SIGDIAL 2010

Proceedings of the SIGDIAL 2010 Conference

The 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue

Edited by
Raquel Fernández, Yasuhiro Katagiri,
Kazunori Komatani, Oliver Lemon, Mikio Nakano

24–25 September 2010
The University of Tokyo
Tokyo, Japan

Manufacturing by
Soubundo Printing, Co. Ltd.
1-7 Toiya-cho, Fukui
Fukui 918-8231
Japan

In cooperation with:

The University of Tokyo Interfaculty Initiative in Information Studies
Association for the Advancement of Artificial Intelligence (AAAI)

We thank our sponsors:

Honda Research Institutes
Inoue Foundation for Science
Microsoft Research
The Japanese Society for Artificial Intelligence
AT&T

Order copies of this and other ACL proceedings from:

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 978-1-932432-85-5

Introduction

It is our great pleasure to present the Proceedings of the SIGDIAL 2010 Conference, the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue. This is the second meeting since the SIGDIAL meeting was elevated from a Workshop to a Conference.

We received a large number of submissions: 97 in total. The members of the Program Committee did a superb job in reviewing the submitted papers, providing helpful comments and contributing to discussions when required. We wish to thank all of them for their advice in selecting the accepted papers and for helping to maintain the high quality of the resulting program. Many submissions received strong recommendations from the Program Committee. In line with the SIGDIAL tradition, our aim has been to create a balanced program that could accommodate as many favorably rated papers as possible. Out of 70 submitted long papers, 23 were accepted as full papers for plenary presentation and 20 were accepted as short papers for poster presentations or demos. In addition, 15 out of the 27 submitted short papers and demo descriptions were accepted.

This year, the review process has included a new initiative: a mentoring program designed to assist authors of papers that contain innovative ideas to improve their quality regarding English language usage or paper organization. Overall, 7 accepted papers participated in the mentoring program, which was coordinated by Jason D. Williams. Our thanks go to the Program Committee members and others who volunteered to serve as mentors, and especially those volunteers who were called upon to mentor. Feedback from authors and mentors on the mentoring program has been very positive, and we hope that mentoring will be included as part of the review process in future SIGDIAL conferences.

We are also grateful to two keynote speakers: Professor Hiroshi Ishiguro and Professor Marilyn Walker for giving thought-provoking talks on the state-of-the-art in dialogue systems and human-robot interaction research.

For the first time, this year a local organizing committee was formed. Our thanks go to its members who worked very hard on the local arrangements such as deciding the venue, maintaining the conference web site, handling registrations, managing the conference bank account, printing proceedings, and arranging the conference lunches and dinner.

We would like to thank the ACL and Priscilla Rasmussen for handling the financial transactions. Thanks also to the SIGDIAL board, in particular Tim Paek, Amanda Stent, and Kristiina Jokinen, for their advice and support in all matters including finding industrial sponsors, budget planning, handling sponsorship, and advertising the call for papers.

Finally, we thank all the authors of the papers in this volume, and all the conference participants for making this event such a great opportunity for new research in dialogue and discourse.

Yasuhiro Katagiri and Mikio Nakano
General Co-Chairs

Raquel Fernández and Oliver Lemon
Program Co-Chairs

Kazunori Komatani
Local Chair

Conference Organization

General Co-Chairs:

Yasuhiro Katagiri, Future University - Hakodate, Japan
Mikio Nakano, Honda Research Institute Japan, Japan

Technical Program Co-Chairs:

Raquel Fernández, University of Amsterdam, Netherlands
Oliver Lemon, Heriot-Watt University, UK

Local Chair:

Kazunori Komatani, Nagoya University, Japan

Local Committee:

Kohji Dohsaka, NTT Corporation, Japan
Shinya Fujie, Waseda University, Japan
Ryuichiro Higashinaka, NTT Corporation, Japan
Masato Ishizaki, The University of Tokyo, Japan
Ikuyo Morimoto, Kwansei Gakuin University, Japan

Mentoring Program Coodinator:

Jason Williams, AT&T Labs - Research, USA

SIGDIAL Organization:

President: Tim Paek, Microsoft Research, USA
Vice-President: Amanda Stent, AT&T Labs - Research, USA
Secretary/Treasurer: Kristiina Jokinen, University of Helsinki, Finland

Program Committee:

Masahiro Araki, Kyoto Institute of Technology, Japan
Gregory Aist, Arizona State University, USA (*)
Jan Alexandersson, DFKI GmbH, Germany
Srinivas Bangalore, AT&T Labs - Research, USA (**)
Ellen Bard, University of Edinburgh, UK
Dan Bohus, Microsoft Research, USA
Johan Bos, Universita di Roma ”La Sapienza”, Italy
Johan Boye, Linkoping University, Sweden
Donna Byron, Northeastern University, USA (*)
Rolf Carlson, Royal Institute of Technology (KTH), Sweden
Robin Cooper, Göteborg University, Sweden (**)
Mark Core, University of Southern California, USA
David DeVault, University of Southern California, USA
Myroslava Dzikovska, University of Edinburgh, UK
Markus Egg, Rijksuniversiteit Groningen, Netherlands (*)
Mary Ellen Foster, Heriot-Watt University, UK (*)

Matthew Frampton, Stanford University, USA (*)
Kallirroi Georgila, University of Southern California, USA (**)
Jonathan Ginzburg, King's College London, UK (**)
Genevieve Gorrell, Sheffield University, UK (*)
Alexander Gruenstein, Google, USA
Helen Hastie, Heriot-Watt University, UK (**)
Pat Healey, Queen Mary University of London, UK
Beth Ann Hockey, University of California at Santa Cruz, USA
Kristiina Jokinen, University of Helsinki, Finland (*)
Tatsuya Kawahara, Kyoto University, Japan
Simon Keizer, University of Cambridge, UK
John Kelleher, Dublin Institute of Technology, Ireland
Alexander Koller, University of Saarbrucken, Germany
Alistair Knott, Otago University, New Zealand
Ivana Kruijff-Korbayová, DFKI, Germany (*)
Gary Geunbae Lee, Pohang University of Science and Technology, Korea (*)
Fabrice Lefevre, University of Cambridge, UK (*)
James Lester, North Carolina State University, USA
Diane Litman, University of Pittsburgh, USA
Ramón López-Cózar, University of Granada, Spain (*)
François Mairesse, University of Cambridge, UK
Michael McTear, University of Ulster, UK (**)
Wolfgang Minker, University of Ulm, Germany
Sebastian Möller, Deutsche Telekom Labs and Technical Univ. Berlin, Germany
Yukiko Nakano, Seikei University, Japan
Tim Paek, Microsoft Research, USA (**)
Paul Piwek, Open University, UK
Massimo Poesio, Univ. of Essex/Univ. of Trento, UK/Italy
Rashmi Prasad, University of Pennsylvania, USA (*)
Matt Purver, Queen Mary University of London, UK (*)
Verena Rieser, University of Edinburgh, UK
Laurent Romary, INRIA, France
Alex Rudnicky, Carnegie Mellon University, USA (*)
David Schlangen, University of Potsdam, Germany (*)
Candy Sidner, BAE Systems AIT, USA
Gabriel Skantze, KTH, Sweden
Ronnie Smith, East Carolina University, USA (**)
Amanda Stent, AT&T Labs - Research, USA (*)
Matthew Stone, Rutgers University, USA
Svetlana Stoyanchev, Open University, UK
Matthew Stuttle, Toshiba Research, UK (*)
Joel Tetreault, Educational Testing Service, USA
David Traum, USC/ICT, USA (*)
Jason Williams, AT&T Labs - Research, USA

(*) Mentor volunteers
(**) Mentor volunteers who mentored a paper this year

Invited Speakers:

Hiroshi Ishiguro, Osaka University, Japan
Marilyn Walker, University of California at Santa Cruz, USA

Table of Contents

Conference Program

Friday, September 24, 2010

9:00-9:20 Introduction

9:20-10:00 Papers 1-2:

Towards Incremental Speech Generation in Dialogue Systems
Gabriel Skantze and Anna Hjalmarsson

Comparing Local and Sequential Models for Statistical Incremental Natural Language Understanding
Silvan Heintze, Timo Baumann and David Schlangen

10:00-11:00 Invited Talk 1:

Dynamic Adaptation in Dialog Systems
Marilyn Walker

11:00-11:20 Coffee break

11:20-12:20 Papers 3-5:

Modeling User Satisfaction Transitions in Dialogues from Overall Ratings
Ryuichiro Higashinaka, Yasuhiro Minami, Kohji Dohsaka and Toyomi Meguro

Evaluation Metrics For End-to-End Coreference Resolution Systems
Jie Cai and Michael Strube

Probabilistic Ontology Trees for Belief Tracking in Dialog Systems
Neville Mehta, Rakesh Gupta, Antoine Raux, Deepak Ramachandran and Stefan Krawczyk

12:20-13:30 Lunch & SIGDIAL Business Meeting

13:30-15:00 Short Paper Poster Session 1

How was your day? An architecture for multimodal ECA systems
Raúl Santos de la Cámara, Markku Turunen, Jaakko Hakulinen and Debora Field

Middleware for Incremental Processing in Conversational Agents
David Schlangen, Timo Baumann, Hendrik Buschmeier, Okko Buß, Stefan Kopp, Gabriel Skantze and Ramin Yaghoubzadeh

Friday, September 24, 2010 (continued)

Towards Semi-Supervised Classification of Discourse Relations using Feature Correlations
Hugo Hernault, Danushka Bollegala and Mitsuru Ishizuka

Using entity features to classify implicit discourse relations
Annie Louis, Aravind Joshi, Rashmi Prasad and Ani Nenkova

Same and Elaboration Relations in the Discourse Graphbank
Irina Borisova and Gisela Redeker

Negotiating causal implicatures
Luciana Benotti and Patrick Blackburn

Presupposition Accommodation as Exception Handling
Philippe de Groote and Ekaterina Lebedeva

Exploring the Effectiveness of Lexical Ontologies for Modeling Temporal Relations with Markov Logic
Eun Y. Ha, Alok Baikadi, Carlyle Licata, Bradford Mott and James Lester

Reference reversibility with Reference Domain Theory
Alexandre Denis

Utilizing Review Summarization in a Spoken Recommendation System
Jingjing Liu, Stephanie Seneff and Victor Zue

Dialogue Management Based on Entities and Constraints
Yushi Xu and Stephanie Seneff

Towards Improving the Naturalness of Social Conversations with Dialogue Systems
Matthew Marge, João Miranda, Alan Black and Alexander Rudnicky

Route Communication in Dialogue: a Matter of Principles
Theodora Koulouri and Stanislao Lauria

The Impact of Dimensionality on Natural Language Route Directions in Unconstrained Dialogue
Vivien Mast, Jan Smeddinck, Anna Strotseva and Thora Tenbrink

Learning Dialogue Strategies from Older and Younger Simulated Users
Kallirroi Georgila, Maria Wolters and Johanna Moore

Friday, September 24, 2010 (continued)

15:00-16:00 Papers 6-8:

Sparse Approximate Dynamic Programming for Dialog Management
Senthilkumar Chandramohan, Matthieu Geist and Olivier Pietquin

Parameter estimation for agenda-based user simulation
Simon Keizer, Milica Gasic, Filip Jurcicek, Francois Mairesse, Blaise Thomson, Kai Yu and Steve Young

Adaptive Referring Expression Generation in Spoken Dialogue Systems: Evaluation with Real Users
Srinivasan Janarthanam and Oliver Lemon

16:00-16:20 Coffee break

16:20-17:20 Papers 9-11:

A unified account of the semantics of discourse particles
Markus Egg

The Effects of Discourse Connectives Prediction on Implicit Discourse Relation Recognition
Zhi Min Zhou, Man Lan, Zheng Yu Niu, Yu Xu and Jian Su

Discourse indicators for content selection in summarization
Annie Louis, Aravind Joshi and Ani Nenkova

19:00-22:00 Conference Dinner

Saturday, September 25, 2010

9:20-10:00 Papers 12-13

Comparing Spoken Language Route Instructions for Robots across Environment Representations
Matthew Marge and Alexander Rudnicky

The Dynamics of Action Corrections in Situated Interaction
Antoine Raux and Mikio Nakano

10:00-11:00 Invited Talk 2:

Understanding Humans by Building Androids
Hiroshi Ishiguro

11:00-11:20 Coffee break

11:20-12:20 Papers 14-16

Non-humanlike Spoken Dialogue: A Design Perspective
Kotaro Funakoshi, Mikio Nakano, Kazuki Kobayashi, Takanori Komatsu and Seiji Yamada

Enhanced Monitoring Tools and Online Dialogue Optimisation Merged into a New Spoken Dialogue System Design Experience
Ghislain Putois, Romain Laroche and Philippe Bretier

Don't tell anyone! Two Experiments on Gossip Conversations
Jenny Brusk, Ron Artstein and David Traum

12:20-13:00 Lunch & Spoken Dialogue Challenge Report

13:00-14:40 Short Paper Poster Session 2 & Demo Session

Gaussian Processes for Fast Policy Optimisation of POMDP-based Dialogue Managers
Milica Gasic, Filip Jurcicek, Simon Keizer, Francois Mairesse, Blaise Thomson, Kai Yu and Steve Young

Coherent Back-Channel Feedback Tagging of In-Car Spoken Dialogue Corpus
Yuki Kamiya, Tomohiro Ohno and Shigeki Matsubara

Representing Uncertainty about Complex User Goals in Statistical Dialogue Systems
Paul A. Crook and Oliver Lemon

Investigating Clarification Strategies in a Hybrid POMDP Dialog Manager
Sebastian Varges, Silvia Quarteroni, Giuseppe Riccardi and Alexei Ivanov

Saturday, September 25, 2010 (continued)

Cooperative User Models in Statistical Dialog Simulators
Meritxell González, Silvia Quarteroni, Giuseppe Riccardi and Sebastian Varges

Modeling Spoken Decision Making Dialogue and Optimization of its Dialogue Strategy
Teruhisa Misu, Komei Sugiura, Kiyonori Ohtake, Chiori Hori, Hideki Kashioka, Hisashi Kawai and Satoshi Nakamura

The vocal intensity of turn-initial cue phrases in dialogue
Anna Hjalmarsson

Pamini: A framework for assembling mixed-initiative human-robot interaction from generic interaction patterns
Julia Peltason and Britta Wrede

Collaborating on Utterances with a Spoken Dialogue System Using an ISU-based Approach to Incremental Dialogue Management
Okko Buß, Timo Baumann and David Schlangen

Cross-Domain Speech Disfluency Detection
Kallirroi Georgila, Ning Wang and Jonathan Gratch

Validation of a Dialog System for Language Learners
Alicia Sagae, W. Lewis Johnson and Stephen Bodnar

I've said it before, and I'll say it again: An empirical investigation of the upper bound of the selection approach to dialogue
Sudeep Gandhe and David Traum

Autism and Interactional Aspects of Dialogue
Peter Heeman, Rebecca Lunsford, Ethan Selfridge, Lois Black and Jan van Santen

Detection of time-pressure induced stress in speech via acoustic indicators
Matthew Frampton, Sandeep Sripada, Ricardo Augusto Hoffmann Bion and Stanley Peters

