utterances into $\frac{N}{2}$ exchange turns $x_i = (u_{2i-1}, u_{2i})$. Each exchange turn is a communication either between multiple users or between user and system. Each exchange turn in chitchat is annotated with a satisfaction label y_i^s and each exchange turn in a task-oriented dialogue has a user action a_i^t. Our goal is to train a USE model using labeled chitchat data S and unlabeled task-oriented dialogue data T to predict the correct satisfaction label y_i^t on T.

3 Framework

This section introduces how to train a user satisfaction model with unsupervised domain adaptation. Figure 2 shows the overall architecture of our proposed framework USMDA with four different parts, including (1) dialogue Transformer encoder to capture a representation of each exchange-turn in the dialogue, (2) joint learning for USE with DAR, (3) reducing domain discrepancy between different distributed datasets, (4) predicting pseudo labels in the task-oriented dialogue, and retraining the model with the top-k pseudo labels.

3.1 Dialogue Transformer encoder

Chitchat and task-oriented dialogue samples are mixed in one batch X, which is fed into the shared backbone BERT (Devlin et al., 2019) to extract the exchange-level representation e_i of each exchange

turn x_i. Each e_i represents the information from an exchange turn:

$$e_i = BERT([CLS]u_{2i-1}[SEP]u_{2i}[SEP]) \quad (1)$$

The shared dialogue-level transformer encoder is built upon the exchange-level representations $\{e_1, e_2, ..., e_M\}$ of M exchange turns within a dialogue window. We adopt a Transformer encoder with a gated attention mechanism to capture the context information in the conversation:

$$\hat{h}_i = \text{Dialogue-Transformer}(e_i) \quad (2)$$

$$g_i = \text{Sigmoid}(W[e_i; \hat{h}_i]) \quad (3)$$

$$h_i = g_i * e_i + (1 - g_i) * \hat{h}_i \quad (4)$$

where $\hat{h}_i$ is the dialogue-level representation, g_i is the learned gated attention weight to combine two different level representations, W is a trainable matrix and h_i is the final representation of x_i.

3.2 Joint learning

The model jointly trains with USE and DAR to learn the specific user actions in the task-oriented dialogue. The USE classifier calculates the loss between the labeled satisfaction classes and predictions in the chitchat dataset. The DAR classifier learns to predict correct user actions in the task-oriented dataset. The joint learning loss is the sum of losses from USE and DAR classifiers:

$$\mathcal{L}_{Joint} = \mathcal{L}_{USE} + \alpha\mathcal{L}_{DAR} \quad (5)$$

where α denotes the hyperparameter to balance USE and DAR tasks.

3.3 Domain discrepancy

The framework uses maximum mean discrepancy (MMD) (Gretton et al., 2012; Long et al., 2015) to measure the distance between chitchat and task-oriented dialogue dataset distributions. MMD computes the distance between two exchange-level representations with Gaussian kernel, i.e., $k(e_i^s, e_j^t) = \exp(-\|e_i^s - e_j^t\|^2)$. Finally, we combine the joint learning loss and MMD as the overall loss:

$$\mathcal{L} = \mathcal{L}_{Joint} + \beta(\frac{4}{|X|^2}\sum_{i=1}^{\frac{|X|}{2}}\sum_{j=1}^{\frac{|X|}{2}} k(e_i^s, e_j^t)) \quad (6)$$

where e_i^s and e_j^t are two exchange-level representations from chitchat and task-oriented dialogue, β denotes the hyperparameter balancing the joint-learning loss and MMD, and $|X|$ is the size of a mixed batch X.

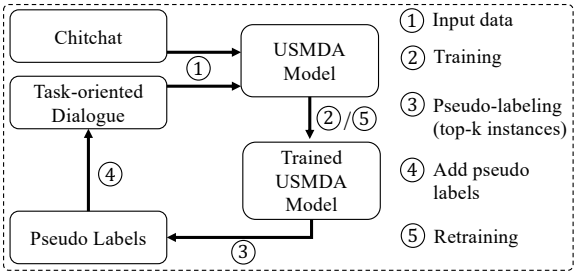


Figure 3: Retraining process with pseudo labels.

3.4 Pseudo-labeling

After joint learning and reducing domain discrepancy, the user satisfaction model makes the satisfaction prediction $\hat{y}_i^t$ on each exchange turn x_i^t from task-oriented dialogue. We measure the confidence of predictions by predicted scores. As shown in Figure 3, the top-k instances with the highest predicted scores are set as pseudo labels for retraining.

4 Experiments

4.1 Datasets and evaluation metrics

We conduct the proposed framework on the chitchat dataset EmoryNLP (Zahiri and Choi, 2018) and two task-oriented dialogue datasets: MultiWOZ 2.1 (MWOZ) (Eric et al., 2020) and Schema Guided Dialogue (SGD) (Rastogi et al., 2020). Moreover, we use the sampled 1000 dialogues from each of the MWOZ and SGD datasets, which are annotated with a five level satisfaction scale by Sun et al. (2021). The seven emotions in chitchat and five rating scores in task-oriented dialogue datasets are mapped into the coarse-grained labels "dissatisfied/neutral/satisfied" following existing work (Deng et al., 2022; Zahiri and Choi, 2018). For the DAR task, the MWOZ dataset is labeled with 21 actions by Eric et al. (2020), and the SGD dataset has 12 actions from Rastogi et al. (2020). We use the EmoryNLP dataset as a labeled source dataset and randomly choose 300 dialogues from each of the task-oriented dialogue datasets as unlabeled target datasets. The remaining 700 labeled dialogues from each task-oriented dialogue dataset are used for testing.

Following most existing work on emotion recognition in conversation, we report Macro-F1 and Micro-F1 scores for evaluating USE performance. Macro-F1 takes the average of all the per-class F1, and Micro-F1 computes the F1 of the aggregated contributions of classes.

	MWOZ		SGD	
Model	Macro	Micro	Macro	Micro
Performance without domain adaptation				
Bert (baseline)	37.98	45.51	40.66	49.15
ToD Bert	31.69	40.49	35.80	43.35
Performance with domain adaptation				
WDGRL	38.58	46.26	41.77	49.91
DANN	37.68	47.91	46.55	51.28
USMDA	**43.27**	**48.50**	**56.01**	**57.91**
Performance with supervised learning				
Upper bound	45.32	48.94	59.66	61.09

Table 1: Primary results with Micro-F1 and Macro-F1 metrics on task-oriented dialogue datasets.

4.2 Other models

We use the BERT model as our baseline model and the backbone for our proposed method for a thorough comparison. The following related models with task-oriented dialogue pretraining or different unsupervised domain adaptation methods are implemented:

- ToD Bert (Wu et al., 2020) is pretrained with masked-language modeling strategy and response selection task on nine task-oriented dialogue datasets.
- WDGRL (Shen et al., 2018) learns domain invariant representations by reducing empirical Wasserstein distance with an adversarial strategy.
- DANN (Ganin et al., 2016) uses domain adversarial training to learn the features that can not discriminate in domain adaptation. The DANN method is most widely used for unsupervised domain adaptation task in natural language processing.

5 Results and Analysis

5.1 Overall performance

Table 1 shows primary experiment results, including the following models: (1) the baseline model and ToD Bert using only the source chitchat dataset, (2) several models with domain adaptation strategies and access to the user actions from the target data, (3) the BERT-based model with supervised learning on task-oriented datasets as upper bound.

We made the following notable observations:

(1) Our unsupervised domain adaptation strategy is effective in improving the performance for USE on two task-oriented dialogue datasets. USMDA leads to a significant improvement in Macro-F1 of 5.29% on MWOZ and 15.35% on SGD, and a performance improvement in Micro-F1 of 2.99% on MWOZ and 8.76% on SGD. Our proposed framework USMDA successfully solves the domain shift problem for USE from chitchat to task-oriented dialogue. USMDA, without any satisfaction labels in task-oriented data, achieves a competitive Micro-F1 48.50% on MWOZ, which is comparable to the upper bound model with supervised learning.

(2) Our framework USMDA achieves the best performance with domain adaptation for two datasets. On average, the models with domain adaptation have better performance than the baseline model. This suggests that the domain-invariant dialogue-related features boost the performance of the user satisfaction model. Compared to other domain adaptation approaches, USMDA leads to a comparatively significant improvement. We demonstrate that our proposed framework USMDA to learning domain-invariant dialogue-related features is more effective than WDGRL and DANN.

(3) Baseline model, using only source chitchat samples, does not perform competitively. Even though ToD-BERT is pretrained with nine task-oriented dialogue datasets, it has a subpar performance without domain adaptation in the USE task. The unsatisfactory results without domain adaptation suggest that specific domain features are valuable and necessary for USE in task-oriented dialogue.

5.2 Ablation study

To understand the impacts of different individual parts in our domain adaptation strategy, we conduct an ablation study on three simplified modules of our proposed framework (see Table 2). We can observe that by removing any module, this results in worse performance. Removing joint learning leads to the most significant loss in Micro-F1 by 6.96% on SGD. This indicates that user actions throughout the dialogue reflect user satisfaction and are important dialogue-related specific features in task-oriented dialogue.

Table 2 shows that the improvement transfers well across both datasets. Learning transferable features using MMD is beneficial because dropping MMD impairs the performance by 1.17% Macro-F1 and 0.85% Micro-F1 on SGD. Moreover, removing the pseudo-labeling degrades the performance

	MWOZ		SGD	
	Macro	Micro	Macro	Micro
w/o pseudo	-5.33	-0.58	-3.88	-1.20
w/o MMD	-0.37	-0.26	-1.17	-0.85
w/o joint	-0.22	-0.50	-6.27	-6.96

Table 2: Ablation study of USMDA on pseudo-labeling, joint learning and MMD. A negative value means a performance loss by removing module.

by 3.9-5.3% Macro-F1 and 0.6-1.2% Micro-F1, indicating the benefits of the data-centric approach to the USE task.

5.3 Discussion and future work

Compared to the kernelized method MMD, the WDGRL and DANN are adversarial training strategies. Table 1 shows that WDGRL improves the model performance only slightly and that DANN does not always lead to the increased target domain performance. While traditional adversarial training strategies are sometimes unable to gain improvements with pre-trained language models, simple MMD is efficient at learning domain-invariant features. Our proposed framework achieves impressive results on the two fixed datasets. In the future, we will evaluate this framework on real-life scenarios.

6 Conclusion

We adopt joint-learning, MMD, and pseudo-labeling with domain adaptation to improve the strong USE model in task-oriented dialogue. The results show that domain adaptation with user actions is effective in the USE task. MMD has positive effects on overall performance by learning domain-invariant dialogue-related feature representations. The pseudo-labeling is important for USE with unsupervised domain adaptation. Our proposed USMDA framework has comparable results like the supervised model, encouraging future work addressing domain adaptation in the USE task.

Acknowledgements

The BMW Group supported the content of this work. We thank Davide Cadamuro and the reviewers for the invaluable feedback.

References

Yang Deng, Wenxuan Zhang, Wai Lam, Hong Cheng, and Helen Meng. 2022. User satisfaction estimation with sequential dialogue act modeling in goal-oriented conversational systems. In *WWW '22: The Web Conference 2022*.

Jan Deriu, Alvaro Rodrigo, Arantxa Otegi, Guillermo Echegoyen, Sophie Rosset, Eneko Agirre, and Mark Cieliebak. 2021. Survey on evaluation methods for dialogue systems. *Artificial Intelligence Review*, 54(1):755–810.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. MultiWOZ 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 422–428, Marseille, France. European Language Resources Association.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from dialogue after deployment: Feed yourself, chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy. Association for Computational Linguistics.

Dong-Hyun Lee. 2013. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*.

Jiawei Liu, Zhe Gao, Yangyang Kang, Zhuoren Jiang, Guoxiu He, Changlong Sun, Xiaozhong Liu, and Wei Lu. 2021. Time to transfer: Predicting and evaluating machine-human chatting handoff. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):5841–5849.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Tulika Saha, Sriparna Saha, and Pushpak Bhattacharyya. 2020. Towards sentiment aided dialogue policy learning for multi-intent conversations using hierarchical reinforcement learning. *PloS one*, 15(7):e0235367.

Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Weiyan Shi and Zhou Yu. 2018. Sentiment adaptive end-to-end dialog systems. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1509–1519, Melbourne, Australia. Association for Computational Linguistics.

Weiwei Sun, Shuo Zhang, Krisztian Balog, Zhaochun Ren, Pengjie Ren, Zhumin Chen, and Maarten de Rijke. 2021. Simulating user satisfaction for the evaluation of task-oriented dialogue systems. In *Proceedings of the 44rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21. ACM.

Jiancheng Wang, Jingjing Wang, Changlong Sun, Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, and Guodong Zhou. 2020. Sentiment classification in customer service dialogue with topic-aware multitask learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9177–9184.

Chien-Sheng Wu, Steven C.H. Hoi, Richard Socher, and Caiming Xiong. 2020. TOD-BERT: Pre-trained natural language understanding for task-oriented dialogue. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 917–929, Online. Association for Computational Linguistics.

Sayyed M Zahiri and Jinho D Choi. 2018. Emotion detection on tv show transcripts with sequence-based convolutional neural networks. In *Workshops at the thirty-second aaai conference on artificial intelligence*.

A Appendix

A.1 Datasets

We perform experiments on dialogue corpora, using 713 dialogues from EmoryNLP, 1000 dialogues from MWOZ, and 1000 dialogues from SGD. A dialogue session is divided into dialogue windows. The number of considered exchange-level turns in a dialogue window is four.