How to Drink from a Fire Hose: One Person Can Annoscribe One Million Utterances in One Month
David Suendermann, Jackson Liscombe and Roberto Pieraccini

Demos:

Advances in the Witchcraft Workbench Project
Alexander Schmitt, Wolfgang Minker and Nada Sharaf

MPOWERS: a Multi Points Of VieW Evaluation Refine Studio
Marianne Laurent and Philippe Bretier

Saturday, September 25, 2010 (continued)

Statistical Dialog Management Methodologies for Real Applications
David Griol, Zoraida Callejas and Ramón López-Cózar

YouBot: A Simple Framework for Building Virtual Networking Agents
Seiji Takegata and Kumiko Tanaka-Ishii

How was your day? An Affective Companion ECA Prototype
Marc Cavazza, Raúl Santos de la Cámara, Markku Turunen, José Relaño Gil, Jaakko Hakulinen, Nigel Crook and Debora Field

14:40-16:00 Papers 17-20

F^2 - New Technique for Recognition of User Emotional States in Spoken Dialogue Systems
Ramón López-Cózar, Jan Silovsky and David Griol

Online Error Detection of Barge-In Utterances by Using Individual Users' Utterance Histories in Spoken Dialogue System
Kazunori Komatani and Hiroshi G. Okuno

Dialogue Act Modeling in a Complex Task-Oriented Domain
Kristy Boyer, Eun Y. Ha, Robert Phillips, Michael Wallis, Mladen Vouk and James Lester

Hand Gestures in Disambiguating Types of You Expressions in Multiparty Meetings
Tyler Baldwin, Joyce Chai and Katrin Kirchhoff

16:00-16:20 Coffee break

16:20-17:20 Papers 21-23

User-adaptive Coordination of Agent Communicative Behavior in Spoken Dialogue
Kohji Dohsaka, Atsushi Kanemoto, Ryuichiro Higashinaka, Yasuhiro Minami and Eisaku Maeda

Towards an Empirically Motivated Typology of Follow-Up Questions: The Role of Dialogue Context
Manuel Kirschner and Raffaella Bernardi

Assessing the effectiveness of conversational features for dialogue segmentation in medical team meetings and in the AMI corpus
Saturnino Luz and Jing Su

17:20-17:50 Best Paper Awards & Closing

Towards Incremental Speech Generation in Dialogue Systems

Gabriel Skantze
Dept. of Speech Music and Hearing
KTH, Stockholm, Sweden
gabriel@speech.kth.se

Anna Hjalmarsson
Dept. of Speech Music and Hearing
KTH, Stockholm, Sweden
annah@speech.kth.se

Abstract

We present a first step towards a model of speech generation for incremental dialogue systems. The model allows a dialogue system to incrementally interpret spoken input, while simultaneously planning, realising and self-monitoring the system response. The model has been implemented in a general dialogue system framework. Using this framework, we have implemented a specific application and tested it in a Wizard-of-Oz setting, comparing it with a non-incremental version of the same system. The results show that the incremental version, while producing longer utterances, has a shorter response time and is perceived as more efficient by the users.

1 Introduction

Speakers in dialogue produce speech in a piecemeal fashion and on-line as the dialogue progresses. When starting to speak, dialogue participants typically do not have a complete plan of how to say something or even what to say. Yet, they manage to rapidly integrate information from different sources in parallel and simultaneously plan and realize new dialogue contributions. Moreover, interlocutors continuously self-monitor the actual production processes in order to facilitate self-corrections (Levelt, 1989). Contrary to this, most spoken dialogue systems use a silence threshold to determine when the user has stopped speaking. The user utterance is then processed by one module at a time, after which a complete system utterance is produced and realised by a speech synthesizer.

This paper has two purposes. First, to present an initial step towards a model of speech generation that allows a dialogue system to incrementally interpret spoken input, while simultaneously planning, realising and self-monitoring the system response. The model has been implemented in a general dialogue system framework. This is described in Section 2 and 3. The second purpose is to evaluate the usefulness of incremental speech generation in a Wizard-of-Oz setting, using the proposed model. This is described in Section 4.

1.1 Motivation

A non-incremental dialogue system waits until the user has stopped speaking (using a silence threshold to determine this) before starting to process the utterance and then produce a system response. If processing takes time, for example because an external resource is being accessed, this may result in a confusing response delay. An incremental system may instead continuously build a tentative plan of what to say as the user is speaking. When it detects that the user's utterance has ended, it may start to asynchronously realise this plan while processing continues, with the possibility to revise the plan if needed.

There are many potential reasons for why dialogue systems may need additional time for processing. For example, it has been assumed that ASR processing has to be done in real-time, in order to avoid long and confusing response delays. Yet, if we allow the system to start speaking before input is complete, we can allow more accurate (and time-consuming) ASR processing (for example by broadening the beam). In this paper, we will explore incremental speech generation in a Wizard-of-oz setting. A common problem in such settings is the time it takes for the Wizard to interpret the user's utterance and/or decide on the next system action, resulting in unacceptable response delays (Fraser & Gilbert, 1991). Thus, it would be useful if the system could start to speak as soon as the user has finished speaking, based on the Wizard's actions so far.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 1–8,
The University of Tokyo, September 24-25, 2010.

1.2 Related work

Incremental speech generation has been studied from different perspectives. From a psycholinguistic perspective, Levelt (1989) and others have studied how speakers incrementally produce utterances while *self-monitoring* the output, both overtly (listening to oneself speaking) and covertly (mentally monitoring what is about to be said). As deviations from the desired output is detected, the speaker may initiate *self-repairs*. If the item to be repaired has already been spoken, an *overt* repair is needed (for example by using an editing term, such as "sorry"). If not, the utterance plan may be altered to accommodate the repair, a so-called *covert* repair. Central to the concept of incremental speech generation is that the realization of overt speech can be initiated before the speaker has a complete plan of what to say. An option for a speaker who does not know what to say (but wants to claim the floor) is to use hesitation phenomena such as *filled pauses* ("eh") or *cue phrases* such as "let's see".

A dialogue system may not need to self-monitor its output for the same reasons as humans do. For example, there is no risk of articulatory errors (with current speech synthesis technology). However, a dialogue system may utilize the same mechanisms of self-repair and hesitation phenomena to simultaneously plan and realise the spoken output, as there is always a risk for revision in the input to an incremental module (as described in Section 2.1).

There is also another aspect of self-monitoring that is important for dialogue systems. In a system with modules operating asynchronously, the dialogue manager cannot know whether the intended output is actually realized, as the user may interrupt the system. Also, the timing of the synthesized speech is important, as the user may give feedback in the middle of a system utterance. Thus, an incremental, asynchronous system somehow needs to self-monitor its own output.

From a syntactic perspective, Kempen & Hoenkamp (1987) and Kilger & Finkler (1995) have studied how to syntactically formulate sentences incrementally under time constraints. Dohsaka & Shimazu (1997) describes a system architecture for incremental speech generation. However, there is no account for revision of the input (as discussed in Section 2.1) and there is no evaluation with users. Skantze & Schlangen (2009) describe an incremental system that partly supports incremental output and that is evaluated with users, but the domain is limited to number dictation.

In this study, the focus is not on syntactic construction of utterances, but on how to build practical incremental dialogue systems within limited domains that can handle revisions and produce convincing, flexible and varied speech output in on-line interaction with users.

2 The Jindigo framework

The proposed model has been implemented in Jindigo – a Java-based open source framework for implementing and experimenting with incremental dialogue systems (www.jindigo.net). We will here briefly describe this framework and the model of incremental dialogue processing that it is based on.