EmoryNLP	Emotion	Satisfaction
Monica: Hey.	Neutral	Neutral
Rachel: Hey.	Neutral	Neutral
Monica: How's the big anniversary dinner?	Neutral	Neutral
Rachel: Well, we never actually got to dinner.	Sad	Unsatisfied
Monica: Ohhh, nice.	Sad	Unsatisfied
Rachel: No, we kinda broke up instead.	Sad	Unsatisfied
Monica: What?!	Scared	Unsatisfied
Rachel: God, Monica it's on the ceiling.	Scared	Unsatisfied

Table 3: Chitchat example from EmoryNLP.

EmoryNLP: EmoryNLP is an annotated chitchat dataset with fine-grained and coarse-grained emotions from the TV show, Friends. The EmoryNLP contains seven emotions: (1) Positive: powerful, joyful, peaceful, (2) Negative: mad, sad, scared, and (3) Neutral: neutral.

MWOZ: Multi-Domain Wizard-of-Oz 2.1 is a multi-domain task-oriented dialogue dataset. It contains dialogue utterances, user goals, and system actions over seven domains.

SGD: Schema-Guided Dialogue has multi-domain task-oriented dialogues over 16 domains and provides a challenging testbed for dialogue state tracking. SGD contains multiple schemas with a dynamic set of slots for task-oriented dialogue.

A.2 Training and hyperparameters

For the backbone of USMDA, we use the implementation of the BERT-base uncased model and of the ToD-Bert model from huggingface[1]. Both models are 768-dimensional Transformer self-attention encoders with 12 layers. The dialogue-level Transformer is a standard Transformer encoder with 2 layers.

We run each experiment four times with different seeds and calculate the average performance. The adamW optimizer is implemented with a learning rate of 2E-5, and the training epoch size is 2.

A.3 Dialogue Examples

The annotated chitchat and task-oriented dialogue examples are presented in Tables 3, 4.

[1]https://huggingface.co/

SGD	Action	Satisfaction
System: Okay, there is an American Airlines outbound flight that will leave at 11:40 am and the return flight will leave at 9:30 am. The cost of the ticket will be only $163.	Offer	-
USER: When is the arrival time of my return flight and to which airport it arrives?	Request	Neutral
System: The destination airport is Dulles International Airport and the flight will arrive at 1:42 pm.	Inform	-
USER: Sounds good to me.	Select	Satisfied
System: Do you want me to book tickets for this flight?	Offer	-
USER: Yup, please book 1 ticket in economy class which should be a refundable one, since my plan might change later on.	Affirm	Neutral
System: Okay! Please can you confirm me that you wish to fly from Atlanta to Washington in American Airlines on March 10th at 11:40 am and the return journey will be on March 14th at 9:30 am and you wish to book only 1 Economy ticket, right?	Confirm	-
USER: Yup, you're right. Is it a zero stops flight?	Affirm	Satisfied

Table 4: Task-oriented dialogue example from SGD.

N-best Response-based Analysis of Contradiction-awareness in Neural Response Generation Models

Shiki Sato[1] **Reina Akama**[1,3] **Hiroki Ouchi**[2,3] **Ryoko Tokuhisa**[1]
Jun Suzuki[1,3] **Kentaro Inui**[1,3]

[1]Tohoku University [2]Nara Institute of Science and Technology [3]RIKEN
{shiki.sato.d1,akama,tokuhisa,jun.suzuki,inui}@tohoku.ac.jp
hiroki.ouchi@is.naist.jp

Abstract

Avoiding the generation of responses that contradict the preceding context is a significant challenge in dialogue response generation. One feasible method is post-processing, such as filtering out contradicting responses from a resulting n-best response list. In this scenario, the quality of the n-best list considerably affects the occurrence of contradictions because the final response is chosen from this n-best list. This study quantitatively analyzes the contextual contradiction-awareness of neural response generation models using the consistency of the n-best lists. Particularly, we used polar questions as stimulus inputs for concise and quantitative analyses. Our tests illustrate the contradiction-awareness of recent neural response generation models and methodologies, followed by a discussion of their properties and limitations.

1 Introduction

Recent advanced response generation models (Zhang et al., 2020; Adiwardana et al., 2020; Roller et al., 2021) can generate relevant and meaningful responses, which can resolve dull response problems (Vinyals and Le, 2015; Sordoni et al., 2015; Serban et al., 2016). This advancement reveals additional flaws in the quality of neural model responses, such as *contradiction*. Contradiction is a critical error in dialogue because a single contradictory response can disrupt the flow of the dialogue (Higashinaka et al., 2015).

A generation model outputs a response by selecting the candidate with the highest likelihood (1-best) from an n-best candidate list. Prior work has demonstrated that generating the n-best lists with noncontradictory 1-bests is an open challenge (Nie et al., 2020; Kim et al., 2020; Li et al., 2021). Thus, one practical technique for avoiding contradiction is to have an accurate contradiction detector that eliminates all contradictory candidates from the n-best list (Nie et al., 2020). In this scenario, the consistency of all candidates in the n-best list, not just the 1-best, substantially impacts whether the final output is contradictory because the final response is chosen from the n-best list. Nonetheless, earlier quantitative investigations of contradiction relied solely on 1-bests from models (Li et al., 2021).

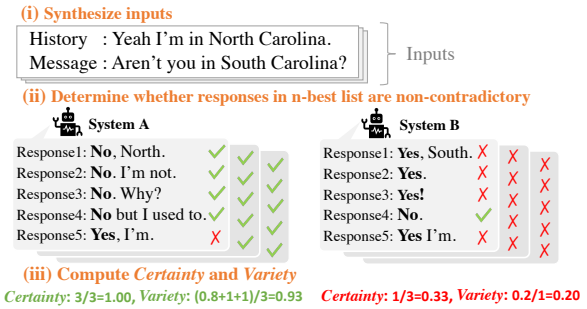


Figure 1: Overview of our analysis framework. The framework analyzes n-best lists by (i) synthesizing a stimulus input that induces contradictions, (ii) automatically determining whether responses in the n-best lists are contradictory, and (iii) computing *Certainty* and *Variety*.

In this study, we analyze the n-best lists generated by the models to explore methods for enhancing neural response generation to avoid contradiction. Specifically, we first consider how analyzing an n-best list should be approached. Then, we propose a method for statistically analyzing the n-best lists (Figure 1). Since it is impractical to study all conceivable contradictions in a dialogue, we first focus on contradictions in response to polar questions.[1] We use our method to highlight the contradiction-awareness of recent high-performance neural response generation models and methodologies. Our results show that beam search has limitations in terms of avoiding contradiction and that the newer techniques, such as unlikelihood training (Welleck et al., 2020), can help overcome these limitations.

[1]Codes and test set are available at https://github.com/shiki-sato/nbest-contradiction-analysis

Proceedings of the SIGdial 2022 Conference, pages 637–644
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

NLI data				Dialogue context for our test		
Entailment	Premise:	yeah i'm in North Carolina	→	**ENTQ**	History:	Yeah I'm in North Carolina.
	Hypothesis:	I'm in North Carolina.			Message:	Are you in North Carolina?
Contradiction	Premise:	yeah i'm in North Carolina	→	**CNTQ**	History:	Yeah I'm in North Carolina.
	Hypothesis:	I'm in South Carolina.			Message:	Aren't you in South Carolina?

Table 1: Acquiring dialogue context by transforming the Natural Language Inference (NLI) data.

2 Analysis perspectives

First, n-best lists must be generated to prevent contradiction, assuming the filters can remove contradictory responses. An ideal model produces output that is noncontradictory and outperforms in many other criteria, such as relevance or informativeness. A model must generate at least one noncontradictory candidate to deliver a noncontradictory output. Furthermore, even noncontradictory candidates could be eliminated based on other criteria (e.g., relevance, informativeness). Therefore, it can be hypothesized that having more noncontradictory responses in an n-best list would enhance the final output quality across various criteria. Taking the above into account, we examine n-best lists based on the certainty of the existence of noncontradictory responses (***Certainty***), and the variety of noncontradictory responses (***Variety***):

- ***Certainty***: The proportion of the n-best lists that have at least one noncontradictory response.
- ***Variety***: The proportion of noncontradictory responses in each n-best list when only the n-best lists with at least one noncontradictory response are collected.

Given a set of inputs $\mathcal{Q}$, we calculate them as follows:

$$\textbf{\textit{Certainty}} = \frac{|\mathcal{Q}'|}{|\mathcal{Q}|}, \textbf{\textit{Variety}} = \frac{1}{|\mathcal{Q}'|}\sum_{q\in\mathcal{Q}'}\frac{\mathrm{cnt}(f(q))}{|f(q)|}$$

$$\mathcal{Q}' = \{q \mid \mathrm{cnt}(f(q)) > 0, q \in \mathcal{Q}\}$$

where $f(\cdot)$ is an n-best list generation function and $\mathrm{cnt}(\cdot)$ is a function that returns the number of noncontradictory responses from a given n-best list. For example, the *Certainty* of a model that generates n-best lists with a combination of noncontradictory and contradictory responses is high, but its *Variety* is low. However, a model that always generates n-best lists with only noncontradictory or contradictory responses has a high *Variety* but a low *Certainty*. We anticipate that n-best lists must include noncontradictory responses (*Certainty*$= 1.0$), with a high proportion (high *Variety*).

3 Analytical inputs and evaluation

To analyze a model from the aforementioned viewpoints, we consider how to prepare the analytical inputs and evaluate the generated responses in this section.

3.1 Inputs for highlighting contradictions

Polar echo question. An *echo question* (Noh, 1998) confirms or clarifies the context information by repeating the utterance of another speaker. It is commonly used when the speaker did not hear or understand what was said correctly, or when the speaker wishes to express incredulity. Based on Li et al. (2021)'s discovery, contradictions emerge mostly when speakers refer to earlier information communicated in dialogue; we use echo questions as stimulus input in our analysis to elicit contradictory responses. We use polar-typed echo questions to make our analysis more succinct and quantitative. Since polar questions allow for basically only two responses, *yes* or *no*, we can clearly determine whether the generated response is contradictory or not. Furthermore, by analyzing the produced responses as a yes/no binary classification issue, it allows for quantitative discussion of experimental outcomes based on the probability level.

Input preparation. We use the dataset from the natural language inference (NLI) task to effectively obtain the analytical inputs described in the preceding paragraph. This dataset specifies the logical relationship (i.e., entailment, neutrality, or contradiction) between a premise and its associated hypothesis. We transform the NLI dataset into dialogue data using a set of basic rewriting rules.[2] Our test involves two types of inputs, which can be classified as follows:

- ENTQ: generating a *confirmation* response.

[2]The details are described in Appendix A.

- CNTQ: generating a *refutation* response.

Table 1 displays the input samples and how they are transformed from the initial NLI data. Each input is made up of the following two utterances: the history and message. In our analysis, the model generates responses to a given message, assuming the model has generated the history in the preceding turn.

3.2 Contradiction detection for output

To compute the *Certainty* and *Variety*, we must first determine whether each generated response in the n-bests compared to the inputs is contradictory. The simplest method for detecting the contradictions is to check whether the response begins with *yes* or *no*. However, in the event of an indirect expression (e.g., *Why not?*), this method cannot detect the contradictions. Therefore, we use an automated yes-no classifier to categorize the n-best responses to ENTQ/CNTQ. We train the classifier by fine-tuning RoBERTa (Liu et al., 2019) using the Circa dataset (Louis et al., 2020), which comprises pairs of polar questions and indirect responses, as well as annotations for the answer's interpretation, to categorize utterances as affirmations or refutations.[3]

4 Experiments

We demonstrate how our framework shows the properties of n-best lists, which could be quite influential in terms of avoiding contradiction. We demonstrate this by comparing the n-bests generated by conventional beam search (BS) versus recently proposed techniques.

4.1 Experimental settings

Inputs preparation. We used the Multi-Genre NLI Corpus (Williams et al., 2018) to obtain analytical inputs, which is a large scale and is consistent in good quality NLI data. We created 2,000 ENTQ/CNTQ inputs by extracting 2,000 samples labeled with *entailment* or *contradiction*.[4]

Response generation models. We used the following two recently developed high-performance models: DialoGPT (Zhang et al., 2020) and Blender (Roller et al., 2021).[5]

[3]The details are described in Appendix B.

[4]We used the samples in the TELEPHONE domain; this domain covers open-domain conversations.

[5]The details of the settings are described in Appendix C.

	Certainty		*Variety*	
Model	ENTQ	CNTQ	ENTQ	CNTQ
Blender 400M	0.806	0.747	0.780	0.775
Blender 1B	0.832	0.752	0.832	0.753
Blender 3B	0.856	0.768	0.824	0.737
DialoGPT 345M	0.938	0.917	0.750	0.669
DialoGPT 762M	0.883	0.918	0.671	0.713

Table 2: *Certainty* and *Variety* of 10-best lists using beam search with beam size $B = 10$.

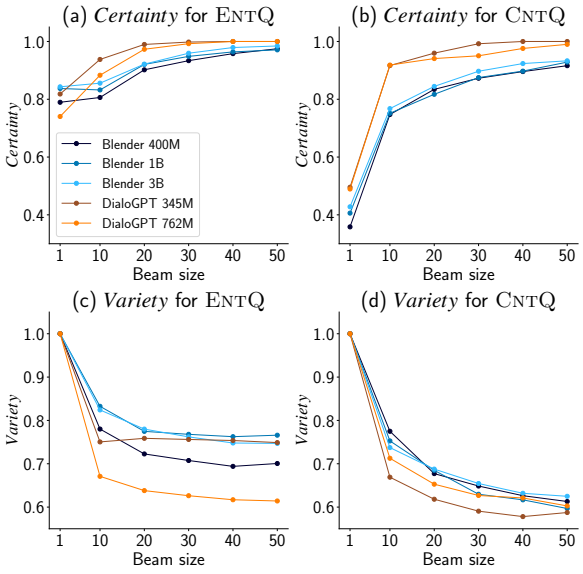


Figure 2: *Certainty* and *Variety* of n-best lists using beam search with various beam sizes.

4.2 Analysis of *n*-best using beam search

Let B denote the beam size during generation. It has been empirically found that using beam search with $B = 10$ to generate a response yields excellent quality results and has a frequently used value (Zhang et al., 2020; Roller et al., 2021). Table 2 displays the *Certainty* and *Variety* of 10-best lists generated using these methods. Figure 2 also depicts the *Certainty* and *Variety* of n-best lists generated using different beam sizes.

***Certainty*.** Table 2 illustrates that in approximately 10% of CNTQ-type inputs, even the highest scoring model generates 10-best lists full of contradictory responses. Even with a perfect response filter, the models are unable to provide noncontradictory answers to these questions. It should be emphasized that the error rate is not low, given that the inputs are polar questions with highly restricted viable responses. Expanding the beam size can increase the number of n-best lists with at least one noncontradictory response. Indeed, increas-

ing the beam size enhances the *Certainty* ((a) and (b) in Figure 2). By increasing B to 40, the *Certainty* of using DialoGPT 345M for both ENTQ- and CNTQ-type inputs achieve 1.0.

***Variety*.** With $B = 10$, all the models' *Variety* are more than 0.5 (chance rate) (Table 2). Therefore, rather than being fully random, the models generate n-best lists with a degree of directionality toward avoiding contradictions. However, increasing the size of beam reduces the *Variety* ((c) and (d) in Figure 2), resulting in lower output quality. For example, the *Variety* of DialoGPT 345M with $B = 40$ for CNTQ-type inputs (a model with *Certainty* of 1.0 for both ENTQ- and CNTQ-type inputs) decreases to 0.58.

Overall. In terms of avoiding contradiction, our analytical framework demonstrated the features of the n-best lists of the beam search. The *Certainty* did not achieve 1.0 in the commonly used configuration ($B = 10$). When the beam size is increased, the *Certainty* increases to 1.0, whereas the *Variety* reduces dramatically. These results show the trade-off between *Certainty* and *Variety* as a function of beam size; in this example, we found constraints in obtaining high *Certainty* and *Variety* with beam search. Furthermore, it is found that the *Certainty* obtained using DialoGPT is greater than that obtained using Blender, whereas the opposite is true for *Variety*, suggesting that various models behave differently in terms of *Certainty* and *Variety*. This study emphasizes the significance of examining the *Certainty* and *Variety* of each model.

4.3 Analysis of *n*-best by various techniques

How to achieve high *Certainty* and *Variety*? One method to increase *Certainty* is to generate n-best lists with a wider range of responses, such that each n-best list is guaranteed to contain a specific number of noncontradictory responses. The diverse beam search (DBS) (Vijayakumar et al., 2016) and nucleus sampling (NS) (Holtzman et al., 2020) methods are used to construct such n-best lists. Furthermore, Li et al. (2020) recently proposed models that use unlikelihood (UL) training to assign low probabilities to contradict responses. Using these models to generate n-best lists will almost certainly enhance both *Certainty* and *Variety*. We assess the n-best lists generated using these three strategies to see how much these techniques enhance *Certainty* and *Variety* (n-best lists generated using DBS and NS, and n-best lists generated using beam search together with the UL training). Appendix C contains a description of the techniques used for this analysis.

	Certainty		*Variety*	
Technique	ENTQ	CNTQ	ENTQ	CNTQ
BS	0.856	0.768	0.824	0.737
DBS	0.999	0.981	0.758	0.478
NS	1.000	0.994	0.755	0.462
UL ($\alpha = 0$)	1.000	0.996	0.406	0.759
UL ($\alpha = 1$)	0.943	0.900	0.920	0.938
UL ($\alpha = 10$)	0.910	0.937	0.969	0.968

Table 3: *Certainty* and *Variety* of 10-best lists using various techniques with Blender 3B.

Result. Table 3 displays the *Certainty* and *Variety* of the 10-best lists generated using BS, DBS, NS, and UL.[6] The values of α show the degree of UL loss during fine-tuning. Here UL with $\alpha = 0$ used the response generation model fine-tuned with maximum likelihood in the same training settings as those used for UL with $\alpha > 0$. Thus, note that comparing UL with $\alpha = 0$ and $\alpha > 0$ allows a fair comparison between likelihood and unlikelihood training. The results reveal the properties of the n-best lists obtained for the three techniques, as well as the extent to which the techniques increase *Certainty* and *Variety*. The *Certainty* obtained using the DBS and NS method reach 1.0 for significantly lower search sizes than that for the BS to attain a *Certainty* of 1.0; the *Variety* for CNTQ-type inputs are less than 0.5 (chance rate). Thus, using the DBS and NS methods efficiently improves *Certainty* compared with the results obtained using the beam search; nevertheless, the methods do not simultaneously attain high *Certainty* and *Variety*. However, the *Certainty* obtained using UL with $\alpha > 0$ are greater than those obtained using the BS, and this was accomplished while maintaining higher *Variety* than those obtained using the BS and UL with $\alpha = 0$ (likelihood training). Our findings show that generation models are advancing toward high *Certainty* and *Variety*, which is particularly true for the recently proposed UL loss method. Despite the highly restricted viable responses, i.e., *yes* or *no*, the *Certainty* obtained using UL with $\alpha > 0$ does not reach 1.0. Thus, we conclude that there is still room for improvement in n-best list generation

[6] For the BS, DBS, and UL, we obtained the 10-best lists setting beam size to 10. For the NS, we got the 10-best lists by performing nucleus sampling ten times.

in terms of avoiding contradiction.

5 Conclusion

Based on the recent development of contradiction detectors, removing contradictory candidates from models' n-best lists is a practical method for avoiding contradiction. In this method, the consistency of all candidates in the n-best lists substantially affects whether the final outputs are contradictory.

We quantitatively examined the properties of the n-best lists in terms of avoiding contradiction, using polar-typed questions as analytical inputs. We demonstrated that the proposed framework exhibits the properties of n-best lists based on *Certainty* and *Variety*. *Certainty* determines whether an n-best list has at least one noncontradictory response, whereas *Variety* evaluates how many noncontradictory responses each n-best list has. The results, particularly, demonstrated the present limitations on achieving high *Certainty* and *Variety* when using the well-established beam search method. In addition, our method emphasizes the improvements in *Certainty* and *Variety* achieved by recently proposed response generation strategies.

Our approach, which analyzes models' n-best lists based on *Certainty* and *Variety*, can be applied to any response generation problem, not just polar-typed response generation, which will be future work.

Acknowledgments

We would like to thank all anonymous reviewers for their insightful comments. We also thank Ana Brassard and Yosuke Kishinami for their valuable feedback and support. This work was partly supported by JSPS KAKENHI Grant Numbers JP21J22383, JP22K17943, JST Moonshot R&D Grant Number JPMJMS2011, and a Bilateral Joint Research Program between RIKEN AIP Center and Tohoku University.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. In *arXiv preprint arXiv:2001.09977*.

Ryuichiro Higashinaka, Kotaro Funakoshi, Masahiro Araki, Hiroshi Tsukahara, Yuka Kobayashi, and Masahiro Mizukami. 2015. Towards taxonomy of errors in chat-oriented dialogue systems. In *Proceedings of the 16th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, pages 87–95.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The Curious Case of Neural Text Degeneration. In *Proceedings of the eighth international conference on learning representations (ICLR)*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Hyunwoo Kim, Byeongchang Kim, and Gunhee Kim. 2020. Will I sound like me? Improving persona consistency in dialogues through pragmatic self-consciousness. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 904–916.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. Don't say that! Making inconsistent dialogue unlikely with unlikelihood training. In *Proceedings of the 58th annual meeting of the association for computational linguistics (ACL)*, pages 4715–4728.

Zekang Li, Jinchao Zhang, Zhengcong Fei, Yang Feng, and Jie Zhou. 2021. Addressing Inquiries about History: An Efficient and Practical Framework for Evaluating Open-domain Chatbot Consistency. In *Findings of the joint conference of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (ACL-IJCNLP)*, pages 1057–1067.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In *arXiv preprint arXiv:1907.11692*.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding Indirect Answers. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 7411–7425.

Alexander H. Miller, Will Feng, Adam Fisch, Jiasen Lu, Dhruv Batra, Antoine Bordes, Devi Parikh, and Jason Weston. 2017. ParlAI: A dialog research software platform. In *Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP): System demonstrations*, pages 79–84.

Yixin Nie, Mary Williamson, Mohit Bansal, Douwe Kiela, and Jason Weston. 2020. I like fish, especially dolphins: Addressing Contradictions in Dialogue Modeling. In *Proceedings of the 59th annual meeting of the association for computational linguistics (ACL)*, pages 1699–1713.

Eun-Ju Noh. 1998. Echo Questions: Metarepresentation and Pragmatic Enrichment. *Linguistics and Philosophy*, 21(6):603–628.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M. Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: Main volume (EACL)*, pages 300–325.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of the 30th AAAI conference on artificial intelligence (AAAI-16)*, pages 3776–3783.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies (NAACL-HLT)*, pages 196–205.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse Beam Search for Improved Description of Complex Scenes. In *Proceedings of the 32nd AAAI conference on artificial intelligence (AAAI-18)*.

Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. In *Proceedings of the 31st international conference on machine learning (ICML) deep learning workshop*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural Text Generation With Unlikelihood Training. In *Proceedings of the eighth international conference on learning representations (ICLR)*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies (NAACL-HLT)*, volume 1, pages 1112–1122.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP): System demonstrations*, pages 38–45.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th annual meeting of the association for computational linguistics (ACL): System demonstrations*, pages 270–278.

A Details of transforming NLI data

As described in Section 3.1, we obtain an analytical input from the NLI dataset. Specifically, we convert the hypothesis sentence of an NLI sample into a yes-no question. We describe the procedure as follows:

1. Detect the first verb of a sentence.
2. Move the verb to the beginning of the sentence, or put one of {*Do*, *Does*, *Did*} at the front of the sentence, changing the verb back to its base (e.g., *made* → *make*).
3. Change first-person pronouns to second-person pronouns and second-person pronouns to first-person pronouns (e.g., *my* → *your*).
4. Change the punctuation mark at the end of the sentence to a question mark.

We used spaCy (`en_core_web_sm`) (Honnibal and Montani, 2017) to detect the verbs of hypothesis sentences. We did not use NLI samples with syntactically complex hypothesis sentences, such as those containing coordinating conjunctions, to avoid obtaining ungrammatical inputs. Further details are provided in our source codes.[7]

B Details of yes-no classifier

Training settings. On the Circa dataset, we fine-tuned the pretrained RoBERTa (`roberta-large`) implemented by Hugging Face (Wolf et al., 2020). We divided the dataset at random into train:valid = 8 : 2. The other training parameters were identical to those used by Louis et al. (2020).

Performance of classifier. To investigate the performance of the classifier, we measured the classification accuracy. First, we manually labeled the top-1 responses in the 10-best lists generated by the analysis presented in Section 4.2 with one of the two following labels: *Contradictory* or *Noncontradictory*. The accuracy with which the automated evaluation categorized the labeled responses was then evaluated. We selected 500 responses[8] from 50 ENTQ/CNTQ inputs drawn at random from our test for the evaluation. The classifier classified 433/500 responses (see Appendix C), and the accuracy was 0.921. Some examples of the classification are shown in Table 4. The classifier correctly detected the contradiction in the model response using an indirect expression, in Example 1. However, in Example 2, the classifier failed to detect the contradiction of the model response, having both a noncontradictory direct expression ("No") and a contradictory indirect expression (the part of the response after "No"). We found that the classifier tended to misclassify model responses containing the contradictions with themselves, such as Example 2.

[7] https://github.com/shiki-sato/nbest-contradiction-analysis

[8] 100 responses generated by each of 5 generation models.

History:	and we didn't ever call it uh Cokes and such you know we call it soda.
Message:	Don't you always call it Coke?
Model Response:	We call it coke.
Human Label:	**Contradictory**
Predicted Label:	**Contradictory**

(a) Example 1

History:	The buying a house was the last thing that i wanted to do.
Message:	Weren't you desperate to buy a house?
Model Response:	No, I just wanted to buy a house.
Human Label:	**Contradictory**
Predicted Label:	**Non-contradictory**

(a) Example 2

Table 4: Examples of the response classification results by the yes-no classifier. The model responses were generated by Blender 400M using beam search with beam size $B = 10$.

Model	ENTQ	CNTQ
Blender 400M	1331 / 2000	1270 / 2000
Blender 1B	1413 / 2000	1316 / 2000
Blender 3B	1566 / 2000	1403 / 2000
DialoGPT 345M	1126 / 2000	924 / 2000
DialoGPT 762M	1044 / 2000	956 / 2000

Table 5: Number of stimulus inputs analyzed to calculate the *Certainty* and *Variety* described in Table 2.