2.1 Incremental units

Schlangen & Skantze (2009) describes a general, abstract model of incremental dialogue processing, which Jindigo is based on. In this model, a system consists of a network of processing modules. Each module has a left buffer, a processor, and a right buffer, where the normal mode of processing is to receive input from the left buffer, process it, and provide output in the right buffer, from where it is forwarded to the next module's left buffer. An example is shown in Figure 1. Modules exchange incremental units (IUs), which are the smallest 'chunks' of information that can trigger connected modules into action (such as words, phrases, communicative acts, etc). IUs are typically part of larger units: individual words are parts of an utterance; concepts are part of the representation of an utterance meaning. This relation of being part of the same larger unit is recorded through *same-level links*. In the example below, IU_2 has a same-level link to IU_1 of type PREDECESSOR, meaning that they are linearly ordered. The information that was used in creating a given IU is linked to it via *grounded-in* links. In the example, IU_3 is grounded in IU_1 and IU_2, while IU_4 is grounded in IU_3.

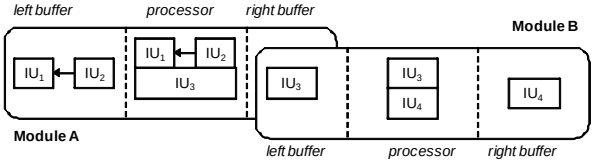


Figure 1: Two connected modules.

String	Right buffer	Update message
t_1: one	w1 — one — w2	[w1, w2]
t_2: one five	w1 — one — w2 — five — w3	[w1, w3]
t_3: one	w1 — one — w2 — five — w3	[w1, w2]
t_4: one four five	w1 — one — w2 — five — w3; four — w4 — five — w5	[w1, w5]
t_5: [commit]	w1 — one — w2 — five — w3; four — w4 — five — w5	[w5,w5]

Table 1: The right buffer of an ASR module, and update messages at different time-steps.

A challenge for incremental systems is to handle *revisions*. For example, as the first part of the word "forty" is recognised, the best hypothesis might be "four". As the speech recogniser receives more input, it might need to revise its previous output, which might cause a chain of revisions in all subsequent modules. To cope with this, modules have to be able to react to three basic situations: that IUs are *added* to a buffer, which triggers processing; that IUs that were erroneously hypothesized by an earlier module are *revoked*, which may trigger a revision of a module's own output; and that modules signal that they *commit* to an IU, that is, won't revoke it anymore.

Jindigo implements an efficient model for communicating these updates. In this model, IUs are associated with edges in a graph, as shown in Table 1. The graph may be incrementally amended without actually removing edges or vertices, even if revision occurs. At each time-step, a new update message is sent to the consuming module. The update message contains a pair of pointers [*C*, *A*]: (*C*) the vertex from which the currently committed hypothesis can be constructed, and (*A*) the vertex from which the currently best tentative hypothesis can be constructed. In Jindigo, all modules run as threads within a single Java process, and therefore have access to the same memory space.

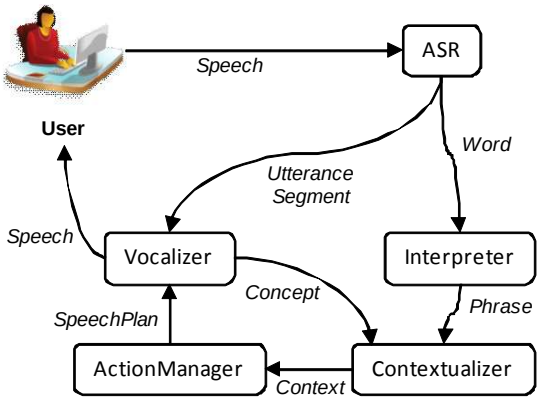


Figure 2: A typical Jindigo system architecture.

2.2 A typical architecture

A typical Jindigo system architecture is shown in Figure 2. The word buffer from the Recognizer module is parsed by the Interpreter module which tries to find an optimal sequence of top phrases and their semantic representations. These phrases are then interpreted in light of the current dialogue context by the Contextualizer module and are packaged as Communicative Acts (CAs). As can be seen in Figure 2, the Contextualizer also self-monitors Concepts from the system as they are spoken by the Vocalizer, which makes it possible to contextually interpret user responses to system utterances. This also makes it possible for the system to know whether an intended utterance actually was produced, or if it was interrupted. The current context is sent to the Action Manager, which generates a SpeechPlan that is sent to the Vocalizer. This is described in detail in the next section.

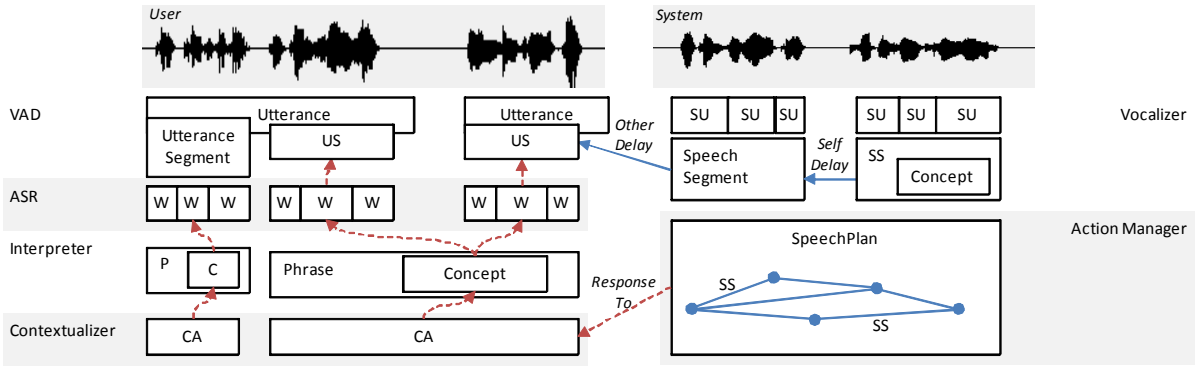


Figure 3: Incremental Units at different levels of processing. Some grounded-in relations are shown with dotted lines. W=Word, SS=SpeechSegment, SU=SpeechUnit, CA=Communicative Act.

3 Incremental speech generation

3.1 Incremental units of speech

In order for user and system utterances to be interpreted and produced incrementally, they need to be decomposed into smaller units of processing (IUs). This decomposition is shown in Figure 3. Using a standard voice activity detector (VAD) in the ASR, the user's speech is chunked into **`Utterance`**-units. The `Utterance` boundaries determine when the ASR hypothesis is committed. However, for the system to be able to respond quickly, the end silence threshold of these Utterances are typically too long. Therefore smaller units of the type **`UtteranceSegment`** (US) are detected, using a much shorter silence threshold of about 50ms. Such short silence thresholds allow the system to give very fast responses (such as backchannels). Information about US boundaries is sent directly from the ASR to the Vocalizer. As Figure 3 illustrates, the grounded-in links can be followed to derive the timing of IUs at different levels of processing.

The system output is also modelled using IUs at different processing levels. The widest-spanning IU on the output side is the **`SpeechPlan`**. The rendering of a `SpeechPlan` will result in a sequence of **`SpeechSegment`**'s, where each `SpeechSegment` represents a continuous audio rendering of speech, either as a synthesised string or a pre-recorded audio file. For example, the plan may be to say "okay, a red doll, here is a nice doll", consisting of three segments. Now, there are two requirements that we need to meet. First, the output should be *varied*: the system should not give exactly the same response every time to the same request. But, as we will see, the output in an incremental system must also be *flexible*, as speech plans are incrementally produced and amended. In order to relieve the Action Manager of the burden of varying the output and making time-critical adjustments, we model the `SpeechPlan` as a directed graph, where each edge is associated with a `SpeechSegment`, as shown in Figure 4. Thus, the Action Manager may asynchronously plan (a set of possible) responses, while the Vocalizer selects the rendering path in the graph and takes care of time-critical synchronization. To control the rendering, each `SpeechSegment` has the properties `optional`, `committing`, `selfDelay` and `otherDelay`, as described in the next section. It must also be possible for an incremental system to interrupt and make self-repairs in the middle of a `SpeechSegment`. Therefore, each `SpeechSegment` may also be decomposed into an array of **`SpeechUnit`**'s, where each `SpeechUnit` contains pointers to the audio rendering in the `SpeechSegment`.