Model	ENTQ	CNTQ
BS	1566 / 2000	1403 / 2000
DBS	991 / 2000	882 / 2000
NS	818 / 2000	684 / 2000
UL ($\alpha = 0$)	1914 / 2000	1871 / 2000
UL ($\alpha = 1$)	1806 / 2000	1887 / 2000
UL ($\alpha = 10$)	1654 / 2000	1811 / 2000

Table 6: Number of stimulus inputs analyzed to calculate the *Certainty* and *Variety* described in Table 3.

C Details of experiments

Number of analyzed stimulus inputs. To simplify the analysis, we omitted from Section 4 and

Appendix B the analytical inputs with one or more ambiguous responses in the n-best lists. We defined ambiguous responses as those that were not identified by the classifier as either affirmations or refutations.[9] Table 5 and Table 6 display the number of analytical inputs from the total of $2,000$ ENTQ/CNTQ used for the two analyses in Section 4.

Generation model settings. In Section 4 experiments, we used DialoGPT (Zhang et al., 2020) and Blender (Roller et al., 2021) as response generation models. We used the codes of ParlAI (Miller et al., 2017) with its default settings, except for `beam_length_penalty=0` to generate responses.

Unlikelihood training settings. We used unlikelihood training with Blender 3B for the study of Section 4.3. To use the unlikelihood training proposed by Li et al. (2020), we require training data that includes the following three elements: input (here, history, and message), gold response, and negative response. These training samples were created by altering the NLI data with entailing and contradicting hypotheses.[10] Table 7 displays the original NLI data and the transformed training samples. One NLI data set yields four types of questions (PositiveQ1, PositiveQ2, NegativeQ1, and NegativeQ2). We synthesized 8,000 samples from 2,000 NLI data and randomly divided them into train : valid = 9 : 1. We tuned the learning rate $\{7.0 \times 10^{-4}, 7.0 \times 10^{-5}, 7.0 \times 10^{-6}, 7.0 \times 10^{-7}, 7.0 \times 10^{-8}\}$ and the number of warmup updates $\{50, 100\}$ for each $\alpha = \{0, 1, 10\}$ for training. The rest of the training parameters are identical to those used by Roller et al. (2021). It is worth noting that we only trained the models marked as UL in Section 4.3 on these transformed data.

Premise:	yeah i'm in North Carolina
Hypothesis – **Entailment**:	I'm in North Carolina.
Hypothesis – **Contradict**:	I'm in South Carolina.

(a) Original NLI data

PositiveQ1

History:	Yeah I'm in North Carolina.
Message:	Are you in North Carolina?
Gold:	Yes, I'm in North Carolina.
Negative:	No, I'm in South Carolina.

PositiveQ2

History:	Yeah I'm in North Carolina.
Message:	Are you in South Carolina?
Gold:	No, I'm in North Carolina.
Negative:	Yes, I'm in South Carolina.

NegativeQ1

History:	Yeah I'm in North Carolina.
Message:	Aren't you in North Carolina?
Gold:	Yes, I'm in North Carolina.
Negative:	No, I'm in South Carolina.

NegativeQ2

History:	Yeah I'm in North Carolina.
Message:	Aren't you in South Carolina?
Gold:	No, I'm in North Carolina.
Negative:	Yes, I'm in South Carolina.

(b) Training samples for UL

Table 7: Example of transforming (a) original NLI data to (b) training sample for UL. We synthesized four questions, i.e., PositiveQ1, PositiveQ2, NegativeQ1, and NegativeQ2, from each NLI sample.

[9] Circa dataset has seven different labels such as "Yes" and "Probably/sometimes yes." We regard the responses classified into "In the middle" or "I am not sure" as ambiguous ones.

[10] Note that we did not use the identical NLI samples to synthesize ENTQ/CNTQ.

A Visually-Aware Conversational Robot Receptionist

Nancie Gunson, Daniel Hernandez Garcia, Weronika Sieińska
Angus Addlesee, Christian Dondrup, Oliver Lemon
Interaction Lab, School of Mathematical and Computer Sciences
Heriot-Watt University, Edinburgh, Scotland, UK
{n.gunson, d.hernandez_garcia, w.sieinska,
a.addlesee, c.dondrup, o.lemon}@hw.ac.uk

Jose L. Part
Alana AI
Edinburgh, UK
jose@alanaai.com

Yanchao Yu
School of Computing
Edinburgh Napier University, Scotland, UK
Y.Yu@napier.ac.uk

Abstract

Socially Assistive Robots (SARs) have the potential to play an increasingly important role in a variety of contexts including healthcare, but most existing systems have very limited interactive capabilities. We will demonstrate a robot receptionist that not only supports task-based and social dialogue via natural spoken conversation but is also capable of visually grounded dialogue; able to perceive and discuss the shared physical environment (e.g. helping users to locate personal belongings or objects of interest). Task-based dialogues include check-in, navigation and FAQs about facilities, alongside social features such as chit-chat, access to the latest news and a quiz game to play while waiting. We also show how visual context (objects and their spatial relations) can be combined with linguistic representations of dialogue context, to support visual dialogue and question answering. We will demonstrate the system on a humanoid ARI robot, which is being deployed in a hospital reception area.

1 Introduction

Socially Assistive Robots (SARs) are increasingly being explored in contexts ranging from education (Papadopoulos et al., 2020) to healthcare (González-González et al., 2021). It has been noted, however, that despite the success of SARs and spoken dialogue systems in their respective research fields, integration of the two is still rare (Lima et al., 2021) and social robots in general still lack interaction capabilities (Cooper et al., 2020). In a similar fashion, even recent research on combining vision and language has tended to centre around the use of still images (Mostafazadeh et al., 2017; Zhou et al., 2020), with few systems able to support visual dialogue as part of a natural, situated conversation.

Figure 1: Interacting with SPRING-ARI

The SPRING project aims to develop such a system in the form of a robot receptionist for visitors to an eldercare outpatient hospital. In this context the robot must be able to communicate naturally with users on a variety of both functional and social topics, including but not limited to those concerning the shared physical environment. We demonstrate our progress towards this goal, with a multi-modal conversational AI system that is integrated on an ARI robot[1] (Fig. 1) and which combines social

[1] https://pal-robotics-com/robots/ari

Proceedings of the SIGdial 2022 Conference, pages 645–648
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

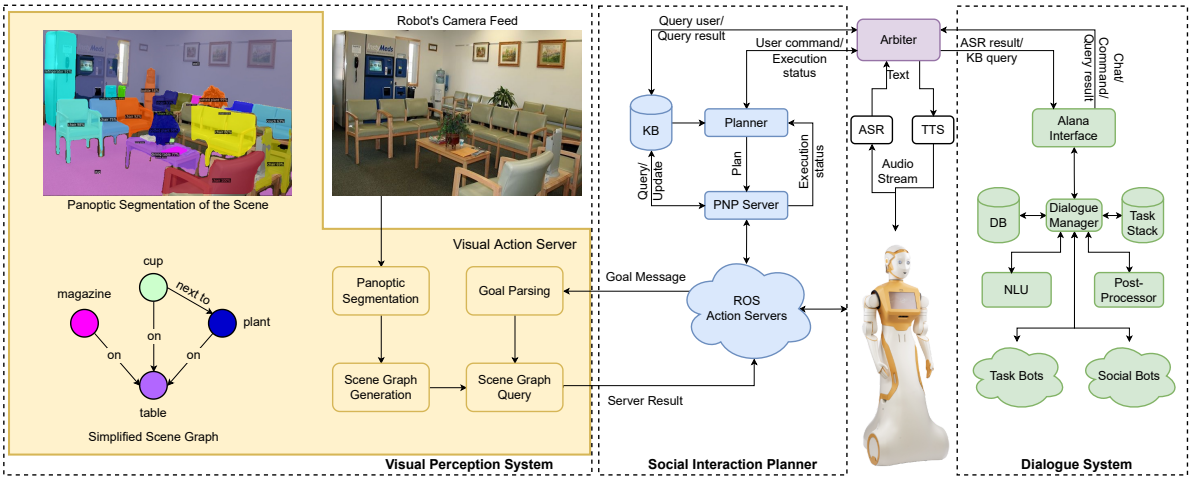


Figure 2: System Architecture. The Social Interaction Planner (blue blocks) interacts with the Dialogue System (green blocks) through an Arbiter module. The Vision module (yellow blocks) is implemented as a ROS Action Server within the planning framework. Based on the user's intent, the Planner can recruit the vision module to respond to questions about the visual scene.

and task-based conversation with visual dialogue regarding navigation and object detection in the shared space. Our system greets visitors, supports them to check-in, answers FAQs and helps users to locate key facilities and objects. It also offers social support/entertainment in the form of chit-chat, a quiz and access to the latest news.

2 System Architecture

The system architecture (Fig. 2), is composed of three main modules; a visual perception system, a dialogue system, and a social interaction planner.

2.1 Visual Perception System

The visual perception system is implemented as a ROS action server and is based on scene segmentation (Wu et al., 2019) from Facebook's Detectron2 framework[2]. From the segmented scene, the goal is to build a scene graph to capture relationships between objects such as location, adjacency, etc.

2.2 Dialogue System

The Dialogue System (Fig. 3) is based on the Alana system (Curry et al., 2018), an ensemble of different bots that compete in parallel to produce a response to user input. There are two types of bot: rule-based bots that can, for example, drive the conversation if it stalls, and express the identity of a virtual 'persona' (e.g. answering questions about the robot's age etc.); and data-driven bots that can retrieve replies from various information sources, e.g., News feeds. The SPRING system retains the rule-based and News bots, supplemented with a number of new, domain-specific bots. *Visual Task Bot* handles visual dialogue within the conversation, converting the user's inferred intent and any entities associated with it to a goal message that is forwarded to the visual action server (Part et al., 2021). *Reception Bot* welcomes visitors, helps them check-in and answers FAQs on, e.g., catering facilities and schedules. *Directions Bot* helps users find key facilities such as the bathrooms and elevator, while *Quiz Bot* is a simple true-or-false game designed to keep users entertained while they wait. The Dialogue Manager decides which response the robot verbalises based on a bot priority list. Automatic Speech Recognition (ASR) on the robot is currently implemented (in English) via Google Cloud[3]. Natural Language Understanding (NLU) is based on the original Alana pipeline, augmented using the RASA framework[4] for the parsing of domain-specific enquiries. Quiz bot employs regex-based intent recognition. Natural Language Generation (NLG) for the majority of bots consists of templates, with only News bot retrieving content from selected online sites. The utterances are voiced on the robot by Acapela's UK English Text-To-Speech voice 'Rachel'[5].

[2] https://github.com/facebookresearch/detectron2

[3] https://cloud.google.com/speech-to-text

[4] https://rasa.com

[5] https://www.acapela-group.com/

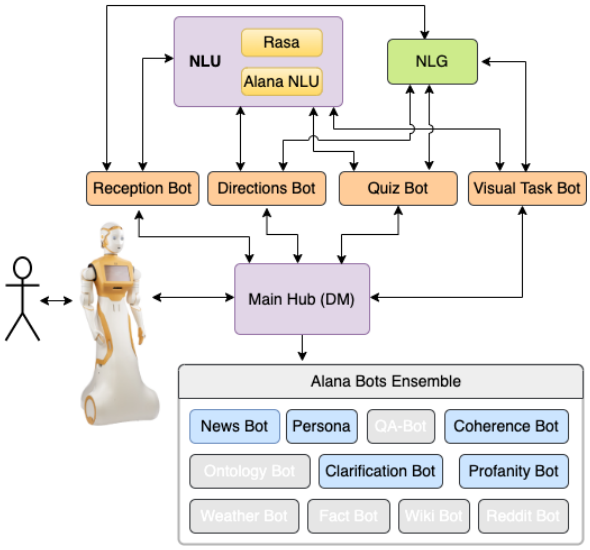


Figure 3: Dialogue System Architecture

2.3 Social Interaction Planner

The Social Interaction Planner interfaces the dialogue, vision systems, and the physical actions of the robot. It creates and executes plan(s) containing dialogue, physical, and perception actions based on the current dialogue context, and is based on the ideas of (Papaioannou et al., 2018; Lemon et al., 2002), enabling multi-threaded task execution and dialogue that is flexible and pausable. As shown in Fig. 2 it comprises several components, with the *Arbiter* managing communication between the dialogue system, the robot, and the planner

The *Planner* is a key component and uses the principle of *recipes and resources* as developed by (Lemon et al., 2002) to eliminate the problems from (re-)planning and concurrent interactive planning and execution. A domain file lists all the possible actions and specifies their types and parameters including their preconditions and effects. Recipes then describe the sequence of actions (i.e. dialogue, physical, or perception) involved in achieving a desired goal. When requested by the Arbiter, these recipes are transformed into Petri-Net Plans (PNP) (Dondrup et al., 2019) and are concurrently executed together with any other plans that may already be running. At run-time, redundant actions whose effects have already been achieved are skipped, or repair actions are executed in cases where an action was unsuccessful. At any time, each action has the ability to communicate with the dialogue system via the Arbiter to allow for clarification or to communicate perception results.

3 The SPRING ARI Robot

ARI is a humanoid robot, designed for use as a socially assistive companion (Cooper et al., 2020). It is 1.65m tall, has a mobile base, a touch-screen on the torso, movable arms and head with LCD eyes that enable gaze behaviour. The version of the robot used here is equipped with several cameras creating a 360º field of view. For audio capture and processing a ReSpeaker Mic v2.0 array, with 4 microphones is mounted on the front of the belly.

4 Example Dialogues

The example dialogue shown in Table 1 illustrates the system's visual dialogue capabilities (in bold), occurring as part of the welcome conversation.

U: Hello, I've got an appointment at 10am.
R: [reception_bot] I can help you with that. May I take the name please?
U: It's Jeanne X.
R: [reception_bot] Pleased to meet you, Jeanne. Is this your first visit?
U: No, I've been before.
R: [reception_bot] OK great, you'll be an expert then! That's you checked in. Please take a seat and a nurse will call you when they're ready. I'm here if you need anything else while you're waiting.
U: **Is there something to read while I wait?**
R: [visual_task_bot] **I can see a magazine on the table.**
U: Aha thanks.
R: No problem.

Table 1: Task-based Dialogue with Visual Dialogue.

The example dialogue in Table 2 further illustrates the robot's natural language social and task-based capabilities.

U: Excuse me, do you know how much longer I will have to wait?
R: [reception_bot] I expect it won't be long now. A nurse will come and get you. Do you want to play a game to pass the time?
U: OK then, yes please.
R: [quiz_bot] Great, let's play! Please remember, whenever you want to leave the game, just say 'end game'. First question, is Covid-19 on all 7 continents?
U: Nope.
R: Correct. Next question .....
R: [quiz_bot] Congratulations! You won. That was fun. Now, please let me know if you need anything else.
U: I need the bathroom, can you tell me where it is?
R: [directions_bot] To find the nearest bathroom, leave the waiting room and go right towards the door with two round windows. The bathroom will then be on your left.

Table 2: Task-based Dialogue with Social Dialogue.

5 Conclusions and Future Work

The proposed system demonstrates how to successfully weave together a wide range of task-based, social, and visually grounded dialogue and physical actions on an SAR in a receptionist environment. Next steps are to generate the scene graphs automatically by combining data-driven approaches (Zellers et al., 2018; Yang et al., 2018; Zhang et al., 2019; Tang et al., 2020) with prudent use of refining rules. Crucially also, we are working on extending the system to handle multi-party interactions, an active area of research and highly likely to occur in this context.

For the demonstration, we will showcase our system on the ARI robot, inviting attendees to interact with it and experience all the capabilities of the system described in this paper.

Acknowledgements

This research has been funded by the EU H2020 program under grant agreement no. 871245 (http://spring-h2020.eu/).

References

Sara Cooper, Alessandro Di Fava, Carlos Vivas, Luca Marchionni, and Francesco Ferro. 2020. ARI: The Social Assistive Robot and Companion. In *29th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2020*, pages 745–751.

Amanda Cercas Curry, Ioannis Papaioannou, Alessandro Suglia, Shubham Agarwal, Igor Shalyminov, Xinnuo Xu, Ondřej Dušek, Arash Eshghi, Ioannis Konstas, Verena Rieser, et al. 2018. Alana v2: Entertaining and Informative Open-Domain Social Dialogue using Ontologies and Entity Linking. *Alexa Prize Proceedings*.

Christian Dondrup, Ioannis Papaioannou, and Oliver Lemon. 2019. Petri Net Machines for Human-Agent Interaction.

Carina Soledad González-González, Verónica Violant-Holz, and Rosa Maria Gil-Iranzo. 2021. Social robots in hospitals: A systematic review. *Applied Sciences*, 11(13).

Oliver Lemon, Alexander Gruenstein, Alexis Battle, and Stanley Peters. 2002. Multi-Tasking and Collaborative Activities in Dialogue Systems. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, pages 113–124.

Maria R. Lima, Maitreyee Wairagkar, Manish Gupta, Ferdinando Rodriguez y Baena, Payam Barnaghi, David J. Sharp, and Ravi Vaidyanathan. 2021. Conversational affective social robots for ageing and dementia support. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–1.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios P. Spithourakis, and Lucy Vanderwende. 2017. Image-Grounded Conversations: Multimodal Context for Natural Question and Response Generation. pages 462–472.

Irena Papadopoulos, Runa Lazzarino, Syed Miah, Tim Weaver, Bernadette Thomas, and Christina Koulouglioti. 2020. A systematic review of the literature regarding socially assistive robots in pre-tertiary education. *Computers Education*, 155:103924.

Ioannis Papaioannou, Christian Dondrup, and Oliver Lemon. 2018. Human-Robot Interaction Requires More Than Slot Filling - Multi-Threaded Dialogue for Collaborative Tasks and Social Conversation. In *Proceedings of the FAIM/ISCA Workshop on Artificial Intelligence for Multimodal Human Robot Interaction*, pages 61–64.

Jose L. Part, Daniel Hernández García, Yanchao Yu, Nancie Gunson, Christian Dondrup, and Oliver Lemon. 2021. Towards Visual Dialogue for Human-Robot Interaction. In *Companion Proceedings of the 16th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 670–672, Boulder, CO, USA.

Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased Scene Graph Generation from Biased Training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3713–3722.

Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. 2018. Graph R-CNN for Scene Graph Generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 670–685.

Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. 2018. Neural Motifs: Scene Graph Parsing with Global Context. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5831–5840.

Ji Zhang, Kevin J. Shih, Ahmed Elgammal, Andrew Tao, and Bryan Catanzaro. 2019. Graphical Contrastive Losses for Scene Graph Parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11535–11543.

Li Zhou, Jianfeng Gao, Di Li, and Heung Yeung Shum. 2020. The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.

Demonstrating EMMA: Embodied MultiModal Agent for Language-guided Action Execution in 3D Simulated Environments

Alessandro Suglia, Bhathiya Hemanthage, Malvina Nikandrou, Georgios Pantazopoulos, Amit Parekh, Arash Eshghi, Claudio Greco, Ioannis Konstas, Oliver Lemon, Verena Rieser

{a.suglia, hsb2000, mn2002, gmp2000, amit.parekh, a.eshghi, c.greco, i.konstas, o.lemon, v.t.rieser}@hw.ac.uk

Abstract

We demonstrate EMMA, an embodied multimodal agent which has been developed for the Alexa Prize SimBot Challenge[1]. The agent acts within a 3D simulated environment for household tasks. EMMA is a unified and multimodal generative model aimed at solving embodied tasks. In contrast to previous work, our approach treats multiple multimodal tasks as a single multimodal conditional text generation problem. Furthermore, we showcase that a single generative agent can solve tasks with visual inputs of varying length, such as answering questions about static images, or executing actions given a sequence of previous frames and dialogue utterances. The demo system will allow users to interact conversationally with EMMA in embodied dialogues in different 3D environments from the TEACh dataset.

1 Introduction

Robots that perform tasks in human spaces can benefit from natural language interactions that provide both high and low-level instructions, as well as the ability to resolve ambiguities. The Alexa Prize SimBot Challenge aims to propel research efforts to develop embodied agents that learn to execute household tasks from instructions, such as *"Please clean all the tableware"*.

Transformers (Vaswani et al., 2017) coupled with joint vision-and-language pretraining have become the standard approach for tasks with single image inputs, where available object-detectors are used produce image features. We demonstrate how this approach can also benefit embodied agents for object manipulation tasks. While representing the scene in terms of object representations (object-centric) can also benefit embodied agents performing tasks involving object manipulation, this approach is not as widely adopted due to the increased computational overhead.

To complete a task, an embodied agent may be required to perform multiple successive actions. Each predicted action is conditioned on all previous observations that yields a new observation. From an object-centric point-of-view, each observation corresponds to a set of detected objects which must remain accessible by the agent to predict the next action. Therefore, even for smaller action trajectories, the resulting input length can become prohibitively large as the number of frames increases.

In this work, we present Embodied MultiModal Agent (EMMA), a language-enabled embodied agent capable of executing actions conditioned on historical dialogue interactions. To address the long-horizon input, we adopt advances from tasks involving processing long-documents (Beltagy et al., 2020). Existing embodied agents in similar environments treat action prediction as a classification task (Suglia et al., 2021; Pashevich et al., 2021). On the other hand, EMMA is a unified, visually-conditioned, autoregressive text generation model that accepts visual (observations) and textual (dialogue) tokens as input, and produces natural language text and executable actions.

2 Background

TEACh The Task-driven Embodied Agents that Chat (TEACh) dataset (Padmakumar et al., 2021) consists of gameplay sessions where two participants must complete household tasks in the AI2-THOR simulator (Kolve et al., 2017). Each session consists of a *Commander* with oracle information, and a *Follower* that interacts with the environment and communicates with the *Commander* to complete the task. This work focuses on Execution from Dialogue History (EDH), which is the reference task for the Alexa Prize SimBot Challenge. EDH instances are created by segmenting game sessions. Each instance is defined by an initial state S^E, action history A_H, set of interaction actions during the session A_I^R, and the goal environment

[1] https://amazon.science/alexa-prize/simbot-challenge

Proceedings of the SIGdial 2022 Conference, pages 649–653
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

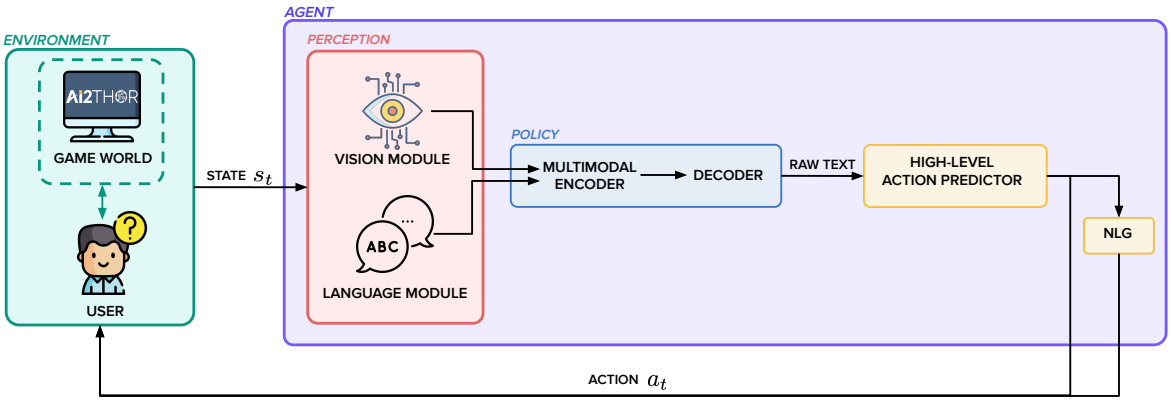


Figure 1: High-level architecture of EMMA. The *Perception* component processes new visual and language input at each timestep. Both streams are then processed by the *Policy* component to output raw text, which is mapped to actions that are executable in the environment. The resulting action a_t can be either a physical action or text (as utterances generated from the dedicated NLG component).

state F^E. The agent models the *Follower* who has to generate the actions leading to the goal state.

Training and Evaluation During training, the A_I^R are used for supervision. At inference time, the model is expected to generate a sequence of interaction actions which would result in F^E. The model is evaluated by comparing the simulator state resulted from inferred actions against F^E.

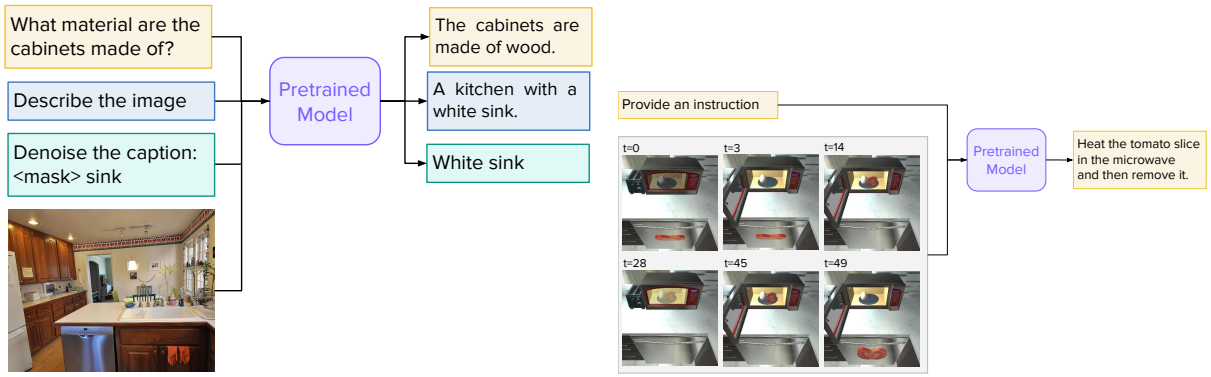
3 System Architecture

As shown in Figure 1, EMMA consists of three components: *Perception*, *Policy*, and *Action Predictor*. At each timestep, the agent generates the next action after receiving information regarding the current and previous states of the environment—including any executed actions and interactions. The agent receives a new observation and has to predict a follow-up action. The process is repeated until the agent outputs a stop action.

Perception This module is responsible for processing the state of the environment—encoding past actions, frames, and dialogue to create the model input. The current state for the EDH task consists of observations obtained after executing an action, or a dialogue utterance from the *Follower* or *Commander*. We extract local and global information from the visual scenes using the VinVL object detector (Zhang et al., 2021), after fine-tuning on the ALFRED images (Shridhar et al., 2020). From each scene, we obtain up to 36 regional features. We obtain the global representation as the mean pooled features from the backbone of the detector. In the second case, the dialogue utterance is concatenated with the dialogue history. We include special tokens to distinguish between *Follower* and *Commander* utterances.