3.2 Producing and consuming SpeechPlans

The `SpeechPlan` does not need to be complete before the system starts to speak. An example of this is shown in Figure 4. As more words are recognised by the ASR, the Action Manager may add more `SpeechSegment`'s to the graph. Thus, the system may start to say "it costs" before it knows which object is being talked about.

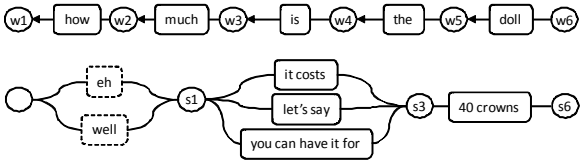


Figure 4: The right buffer of an ASR (top) and the `SpeechPlan` that is incrementally produced (bottom). Vertex s1 is associated with w1, s3 with w3, etc. Optional, non-committing `SpeechSegment`'s are marked with dashed outline.

The `SpeechPlan` has a pointer called `finalVertex`. When the Vocalizer reaches the `finalVertex`, the `SpeechPlan` is completely realised. If `finalVertex` is not set, it means that the `SpeechPlan` is not yet completely constructed. The `SpeechSegment` property `optional` tells whether the segment needs to be realised or if it could be skipped if the `finalVertex` is in sight. This makes it possible to insert floor-keeping `SpeechSegment`'s (such as "eh") in the graph, which are only realised if needed. The Vocalizer also keeps track of which `SpeechSegment`'s it has realised before, so that it can look ahead in the graph and realise a more varied output. Each `SpeechSegment` may carry a semantic representation of the segment (a `Concept`). This is sent by the Vocalizer to the Contextualizer as soon as the segment has been realised.

The `SpeechSegment` properties `selfDelay` and `otherDelay` regulate the timing of the output (as illustrated in Figure 3). They specify the number of milliseconds that should pass before the Vocalizer starts to play the segment, depending on the previous speaker. By setting the `otherDelay` of a segment, the Action Manager may delay the response depending on how certain it is that it is appropriate to speak, for example by considering pitch and semantic completeness. (See Raux & Eskenazi (2008) for a study

on how such dynamic delays can be derived using machine learning.)

If the user starts to speak (i.e., a new `UtteranceSegment` is initiated) as the system is speaking, the Vocalizer pauses (at a `SpeechUnit` boundary) and waits until it has received a new response from the Action Manager. The Action Manager may then choose to generate a new response or simply ignore the last input, in which case the Vocalizer continues from the point of interruption. This may happen if, for example, the `UtteranceSegment` was identified as a back-channel, cough, or similar.

3.3 Self-repairs

As Figure 3 shows, a `SpeechPlan` may be grounded in a user CA (i.e., it is a response to this CA). If this CA is revoked, or if the `SpeechPlan` is revised, the Vocalizer may initialize a self-repair. The Vocalizer keeps a list of the `SpeechSegment`'s it has realised so far. If the `SpeechPlan` is revised when it has been partly realised, the Vocalizer compares the history with the new graph and chooses one of the different repair strategies shown in Table 2. In the best case, it may smoothly switch to the new plan without the user noticing it (covert repair). In case of a unit repair, the Vocalizer searches for a zero-crossing point in the audio segment, close to the boundary pointed out by the `SpeechUnit`.

covert segment repair	you are right → it is blue you are right → they are blue
overt segment repair	you are right → it is blue sorry you are wrong → it is red
covert unit repair	you are right → it is blue you are wrong → it is red
overt unit repair	you are right → it is blue sorry you are wrong → it is red

Table 2: Different types of self-repairs. The shaded boxes show which `SpeechUnit`'s have been realised, or are about to be realised, at the point of revision.

The `SpeechSegment` property `committing` tells whether it needs to be repaired if the `SpeechPlan` is revised. For example, a filled pause such as "eh" is not committing (there is no need to insert an editing term after it), while a request or an assertion usually is. If (parts of) a committing segment has already been realised and it cannot be part of the new plan, an overt repaired is made with the help of an editing term (e.g., "sorry"). When comparing the history with the new graph, the Vocalizer searches the graph and tries to find a path so that it may avoid making an overt repair. For example if the graph in Figure 4 is replaced with a corresponding one that ends with "60 crowns", and it has so far partly realised "it costs", it may choose the corresponding path in the new `SpeechPlan`, making a covert repair.

4 A Wizard-of-Oz experiment

A Wizard-of-Oz experiment was conducted to test the usefulness of the model outlined above. All modules in the system were fully functional, except for the ASR, since not enough data had been collected to build language models. Thus, instead of using ASR, the users' speech was transcribed by a Wizard. As discussed in section 1.1, a common problem is the time it takes for the Wizard to transcribe incoming utterances, and thus for the system to respond. Therefore, this is an interesting test-case for our model. In order to let the system respond as soon as the user finished speaking, even if the Wizard hasn't completed the transcription yet, a VAD is used. The setting is shown in Figure 5 (compare with Figure 2). The Wizard may start to type as soon as the user starts to speak and may alter whatever he has typed until the return key is pressed and the hypothesis is committed. The word buffer is updated in exactly the same manner as if it had been the output of an ASR.

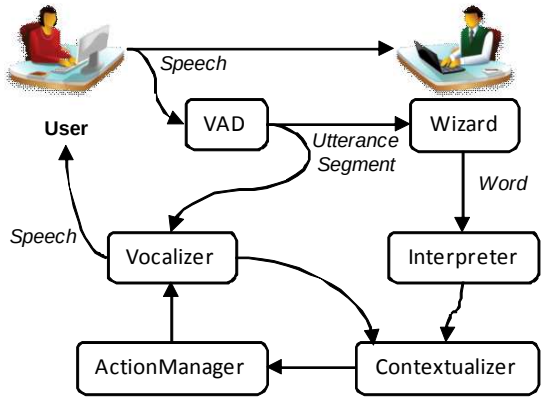


Figure 5: The system architecture used in the Wizard-of-Oz experiment.

For comparison, we also configured a non-incremental version of the same system, where nothing was sent from the Wizard until he com-

mitted by pressing the return key. Since we did not have mature models for the Interpreter either, the Wizard was allowed to adapt the transcription of the utterances to match the models, while preserving the semantic content.

4.1 The DEAL domain

The system that was used in the experiment was a spoken dialogue system for second language learners of Swedish under development at KTH, called DEAL (Hjalmarsson et al., 2007). The scene of DEAL is set at a flea market where a talking agent is the owner of a shop selling used goods. The student is given a mission to buy items at the flea market getting the best possible price from the shop-keeper. The shop-keeper can talk about the properties of goods for sale and negotiate about the price. The price can be reduced if the user points out a flaw of an object, argues that something is too expensive, or offers lower bids. However, if the user is too persistent haggling, the agent gets frustrated and closes the shop. Then the user has failed to complete the task.

For the experiment, DEAL was re-implemented using the Jindigo framework. Figure 6 shows the GUI that was shown to the user.

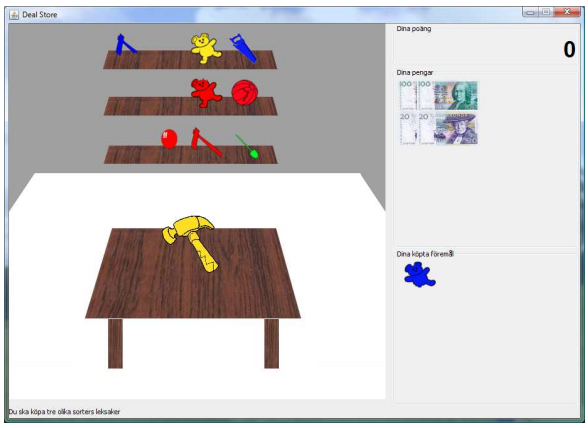


Figure 6: The user interface in DEAL. The object on the table is the one currently in focus. Example objects are shown on the shelf. Current game score, money and bought objects are shown on the right.

4.2 Speech segments in DEAL

In a previous data collection of human-human interaction in the DEAL domain (Hjalmarsson, 2008) it was noted that about 40% of the speaker turns were initiated with standardized lexical expressions (cue phrases) or filled pauses. Such speech segments commit very little semantically to the rest of the utterance and are therefore very useful as initiations of utterances, since such speech segments can be produced immediately after the user has stopped speaking, allowing the Wizard to exploit the additional time to transcribe the rest of the utterance.

The DEAL corpus was used to create utterance initial speech segments for the experiment. The motivation to use speech segments derived from human recordings was to make the system sound convincing in terms of both lexical choice and intonation. In particular, we wanted a repertoire of different types of filled pauses and feedback expression such as “eh” and “mm” in order to avoid a system that sounds monotone and repetitive. First, a number of feedback expression such as “ja”, “a”, “mm” (Eng: “yes”), filled pauses such as “eh”, “ehm” and expressions used to initiate different domain specific speech acts (for example “it costs” and “let me see”) were extracted. The segments were re-synthesized using Expros, a tool for experimentation with prosody in diphone voices (Gustafson & Edlund, 2008). Based on manual transcriptions and sound files, Expros automatically extracts pitch, duration and intensity from the human voice and creates a synthetic version using these parameters. In the speech plan, these canned segments were mixed with generated text segments (for example references to objects, prices, etc) that were synthesized and generated on-line with the same diphone voice.

An example interaction with the incremental version of the system is shown in Table 3. S.11 exemplifies a self-correction, where the system prepares to present another bid, but then realizes that the user’s bid is too low to even consider. A video (with subtitles) showing an interaction with one of the users can be seen at http://www.youtube.com/watch?v=cQQmgItIMvs.

S.1	[welcome] [how may I help you]
U.2	*I want to buy a doll*
S.3	[eh] [here is] [a doll]
U.4	*how much is it?*
S.5	[eh] [it costs] [120 crowns]
U.6	*that is too expensive* *how much is the teddy bear?*
S.7	[well] [you can have it for] [let’s see] [40 crowns]
U.8	*I can give you 30 crowns*
S.9	[you could have it for] [37 crowns]
U.10	*I can give you 10 crowns*
S.11	[let’s say] [or, I mean] [that is way too little]

Table 3: An example DEAL dialogue (translated from Swedish). Speech segments are marked in brackets.

4.3 Experimental setup

In order to compare the incremental and non-incremental versions of the system, we conducted an experiment with 10 participants, 4 male and 6 female. The participants were given a mission: to buy three items (with certain characteristics) in DEAL at the best possible price from the shop-keeper. The participants were further instructed to evaluate two different versions of the system, System A and System B. However, they were not informed how the versions differed. The participants were lead to believe that they were interacting with a fully working dialogue system and were not aware of the Wizard-of-Oz set up. Each participant interacted with the system four times, first two times with each version of the system, after which a questionnaire was completed. Then they interacted with the two versions again, after which they filled out a second questionnaire with the same questions. The order of the versions was balanced between subjects.

The mid-experiment questionnaire was used to collect the participants' first opinions of the two versions and to make them aware of what type of characteristics they should consider when interacting with the system the second time. When filling out the second questionnaire, the participants were asked to base their ratings on their overall experience with the two system versions. Thus, the analysis of the results is based on the second questionnaire. In the questionnaires, they were requested to rate which one of the two versions was most prominent according to 8 different dimensions: which version they *preferred*; which was more *human-like*, *polite*, *efficient*, and *intelligent*; which gave a *faster response* and better *feedback*; and with which version it was easier to know *when to speak*. All ratings were done on a continuous horizontal line with System A on the left end and System B on the right end. The centre of the line was labelled with "no difference".

The participants were recorded during their interaction with the system, and all messages in the system were logged.

4.4 Results

Figure 7 shows the difference in response time between the two versions. As expected, the incremental version started to speak more quickly (M=0.58s, SD=1.20) than the non-incremental version (M=2.84s, SD=1.17), while producing longer utterances. It was harder to anticipate whether it would take more or less time for the incremental version to finish utterances. Both versions received the final input at the same time. On the one hand, the incremental version initiates utterances with speech segments that contain little or no semantic information. Thus, if the system is in the middle of such a segment when receiving the complete input from the Wizard, the system may need to complete this segment before producing the rest of the utterance. Moreover, if an utterance is initiated and the Wizard alters the input, the incremental version needs to make a repair which takes additional time. On the other hand, it may also start to produce speech segments that are semantically relevant, based on the incremental input, which allows it to finish the utterance more quickly. As the figure shows, it turns out that the average response completion time for the incremental version (M=5.02s, SD=1.54) is about 600ms faster than the average for non-incremental version (M=5.66s, SD=1.50), ($t(704)=5.56$, $p<0.001$).

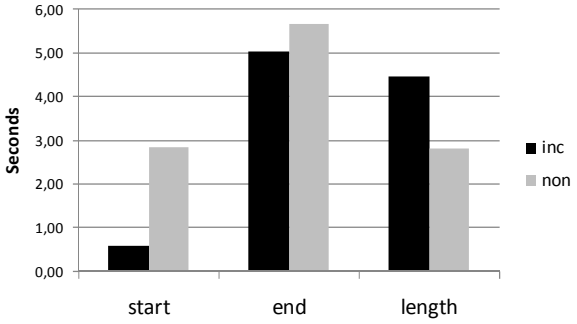


Figure 7: The first two column pairs show the average time from the end of the user's utterance to the *start* of the system's response, and from the end of the user's utterance to the *end* of the system's response. The third column pair shows the average total system utterance *length* (end minus start).

In general, subjects reported that the system worked very well. After the first interaction with the two versions, the participants found it hard to point out the difference, as they were focused on solving the task. The marks on the horizontal continuous lines on the questionnaire were measured with a ruler based on their distance from the midpoint (labelled with "no difference") and normalized to a scale from -1 to 1, each extreme representing one system version. A Wilcoxon Signed Ranks Test was carried out, using these rankings as differences. The results are shown in Table 4. As the table shows, the two versions differed significantly in three dimensions, all in favour of the incremental version.

Hence, the incremental version was rated as more polite, more efficient, and better at indicating when to speak.

	diff	z-value	p-value
preferred	0.23	-1.24	0.214
human-like	0.15	-0.76	0.445
polite	0.40	-2.19	0.028*
efficient	0.29	-2.08	0.038*
intelligent	0.11	-0.70	0.484
faster response	0.26	-1.66	0.097
feedback	0.08	-0.84	0.400
when to speak	0.35	-2.38	0.017*

Table 4: The results from the second questionnaire. All differences are positive, meaning that they are in favour of the incremental version.