Policy The core component of EMMA is a unified autoregressive text generation model. Given the current state, the previous observations and interactions, the model generates raw textual output. Assuming the input sequence consists of V frames—with each encoded into N_V scene and object tokens—and L language tokens, the total sequence length $V \times N_V + L$ will be dominated by the number of visual tokens. To reduce the impact of having a large V, we adapt the sparse attention pattern following Beltagy et al. (2020). Each token attends to its neighbouring tokens within a local window, and a subset of tokens are regarded as global to aggregate information from longer contexts. Global tokens act as a bottleneck of relevant information over the entire sequence. These tokens can attend to, and are attended by, all other tokens in the input sequence under causal masking.

To infuse our agent with knowledge about objects and their properties, we pretrain the model several image-text and video-text tasks. We use COCO (Lin et al., 2014), VisualGenome (Krishna et al., 2016), and GQA (Hudson and Manning, 2019) to learn an alignment between language and vision. Furthermore, we incorporate ALFRED (Shridhar et al., 2020), and EPIC-KITCHENS (Damen et al., 2018), two video-based datasets involving action execution and recognition to enable temporal reasoning.



(a) Output for three different tasks when given the same image.

(b) Output for the Instruction Prediction task, given an input video from the agent's trajectory.

Figure 2: Example of generated output for various pretraining tasks, showing how EMMA can be prompted for the task using Natural Language prefixes.

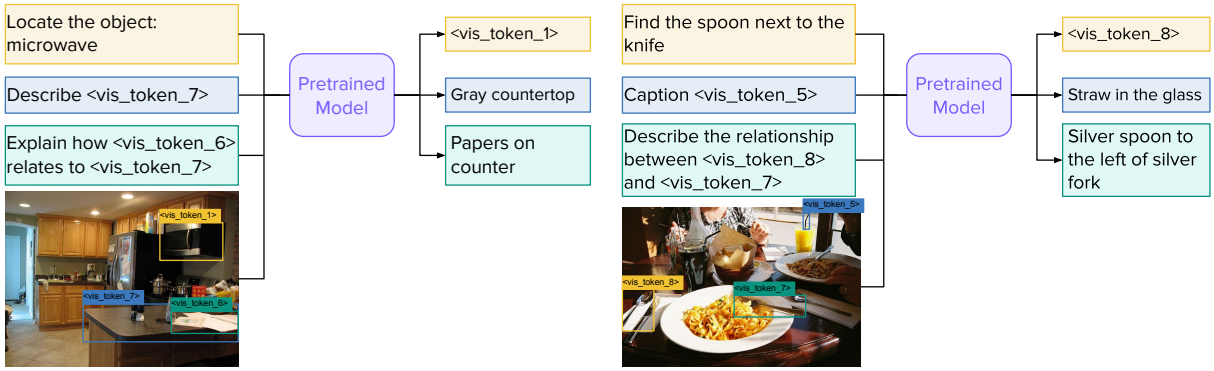


Figure 3: Example of generated output for pretraining tasks showing the use of visual tokens in order to reference specific objects. Visual tokens follow the format <vis_token_i> to refer to the i-th predicted bounding box.

Action Predictor The final component of EMMA is responsible for converting generated raw text into actions which are executable in the environment. We parse the raw text and map it to either a navigation (e.g., Forward) or interaction action (e.g., Pickup Mug). For interaction actions, we also select the associated object using its coordinates available from the *Perception* module.

4 System Demonstration

We demonstrate the ability of our model to solve several downstream tasks ranging from captioning to embodied action execution after casting all tasks into the same sequence-to-sequence framework. After training EMMA, we can use natural language *task prompts* to trigger specific behaviours, following literature on prompting for text-only models (Raffel et al., 2020; Brown et al., 2020).

4.1 Pretraining Tasks

Figures 2-3 show examples of outputs generated for various pretraining tasks. Figure 2a illustrates outputs of a model with the same weights for three image-based tasks: Visual Question-Answering (VQA), Image Captioning, and Masked Language Modelling (MLM). Figure 3 demonstrates the pretraining tasks that require referencing specific objects in the image: Visual Grounding, Dense Captioning and Relationship Detection. Without any special task-specific tokens, EMMA can infer the target task to generate summary descriptions for images, and can also respond to queries regarding attributes of specified objects. Figure 2b shows an example of a video pretraining task using a trajectory from the ALFRED (Shridhar et al., 2020) dataset. Given the task prefix "Provide an instruction" and a sequence of frames, EMMA learns to generate an high-level description of the action trajectory.

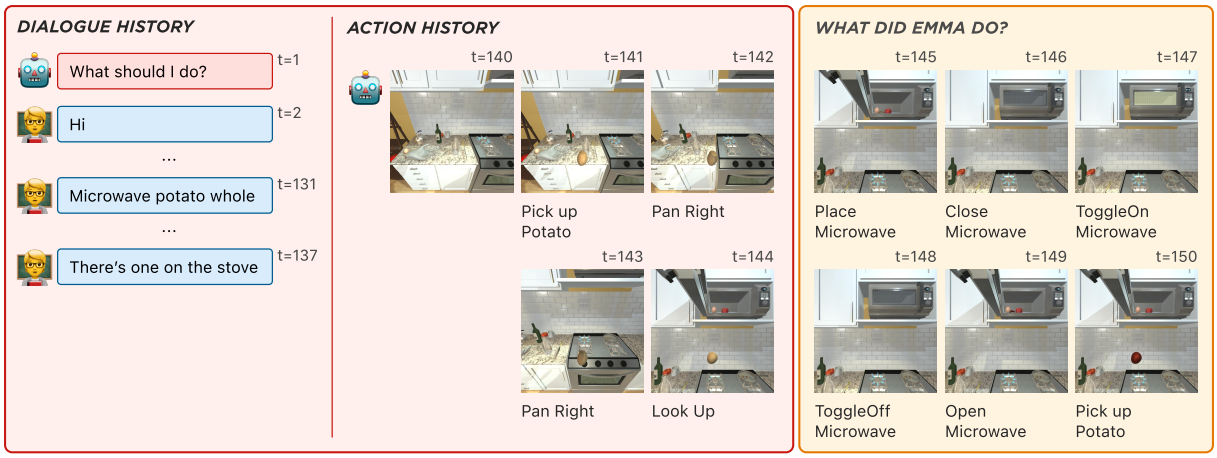


Figure 4: Example of action execution in the AI2Thor 3D environment. EMMA conditions the action generation on both the dialogue and the visual history.

4.2 Action Execution

Figure 4 provides an example of action execution from dialogue history using an episode from the TEACh dataset. The goal of the episode is to microwave a potato. The initial input to the model consists of the dialogue between the *Commander* and the *Follower* as well as the frames corresponding to the previously executed actions. Up to that point, the *Commander* has expressed the end goal and helped the agent locate a potato. Based on this input, EMMA executes a sequence of actions that successfully complete the task. At each step the initial input is augmented with the agent's egocentric observation after executing the most recent action. The process is repeated until the timestep 49, where EMMA predicts a stop action. For this particular example, the human follower completed the task in 10 steps including redundant actions such as looking up and down. EMMA's action trajectory is more efficient than the human demonstration by performing only the necessary actions.

5 Conclusion

In this work we presented EMMA, an embodied agent that learns to execute actions from dialogue, developed for the Alexa Prize SimBot Challenge. EMMA is based on a unified text generation model that is pretrained on multiple image and video-based tasks using natural language prompts. We will provide a conversational web-based demonstration of interaction with EMMA in 3D environments.

References

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*.

Drew A Hudson and Christopher D Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Kumar Gupta, and Ali Farhadi. 2017. AI2-THOR: An interactive 3d environment for visual ai. *ArXiv*, abs/1712.05474.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2021. TEACh: Task-driven embodied agents that chat. *arXiv preprint arXiv:2110.00534*.

Alexander Pashevich, Cordelia Schmid, and Chen Sun. 2021. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alessandro Suglia, Qiaozi Gao, Jesse Thomason, Govind Thattai, and Gaurav Sukhatme. 2021. Embodied BERT: A transformer model for embodied, language-guided visual task completion. *arXiv*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. VinVL: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588.

GRILLBot: An Assistant for Real-World Tasks with Neural Semantic Parsing and Graph-Based Representations

Carlos Gemmell, Iain Mackie, Paul Owoicho, Federico Rossetto
{c.gemmell.1, i.mackie.1, p.owoicho.1, f.rossetto.1}@research.gla.ac.uk

Sophie Fischer, Jeffrey Dalton
{sophie.fischer, jeff.dalton}@glasgow.ac.uk

University of Glasgow
Glasgow, Scotland, UK

Abstract

GRILLBot is the winning system in the 2022 Alexa Prize TaskBot Challenge, moving towards the next generation of multimodal task assistants. It is a voice assistant to guide users through complex real-world tasks in the domains of cooking and home improvement. These are long-running and complex tasks that require flexible adjustment and adaptation. The demo highlights the core aspects, including a novel Neural Decision Parser for contextualized semantic parsing, a new "TaskGraph" state representation that supports conditional execution, knowledge-grounded chit-chat, and automatic enrichment of tasks with images and videos.

1 Introduction

We present GRILLBot, a task-oriented multimodal conversational assistant developed during the 2021/2022 Alexa Prize TaskBot Challenge (Gemmell et al., 2022). GRILLBot aims to be an open research platform for complex tasks and supports flexible graph-based task representations, contextual semantic parsing, and incorporates image and video content for clarity and instruction. We release the core components of the system as OAT[1] (Open Assistant Toolkit).

GRILLBot is still deployed throughout the United States with users able to invoke the bot by issuing the command "Hey Alexa, Assist me" to their voice-only or screened Alexa device. Our system provides open-ended assistance focusing in the domains of cooking and home improvement. It guides the user through all phases of the task, from performing preference elicitation to guiding a user to a relevant task from large task corpora, (i.e. "making a New York-style pizza" or "how to paint a wall") and then proceeds to assist in executing the task in an engaging way. Its capabilities include question answering, task-oriented chit-chat, and instructional video content.

GRILLBot is part of the first generation of assistants (Ipek et al.) that leverage screen-enabled conversational devices for complex real-world tasks. These tasks are extensive, with some taking over an hour. As a result, a performant system requires long-term state tracking with capabilities to adapt to a changing environment. It achieves this by introducing a new novel task structure, a *TaskGraph*, that captures the actions and information dependencies to guide the user through a complex task. TaskGraphs are enriched offline with content from information extraction, knowledge-based content, and multimedia images and videos.

Traditional task-oriented dialogue systems (Young et al., 2013) take a slot-filling approach to deriving system actions. Academic datasets such as MultiWOZ (Budzianowski et al., 2018) capture slot-value pairs from the user utterances within a constrained set of domains enabling data-driven neural models. Andreas et al. (2020) extend this traditional representation towards semantic parsing with dataflow graphs while constrained to the domain of events booking in the SMCalFlow dataset. The neural decision parser in GRILLBot similarly generates code but focused on all aspects of a conversation from navigation to task search and question answering. Other challenges such as DSTC11 (Kottur et al., 2021) attempt this fully featured task-oriented experience, yet only do so in a virtual setting. GRILLBot stands apart as a system required to engage with real-world users in their environment and assist in complex tasks for cooking and home improvement.

2 System overview

The system uses a micro-service architecture with a centralized *Orchestrator* that defines the system behavior. We use a phase-based policy to transition

[1] https://github.com/grill-lab/OAT

Proceedings of the SIGdial 2022 Conference, pages 654–658
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

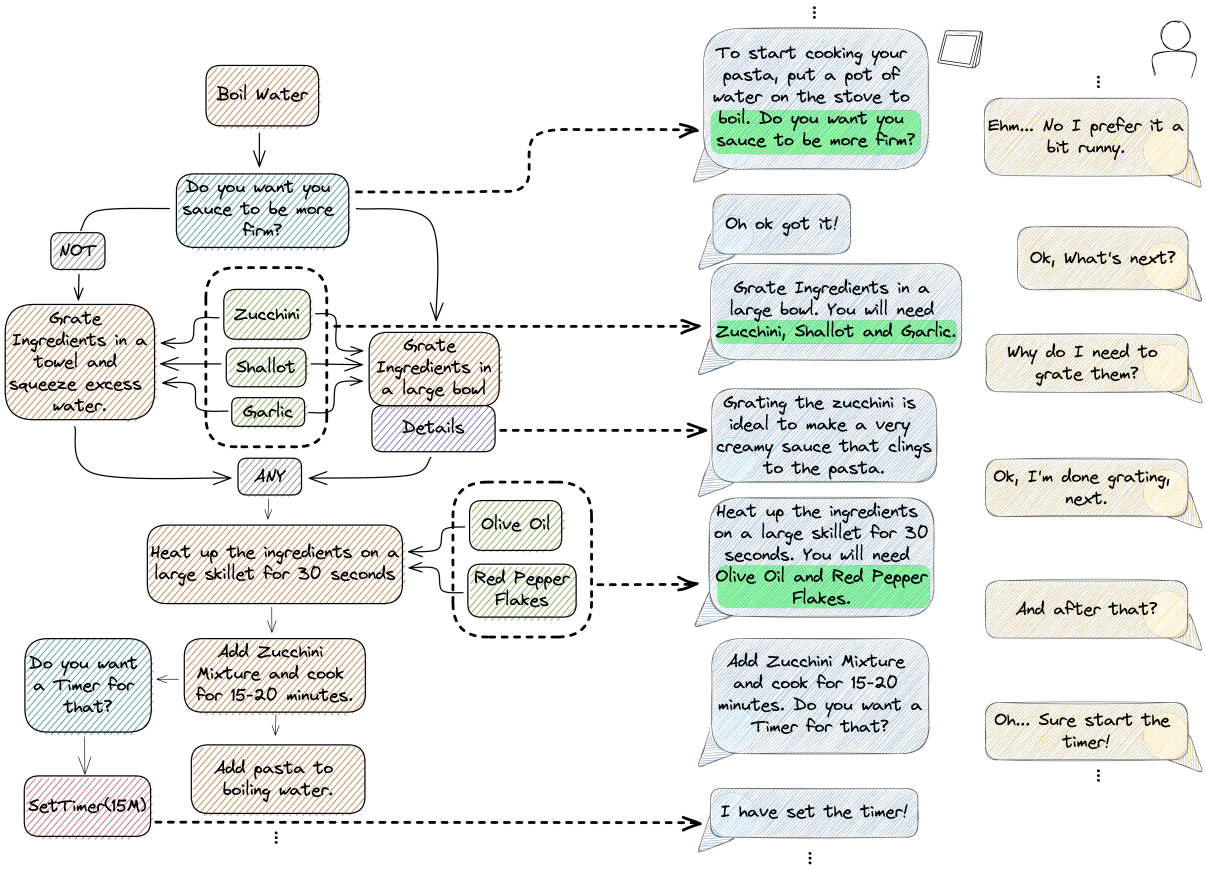


Figure 1: Example TaskGraph (right) and conversation (left) connect each utterance with the information in the graph nodes. The figure shows how we use *conditional nodes* (in light blue) to manage yes/no questions to unlock different branches of instructions. Conditional nodes can also unlock autonomous actions like setting a timer. The figure also highlights how the *requirement nodes* (in light green) are used by the system to enrich the experience by adding specific information that the user will need to perform the step. Finally, the purple box highlights extra information contained inside the *step nodes* to ground the QA system in domain knowledge.

from searching for a suitable task (i.e. *planning phase*) to guiding users in performing a task (i.e. *execution phase*).