A well known phenomena in dialogue is that of *entrainment* (or *adaptation* or *alignment*), that is, speakers (in both human-human and human-computer dialogue) tend to adapt the conversational behaviour to their interlocutor (e.g., Bell, 2003). In order to examine whether the different versions affected the user's behaviour, we analyzed both the user utterance length and user response time, but found no significant differences between the interactions with the two versions.

5 Conclusions & Future work

This paper has presented a first step towards incremental speech generation in dialogue systems. The results are promising: when there are delays in the processing of the dialogue, it is possible to incrementally produce utterances that make the interaction more efficient and pleasant for the user.

As this is a first step, there are several ways to improve the model. First, the edges in the `SpeechPlan` could have probabilities, to guide the path planning. Second, when the user has finished speaking, it should (in some cases) be possible to anticipate how long it will take until the processing is completed and thereby choose a more optimal path (by taking the length of the `SpeechSegment`'s into consideration). Third, a lot of work could be done on the dynamic generation of `SpeechSegment`'s, considering syntactic and pragmatic constraints, although this would require a speech synthesizer that was better at convincingly produce conversational speech.

The experiment also shows that it is possible to achieve fast turn-taking and convincing responses in a Wizard-of-Oz setting. We think that this opens up new possibilities for the Wizard-of-Oz paradigm, and thereby for practical development of dialogue systems in general.

6 Acknowledgements

This research was funded by the Swedish research council project GENDIAL (VR #2007-6431).

References

Bell, L. (2003). *Linguistic adaptations in spoken human-computer dialogues. Empirical studies of user behavior*. Doctoral dissertation, Department of Speech, Music and Hearing, KTH, Stockholm.

Dohsaka, K., & Shimazu, A. (1997). System architecture for spoken utterance production in collaborative dialogue. In *Working Notes of IJCAI 1997 Workshop on Collaboration, Cooperation and Conflict in Dialogue Systems*.

Fraser, N. M., & Gilbert, G. N. (1991). Simulating speech systems. *Computer Speech and Language, 5*(1), 81-99.

Gustafson, J., & Edlund, J. (2008). expros: a toolkit for exploratory experimentation with prosody in customized diphone voices. In *Proceedings of Perception and Interactive Technologies for Speech-Based Systems (PIT 2008)* (pp. 293-296). Berlin/Heidelberg: Springer.

Hjalmarsson, A., Wik, P., & Brusk, J. (2007). Dealing with DEAL: a dialogue system for conversation training. In *Proceedings of SigDial* (pp. 132-135). Antwerp, Belgium.

Hjalmarsson, A. (2008). Speaking without knowing what to say... or when to end. In *Proceedings of SIGDial 2008*. Columbus, Ohio, USA.

Kempen, G., & Hoenkamp, E. (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science, 11*(2), 201-258.

Kilger, A., & Finkler, W. (1995). *Incremental Generation for Real-Time Applications*. Technical Report RR-95-11, German Research Center for Artificial Intelligence.

Levelt, W. J. M. (1989). *Speaking: From Intention to Articulation.* Cambridge, Mass., USA: MIT Press.

Raux, A., & Eskenazi, M. (2008). Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. In *Proceedings of SIGdial 2008*. Columbus, OH, USA.

Schlangen, D., & Skantze, G. (2009). A general, abstract model of incremental dialogue processing. In *Proceedings of EACL-09*. Athens, Greece.

Skantze, G., & Schlangen, D. (2009). Incremental dialogue processing in a micro-domain. In *Proceedings of EACL-09*. Athens, Greece.

Comparing Local and Sequential Models for Statistical Incremental Natural Language Understanding

Silvan Heintze, Timo Baumann, David Schlangen
Department of Linguistics
University of Potsdam, Germany
firstname.lastname@uni-potsdam.de

Abstract

Incremental natural language understanding is the task of assigning semantic representations to successively larger prefixes of utterances. We compare two types of statistical models for this task: a) *local models*, which predict a single class for an input; and b), *sequential models*, which align a sequence of classes to a sequence of input tokens. We show that, with some modifications, the first type of model can be improved and made to approximate the output of the second, even though the latter is more informative. We show on two different data sets that both types of model achieve comparable performance (significantly better than a baseline), with the first type requiring simpler training data. Results for the first type of model have been reported in the literature; we show that for our kind of data our more sophisticated variant of the model performs better.

1 Introduction

Imagine being at a dinner, when your friend Bert says "My friend, can you pass me the salt over there, please?". It is quite likely that you get the idea that something is wanted of you fairly early into the utterance, and understand what exactly it is that is wanted even before the utterance is over.

This is possible only because you form an understanding of the meaning of the utterance even before it is complete; an understanding which you refine—and possibly revise—as the utterance goes on. You understand the utterance *incrementally*. This is something that is out of reach for most current dialogue systems, which process utterances non-incrementally, *en bloc* (cf. (Skantze and Schlangen, 2009), *inter alia*).

Enabling incremental processing in dialogue systems poses many challenges (Allen et al., 2001; Schlangen and Skantze, 2009); we focus here on the sub-problem of modelling incremental understanding—a precondition for enabling truly interactive behaviour. More specifically, we look at statistical methods for learning mappings between (possibly partial) utterances and meaning representations. We distinguish between two types of understanding, which were sketched in the first paragraph above: a) forming a *partial* understanding, and b) *predicting* a complete understanding.

Recently, some results have been published on b), predicting utterance meanings, (Sagae et al., 2009; Schlangen et al., 2009). We investigate here how well this predictive approach works in two other domains, and how a simple extension of techniques (ensembles of slot-specific classifiers vs. one frame-specific one) can improve performance. To our knowledge, task a), computing partial meanings, has so far only been tackled with symbolic methods (e.g., (Milward and Cooper, 1994; Aist et al., 2006; Atterer and Schlangen, 2009));[1] we present here some first results on approaching it with statistical models.

Plan of the paper: First, we discuss relevant previous work. We then define the task of incremental natural language understanding and its two variants in more detail, also looking at how models can be evaluated. Finally, we present and discuss the results of our experiments, and close with a conclusion and some discussion of future work.

2 Related Work

Statistical natural language understanding is an active research area, and many sophisticated models for this task have recently been published, be that *generative* models (e.g., in (He and Young, 2005)), which learn a joint distribution over in-

[1] We explicitly refer to computation of incremental interpretations here; there is of course a large body of work on statistical incremental *parsing* (e.g., (Stolcke, 1995; Roark, 2001)).

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 9–16,
The University of Tokyo, September 24-25, 2010.

(Mairesse et al., 2009)	94.50
(He and Young, 2005)	90.3
(Zettlemoyer and Collins, 2007)	95.9
(Meza et al., 2008)	91.56

Table 1: Recent published f-scores for non-incremental statistical NLU, on the ATIS corpus

put, output and possibly hidden variables; or, more recently, *discriminative* models (e.g., (Mairesse et al., 2009)) that directly learn a mapping between input and output. Much of this work uses the ATIS corpus (Dahl et al., 1994) as data and hence is directly comparable. In Table 1, we list the results achieved by this work; we will later situate our results relative to this.

That work, however, only looks at mappings between complete utterances and semantic representations, whereas we are interested in the process of mapping semantic representations to successively larger utterance fragments. More closely related then is (Sagae et al., 2009; DeVault et al., 2009), where a maximum entropy model is trained for mapping utterance fragments to semantic frames. (Sagae et al., 2009) make the observation that often the quality of the prediction does not increase anymore towards the end of the utterance; that is, the meaning of the utterance can be predicted before it is complete.

In (Schlangen et al., 2009), we presented a model that predicts incrementally a specific aspect of the meaning of a certain type of utterance, namely the intended referent of a referring expression; the similarity here is that the output is of the same type regardless of whether the input utterance is complete or not.