The *Orchestrator* is the central process that directs and receives all the information from other microservices. Specifically, these child components are called *functionalities* and provide the necessary tools required by policies during conversations. The main components inside functionalities are: *Neural Decision Parser*, *Task Searcher* and *Question Answering*. We discuss the *Neural Decision Parser* in Section 4.

The *task searcher* leverages our collection of TaskGraphs to find candidate tasks. Our approach is based on a combination of traditional sparse & dense retrieval and neural re-ranking (Gemmell et al., 2022). The *question answering system* provides extra task information, handles user questions, and provides chit-chat elements. It uses a collection of QA systems across six categories.

3 TaskGraphs

A *TaskGraph* is a new graph-based representation based on a directed acyclic graph that encodes the actions and information dependencies that the system needs to enable complex dialogue flows. Information is represented with heterogeneous nodes, each with a specific role:

- **Steps**: Represent a task instruction for the user, including visual information and textual descriptions.
- **Requirements**: Represent tools and ingredients that are needed to perform the task and can be grounded to specific steps.
- **Conditions**: Represents yes/no gates that require external information to resolve during execution dynamically.
- **Logic**: Represents logical operations. These can be used in conjunction with other nodes to

enable compact dependencies and smoother execution flows. We currently support $\wedge$, $\vee$ and $\neg$ operations.

- **Actions**: Represents actions taken by the system. This could be operations like setting a timer or adding items to a list.

- **Extra Information**: Represents domain/task-specific knowledge like tips or fun facts that can enrich the user experience during the execution of the task.

Combining all these nodes, we can obtain adaptive interactions where system-initiative allows the system to adapt to the user's needs. Figure 3 shows how TaskGraphs can be leveraged using a task-oriented conversation helping a user cooking "creamy zucchini pasta."

3.1 Offline TaskGraph Curation

A key part of the system is providing relevant and high-quality TaskGraphs that satisfy the user's task goal. For example, if a user asks, "I want to cook a gluten-free meal based around lamb shoulder", the system must find a suitable TaskGraph.

To enable this, the system has to process rich and executable TaskGraphs offline with enough scale to cover most user needs. Offline processing also decouples the heavy processing stages and data enrichment from online processing.

Web content We leverage domain experts to identify high-quality seed websites for each domain, e.g. `wholefoodsmarket.com` and `seriouseat.com` for cooking and `wikihow.com` for home improvement. We use Common Crawl to download the raw HTML for target domains and develop website-specific wrappers to extract semi-structured information about each task, i.e. title, author, description, ingredients, images, steps, ratings, videos, infoboxes, FAQs.

Synthesize TaskGraphs The next stage of the offline process takes the semi-structured information extracted and synthesizes executable TaskGraphs. This creates multi-modal task nodes and connections from previously linear task steps. For example, we can create expressive graphs that contain a summary and a detailed description for each task step, which can be accessed by users who require additional context. We also leverage information extraction methods, such as noun phrase detection (Honnibal and Montani, 2017), to create graph connections that link required ingredients and tools to each step. Additionally, complex graph structure and manual augmentation can be added using a custom-developed *excalidraw.com* graph interface. This allows loading automatically processed TaskGraphs, adding additional graph nodes and connections, and exporting the updated TaskGraphs.

Multimodal augmentation Visual information plays a crucial role in improving the success and enjoyment of users being guided through real-world tasks. For example, showing "How-to" videos, images and lists of tools and ingredients offers a more compelling and useful user experience. Figure 2 depicts the multi-modal experience where the screen text outlines the instruction, a list shows the ingredients required, an image enriches the user experience, and a video offers a technical demonstration.

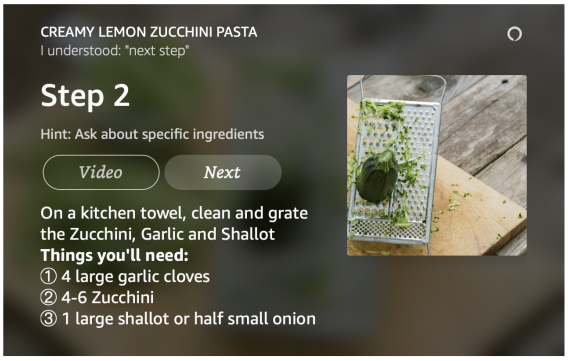


Figure 2: Multimodal UI containing text, buttons, images, and videos.

We also develop a means of enriching task nodes if the task steps do not have aligned images and videos. First, we extract actions (i.e "cut the beef") from a step based on a dependency parse of the step text using the spaCy toolkit. For images, we use CLIP (Radford et al., 2021) to search over an image corpus of all other task steps images. This uses the cosine similarity between image and step action embeddings to identify the relevant images for each step. For videos, we develop a video corpus of domain-focused techniques, which is an index based on the video title using S-BERT (Reimers and Gurevych, 2019). Similar to image retrieval, we embed the step action as a query and rank the titles of each "How-To" video through a cosine similarity of the step action embedding.

4 Neural Decision Parser

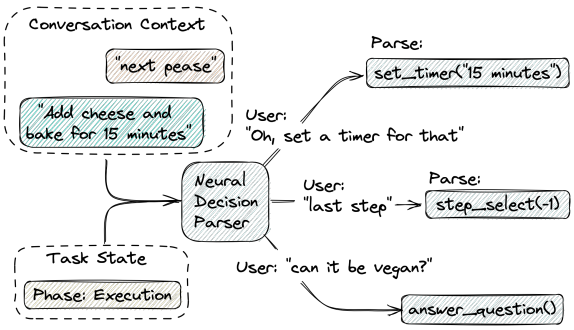


Figure 3: Example of several possible in-context parses during task execution. The Neural Decision Parser autoregressively generates the function call and arguments as code in our DSL.

TaskGraphs allows complex representations of real-world tasks. Due to the complex conversational dialogue required, traditional non-contextual intent classifiers struggle to manage stateful transitions. For this reason, we develop a *Neural Decision Parser* that leverages both TaskGraphs and user history for contextualized semantic parsing. Specifically, the model takes in natural language representations of a TaskGraph and prior conversational context to generate actions in the form of the custom GRILLBot Domain Specific Language (DSL). These generated arguments supply the task sub-components with parsed knowledge relating to the conversation.

For example, if a user asks "Can you go back to the first step?", the Neural Decision Parser would generate a parameterized parse `step_select(1)`.

Contextual Semantic Parsing as a DSL Figure 4 shows our state transition domain specific language (DSL) that captures all system actions. This DSL outlines a parameterized global command set that is understood throughout the system to derive what actions or external APIs should be called, and what the response utterance should be. This flexible navigation allows for a complex conversation design that leverages TaskGraphs.

Model We use a single T5 large model (Raffel et al., 2020) to generate an agent action based on the TaskGraph and conversational content. Using a pre-trained language model allows advanced language capabilities to be leveraged across all system parts, including coreference resolution, search parameterization, setting timers, and state prediction.

```
# User specifies which task to execute
> select(option=Int)

# Catch all for user questions
> answer_question()

# Catch all for task search
# Vague and Theme query categories
> search(vague=Bool, theme=String)

# Go to prior node
> previous()

# Go to next scheduled node
> next()

# Navigate to specific task steps
> step_select(step=Int)

# Set timer with parsed time span
> timer(span=String)

# Provide details about a step
> chit_chat()
```

Figure 4: A sample of the Neural Decision Parser output DSL with intent-based functions and parameterized arguments that a T5 model generates at inference time.

We train the Neural Decision Parser by annotating simulated conversations with the appropriate function calls and associated arguments. Our annotated training data comprises 1,200 turns across various conversation stages and includes TaskGraph and conversational context. Through user studies and system comparisons, we find that this approach achieves strong performance, and allows flexible task navigation.

5 Conclusion

This demo presents GRILLBot, a newly developed Alexa Prize Taskbot system for complex real-world tasks with rich multimodal capability. It demonstrates multiple novel components including TaskGraphs to manage long complex tasks that are automatically enriched with offline process to add multimodal image and instructional video content. It also shows key elements of the system that make it engaging, including its flexible Neural Decision Parser that performs contextual semantic parsing as parametrized code generation. The result is a demonstration of a new research platform designed from the ground-up around around flexible cloud micro-services and large-scale neural language models.

References

Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, et al. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on EMNLP*, pages 5016–5026.

Carlos Gemmell, Sophie Fisher, Iain Mackie, Paul Owoicho, Federico Rossetto, and Jeffrey Dalton. 2022. Grillbot: A flexible conversational agent for solving complex real-world tasks. In *2022 Alexa Prize Proceedings*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Anna Gottardi Osman Ipek, Giuseppe Castellucci Shui Hu, Lavina Vaz Yao Lu, Anju Khatri, Anjali Chadha, Desheng Zhang, Sattvik Sahai, Prerna Dwivedi, Hangjie Shi, Lucy Hu, et al. Alexa, let's work together: Introducing the first alexa prize taskbot challenge on conversational task assistance.

Satwik Kottur, Seungwhan Moon, Alborz Geramifard, and Babak Damavandi. 2021. SIMMC 2.0: A task-oriented dialog dataset for immersive multimodal conversations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4903–4912, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of ML Research*, 21:1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

A System For Robot Concept Learning Through Situated Dialogue

Benjamin Kane*
University of Rochester
bkane2@ur.rochester.edu

Felix Gervits
DEVCOM Army Research Laboratory
felix.gervits.civ@army.mil

Matthias Scheutz
Tufts University
matthias.scheutz@tufts.edu

Matthew Marge
DEVCOM Army Research Laboratory
matthew.r.marge.civ@army.mil

Abstract

Robots operating in unexplored environments with human teammates will need to learn unknown concepts on the fly. To this end, we demonstrate a novel system that combines a computational model of question generation with a cognitive robotic architecture. The model supports dynamic production of back-and-forth dialogue for concept learning given observations of an environment, while the architecture supports symbolic reasoning, action representation, one-shot learning and other capabilities for situated interaction. The system is able to learn about new concepts including objects, locations, and actions, using an underlying approach that is generalizable and scalable. We evaluate the system by comparing learning efficiency to a human baseline in a collaborative reference resolution task and show that the system is effective and efficient in learning new concepts, and that it can informatively generate explanations about its behavior.

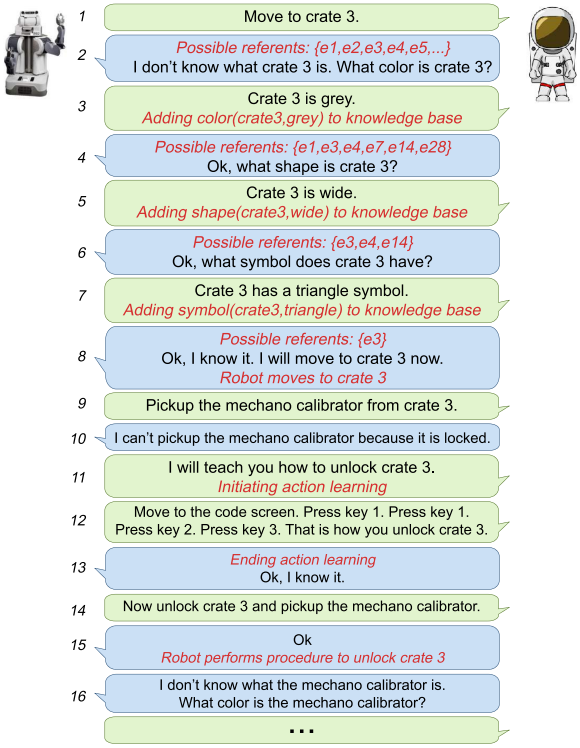


Figure 1: Example dialogue between a human and our system situated in an unexplored spacecraft environment, where the robot must learn new locations, objects, and actions through interaction with the human. The system's behavior is indicated in red.

1 Introduction

An autonomous robot interacting with a human teammate will often encounter concepts[1] that are unfamiliar to it. This is especially problematic when exploring new environments where training data is naturally limited or expensive to collect and label. For example, a robot for space exploration may encounter objects or need to carry out procedures that do not exist on Earth, and for which data is scarce. A natural approach to online learning in many environments is to take advantage of the human-robot interface, enabling the robot to engage in multi-turn dialogue where it asks the human questions to learn about a novel concept.

Many of the foundations of this approach – e.g., robot architectures capable of engaging in natural language dialogue, and mechanisms for conversational grounding and question generation – have previously been explored, but were designed as piecemeal contributions, leaving a gap in the overall problem of learning concepts through dialogue.

In this work, we demonstrate a generalizable cognitive robotic system that is able to efficiently learn about unknown concepts through interactive natural language dialogue. This system leverages a probabilistic decision network model[2] to dynam-

*Work performed during a summer position at the Army Research Laboratory.

[1]The term 'concept' in this paper refers to any entity in the task domain, including objects, locations, and actions.

[2]Our decision network model, as well as the HuRDL dataset used to evaluate the system in Section 4, can be found at the following URL: https://github.com/USArmyResearchLab/ARL-HuRDL

Proceedings of the SIGdial 2022 Conference, pages 659–662
Heriot-Watt University, Edinburgh, UK. 07-09, September, 2022

ically generate and ask optimal questions for concept learning within any environment, while also employing natural language capabilities and an explicit knowledge representation enabled by a cognitive robotic architecture. An example dialogue from our system is shown in Figure 1.

2 Background

Early work in robot concept learning through dialogue explored the use of pre-specified ontologies or graphical models to allow an agent to ask questions about objects in an environment (Lemaignan et al., 2012; Chai et al., 2018; Perera et al., 2018), or to learn actions through dialogue (She et al., 2014). Other work explores the use of proactive symbol grounding or pragmatic models for reference resolution (Williams et al., 2019; Arkin et al., 2020). In contrast to these studies, our work includes a notion of uncertainty and can scale to new task domains through dynamic adaptation of a decision network.

Recent work has built upon these approaches by introducing information-theoretic measures for selecting optimal questions. Skočaj et al. (2011) propose a robot that can ask questions about object properties that maximize information gain, and test the system using colors and shapes as properties. Deits et al. (2013) relatedly demonstrate a system that can instantiate templatic questions to minimize entropy of the robot's probabilistic symbol grounding function. Both approaches, however, rely on the use of a small fixed set of properties or question templates; we present a scalable approach that can generate questions from arbitrary properties.

3 System Design

Our system combines a decision network model for question selection (Gervits et al., 2021a) with the DIARC (Distributed Integrated Affect Reflection Cognition) robotic architecture (Scheutz et al., 2019) in order to enable interactive concept learning. The DIARC architecture, which follows a distributed, component-based design, allows for semantic parsing, introspection on knowledge, explanation generation, and support for one-shot learning of actions. The particular configuration of DIARC used by our system is shown in Figure 2. In the remainder of this section, we describe the primary components of this architecture.

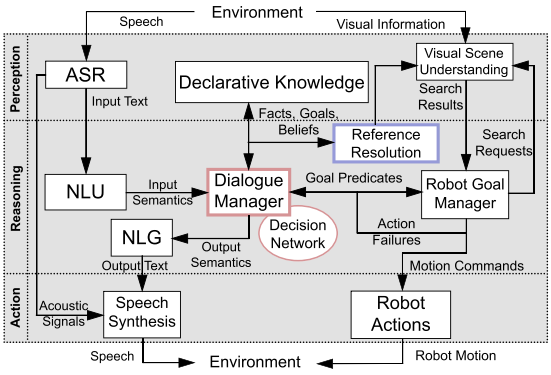


Figure 2: Architecture of the system's DIARC configuration. The core components that drive concept learning are the dialogue manager, which interacts with a decision network for question generation, and a reference resolution component for resolving concepts in user instructions to observed objects in the environment.

3.1 Decision Network

The dialogue manager component of our DIARC configuration is extended with a decision network model (Gervits et al., 2021a) that combines a Bayesian network with action and utility nodes. The model represents the robot's knowledge for a target referent and selects a question to help reduce ambiguity and acquire new concept knowledge.

Figure 3 shows a generic example of a decision network constructed by the system. The green boxes represent *chance nodes* which are random variables corresponding to the agent's knowledge of the object properties, the number of target referents, and the instruction. The blue diamond is a *utility node* which represents the utilities associated with asking questions from the red *decision node* conditioned on the chance nodes.

Since the robot's goal in asking a question is to reduce ambiguity (in the case of reference resolution, narrowing down the number of possible referents for a concept), the model selects a "best" question by calculating maximum expected utility from the model, with utilities set by calculating the Shannon entropy for each object property.

As shown by Gervits et al. (2021a), this approach is well-suited to dialogue learning in novel environments because the decision network is dynamically constructed for any novel environment given only observed object properties. Moreover, the network is constructed with the minimum set of nodes needed to disambiguate all entities in the environment, and can be re-constructed on the fly if new entities are discovered. This greatly enhances the

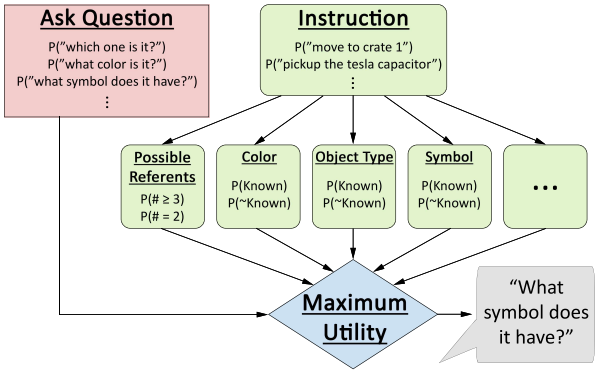


Figure 3: An instance of the decision network produced in our evaluation domain. The probabilistic chance nodes are shown in green. The red node represents the decisions available to the system, while the blue node represents the utilities associated with each decision, and outputs the decision with maximum expected utility.

flexibility of the approach, enabling it to generalize and scale to a variety of unexplored environments.

3.1.1 Semantic Parser and Declarative Knowledge

The NLU component uses a CCG grammar to map input text to a logical semantic representation[3], including the speech act type of the input (e.g., instruction or statement). The system is also able to use pragmatic inference rules to further reason about the contextual meaning of the user's utterance. The system maintains a declarative knowledge base of the system's beliefs, such as observed properties of objects, interpretations from the NLU component, and any logical inferences thereof.

3.1.2 Goal-based Dialogue Manager and Robot Actions

The dialogue manager component is responsible for handling the semantics of a speaker's input and forming system goals based on the speech act type of the user's input. In the case of an instruction, the intent of the speaker will be adopted as the robot's goal, which will either be handled by invoking an action satisfying the goal (if all referents are known), or using the decision network to generate a clarification question. In the case of a statement, the system will modify its declarative knowledge with any facts expressed in (or inferred from) the input. In both cases, the NLG component will be used to create a response by the robot; typically a simple acknowledgement.

Robot actions are implemented as *action scripts* that provide abstract logical formulations of actions consisting of preconditions, effects, and constituent steps (Scheutz et al., 2019). In our system, the robot has action scripts for every basic action that it is able to perform, such as moving to a location or picking up an object. Furthermore, DIARC allows for one-shot learning of novel actions through issuing sequences of lower-level instructions (Scheutz et al., 2017).

3.1.3 Reference Resolution

Our system is able to learn novel objects through a reference resolution component that interacts with the dialogue manager. When an unknown referent is encountered, the system will compute the number of possible entities that it could refer to, based on the properties that the system currently knows about the concept. If there are multiple possible referents, the dialogue manager will utilize the decision network model to generate a clarification question; any responses from the user are interpreted and used to update the system's declarative knowledge. Once a single referent is obtained, the system will identify the object with the corresponding concept and execute the instruction. Thus, the system is able to acquire knowledge about concepts through repeated application of this process.

4 Evaluation

To evaluate the integrated system, we implemented it on a PR2 robot in a virtual spacecraft environment containing unknown objects and procedures for the robot to learn. The robot performed a collaborative tool organization task in which it was instructed via typed natural language commands to place novel tools in their correct containers. In our evaluation, the robot is given sequences of commands from a subset of the Human-Robot Dialogue Learning (HuRDL) corpus (Gervits et al., 2021b) consisting of dialogues from 10 participants[4]. The human-generated questions in these dialogues are compared to the questions generated by the robot for the same commands in terms of *accuracy* (the proportion of commands that the robot is able to execute after resolving unknown referents) and *question efficiency* (the average number of questions that the agent must ask to learn each new concept).

The spacecraft environment contains 18 tools, with six main types and three instances of each type

[3]The logical representation used by DIARC is an extension of first-order predicate logic (Scheutz et al., 2019).

[4]We use only "low-level" dialogues with Commander initiative from the HuRDL corpus to match the robot task.

Table 1: Comparison of human performance to integrated system on question efficiency and accuracy.

	Human (N=10)	**Robot (N=1)**
# Questions	31	55
Question Ef.	1.72	2.29
Accuracy	0.77	1.00

(given novel sci-fi names, such as "electro capacitor") that also vary along six feature dimensions such as color, size, etc. The environment also contains 18 containers such as platforms, lockers, and crates; some of these are locked and require learning specialized procedures to open. The robot starts with a basic perceptual representation of the entities in the environment, including their observed properties (e.g., an entity is red, small, etc.), but without a name for any of them.

Our results are summarized in Table 1[5]. Overall, the robot asked more questions than the humans on average, but attains a higher accuracy, being able to resolve every entity in the task with enough questions. These results highlight a trade-off between accuracy and question efficiency relative to human performance: as our system lacks commonsense knowledge that humans are able to draw upon when learning new concepts, it generally needs to ask more questions per object, but its systematic approach to disambiguation allows it to avoid mistakes that humans would occasionally make, such as overlooking an entity in the environment.

5 Conclusion and Future Work

We presented a robotic system that combines a decision network model for question generation with a cognitive robotic architecture to allow the system to efficiently learn about new concepts in unexplored environments through dialogue. The design of our system is scalable due to the dynamic construction of the decision network, while the robotic architecture allows for broader situated interaction including symbolic reasoning and explanation generation. Our evaluation demonstrated that our system, while having slightly lower question efficiency than human participants on the same task, was adept at learning new concepts in our experimental setting. In the future, we aim to allow the robot to automatically acquire property knowledge through exploration prior to concept learning.

[5]Since the robot produces deterministic outcomes for the same command, we perform only a single trial for the robot.

References

J. Arkin, R. Paul, S. Roy, D. Park, N. Roy, and T. M. Howard. 2020. Real-time human-robot communication for manipulation tasks in partially observed environments. In *Proc. of ISER*.

Joyce Y. Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. 2018. Language to action: Towards interactive task learning with physical agents. In *Proc. of IJCAI*.

Robin Deits, Stefanie Tellex, Pratiksha Thaker, Dimitar Simeonov, Thomas Kollar, and Nicholas Roy. 2013. Clarifying commands with information-theoretic human-robot dialog. *JHRI*, 2(2).

Felix Gervits, Gordon Briggs, Antonio Roque, Genki A. Kadomatsu, Thurston Dean, Matthias Scheutz, and Matthew Marge. 2021a. Decision-theoretic question generation for situated reference resolution: An empirical study and computational model. In *Proc. of ICMI*.

Felix Gervits, Antonio Roque, Gordon Briggs, Matthias Scheutz, and Matthew Marge. 2021b. How should agents ask questions for situated learning? An annotated dialogue corpus. In *Proc. of SIGdial*.

Séverin Lemaignan, Raquel Ros, E Akin Sisbot, Rachid Alami, and Michael Beetz. 2012. Grounding the interaction: Anchoring situated discourse in everyday human-robot interaction. *International Journal of Social Robotics*, 4(2).

Ian Perera, James Allen, Choh Man Teng, and Lucian Galescu. 2018. Building and learning structures in a situated blocks world through deep language understanding. In *Proc. of SpLU*.

Matthias Scheutz, Evan Krause, Brad Oosterveld, Tyler Frasca, and Robert Platt. 2017. Spoken instruction-based one-shot object and action learning in a cognitive robotic architecture. In *Proc. of AAMAS*.

Matthias Scheutz, Thomas Williams, Evan Krause, Brley Oosterveld, Vasanth Sarathy, and Tyler Frasca. 2019. An overview of the distributed integrated affect and reflection cognitive DIARC architecture. In *Cognitive Architectures*.

Lanbo She, Shaohua Yang, Yu Cheng, Yunyi Jia, Joyce Chai, and Ning Xi. 2014. Back to the blocks world: Learning new actions through situated human-robot dialogue. In *Proc. of SIGdial*.

Danijel Skočaj, Matej Kristan, Alen Vrečko, Marko Mahnič, Miroslav Janíček, Geert-Jan M. Kruijff, Marc Hanheide, Nick Hawes, Thomas Keller, Michael Zillich, and Kai Zhou. 2011. A system for interactive learning in dialogue with a tutor. In *Proc. of IROS*.

Tom Williams, Fereshta Yazdani, Prasanth Suresh, Matthias Scheutz, and Michael Beetz. 2019. Dempster-shafer theoretic resolution of referential ambiguity. *Autonomous Robots*, 43(2).

Author Index