(DeVault et al., 2009) discuss how such 'mind reading' can be used interactionally in a dialogue system, e.g. for completing the user's utterance as an indication of the system's grounding state. While these are interesting uses, the approach is somewhat limited by the fact that it is incremental only on the input side, while the output does not reflect how 'complete' (or not) the input is. We will compare this kind of incremental processing in the next section with one where the output is incremental as well, and we will then present results from our own experiments with both kinds of incrementality in statistical NLU.

3 Task, Evaluation, and Data Sets

3.1 The Task

We have said that the task of incremental natural language understanding consists in the assignment of semantic representations to progressively more complete prefixes of utterances. This description can be specified along several aspects, and this yields different versions of the task, appropriate for different uses. One question is what the assigned representations are, the other is what exactly they are assigned to. We investigate these questions here abstractly, before we discuss the instantiations in the next sections.

Let's start by looking at the types of representations that are typically assigned to *full* utterances. A type often used in dialogue systems is the *frame*, an attribute value matrix. (The attributes are here typically called *slots*.) These frames are normally typed, that is, there are restrictions on which slots can (and must) occur together in one frame. The frames are normally assigned to the utterance as a whole and not to individual words.

In an incremental setting, where the input potentially consists of an incomplete utterance, choosing this type of representation and style of assignment turns the task into one of *prediction* of the utterance meaning. What we want our model to deliver is a guess of what the meaning of the utterance is going to be, even if we have only seen a prefix of the utterance so far; we will call this "whole-frame output" below.[2]

Another popular representation of semantics in applied systems uses semantic *tags*, i.e., markers of semantic role that are attached to individual parts of the utterance. Such a style of assignment is inherently 'more incremental', as it provides a way to assign meanings that represent only what has indeed been said so far, and does not make assumptions about what will be said. The semantic representation of the prefix simply contains all and only the tags assigned to the words in the prefix; this will be called "aligned output" below. To our knowledge, the potential of this type of representation (and the models that create them) for incremental processing has not yet been explored; we present our first results below.

Finally, there is a hybrid form of representation and assignment. If we allow the output frames to 'grow' as more input comes in (hence possibly violating the typing of the frames as they are expected for full utterances), we get a form of representation with a notion of 'partial semantics' (as

[2] In (Schlangen and Skantze, 2009), this type of incremental processing is called "input incremental", as only the input is incrementally enriched, while the output is always of the same type (but may increase in quality).

only that is represented for which there is evidence in what has already been seen), but without *direct* association of parts of the representation and parts of the utterance or utterance prefix.

3.2 Evaluation

Whole-Frame Output A straightforward metric is *Correctness*, which can take the values 1 (output is exactly as expected) or 0 (output is *not* exactly as expected). Processing a test corpus in this way, we get one number for each utterance prefix, and, averaging this number, one measurement for the whole corpus.

This can give us a first indication of the general quality of the model, but because it weighs the results for prefixes of all lengths equally, it cannot tell us much about how well the incremental processing worked. In actual applications, we presumably do not expect the model to be correct from the very first word on, but do expect it to get better the longer the available utterance prefix becomes. To capture this, we define two more metrics: *first occurrence* (FO), as the position (relative to the eventual length of the full utterance) where the response was correct first; and *final decision* (FD) as the position from which on the response stayed correct (which consequently can only be measured if indeed the response stays correct).[3] The difference between FO and FD then tells us something about the stability of hypotheses of the model.

In some applications, we may indeed only be able to do further processing with fully correct—or at least correctly typed—frames; in which case *correctness* and FO/FD on frames are appropriate metrics. However, sometimes even frames that are only partially correct can be of use, for example if specific system reactions can be tied to individual slots. To give us more insight about the quality of a model in such cases, we need a metric that is finer-grained than binary correctness. Following (Sagae et al., 2009), we can conceptualise our task as one of retrieval of slot/value pairs, and use *precision* and *recall* (and, as their combination, *f-score*) as metrics. As we will see, it will be informative to plot the development of this score over the course of processing the utterance.

For these kinds of evaluations, we need as a gold standard only one annotation per utterance, namely the final frame.

Aligned Output As sequence alignments have more structure—there is a linear order between the tags, and there is exactly one tag per input token—correctness is a more fine-grained, and hence more informative, metric here; we define it as the proportion of tags that are correct in a sequence. We can also use precision and recall here, looking at each position in the sequence individually: Has the tag been recalled (true positive), or has something else been predicted instead (false negative, *and* false positive)? Lastly, we can also reconstruct frames from the tag sequences, where sequences of the same tag are interpreted as segmenting off the slot value. (And hence, what was several points for being right or wrong, one for each tag, becomes one, being either the correct slot value or not. We will discuss these differences when we show evaluations of aligned output.)

For this type of evaluation, we need gold-standard information of the same kind, that is, we need aligned tag sequences. This information is potentially more costly to create than the one final semantic representation needed for the whole-frame setting.

Hybrid Output As we will see below, the hybrid form of output ('growing' frames) is produced by ensembles of local classifiers, with one classifier for each possible slot. How this output can be evaluated depends on what type of information is available. If we only have the final frame, we can calculate f-score (in the hope that *precision* will be better than for the whole-frame classifier, as such a classifier ensemble can focus on predicting slots/value pairs for which there is direct evidence); if we do have sequence information, we can convert it to growing frames and evaluate against that.

3.3 The Data Sets

ATIS As our first dataset, we use the ATIS air travel information data (Dahl et al., 1994), as preprocessed by (Meza et al., 2008) and (He and Young, 2005). That is, we have available for each utterance a semantic frame as in (1), and also a tag sequence that aligns semantic concepts (same as the slot names) and words. One feature to note here about the ATIS representations is that the slot values / semantic atoms are just the words in the utterance. That is, the word itself is its own semantic representation, and no additional abstrac-

[3] These metrics of course can only be computed post-hoc, as during processing we do not know how long the utterance is going to be.

tion is performed. In this domain, this is likely unproblematic, as there aren't many different ways (that are to be expected in this domain) to refer to a given city or a day of the week, for example.

(1) "What flights are there arriving in Chicago after 11pm?"

$$\begin{bmatrix} \text{GOAL} = \text{FLIGHT} \\ \text{TOLOC.CITY\_NAME} = \text{Chicago} \\ \text{ARRIVE\_TIME.TIME\_RELATIVE} = \text{after} \\ \text{ARRIVE\_TIME.TIME} = \text{11pm} \end{bmatrix}$$

In our experiments, we use the ATIS training set which contains 4481 utterances, between 1 and 46 words in length (average 11.46; sd 4.34). The vocabulary consists of 897 distinct words. There are 3159 distinct frames, 2594 (or 58% of all frames) of which occur only once. Which of the 96 possible slots occur in a given frame is distributed very unevenly; there are some very frequent slots (like FROMLOC.CITYNAME or DEPART_DATE.DAY_NAME) and some very rare or even unique ones (e.g., ARRIVE_DATE.TODAY_RELATIVE, or TIME_ZONE).

Pentomino The second corpus we use is of utterances in a domain that we have used in much previous work (e.g., (Schlangen et al., 2009; Atterer and Schlangen, 2009; Fernández and Schlangen, 2007)), namely, instructions for manipulating puzzle pieces to form shapes. The particular version we use here was collected in a Wizard-of-Oz study, where the goal was to instruct the computer to pick up, delete, rotate or mirror puzzle tiles on a rectangular board, and drop them on another one. The user utterances were annotated with semantic frames and also aligned with tag sequences. We use here a frame representation where the slot value is a part of the utterance (as in ATIS), an example is shown in (2). (The corpus is in German; the example is translated here for presentation.) We show the full frame here, with all possible slots; unused slots are filled with "empty". Note that this representation is somewhat less directly usable in this domain than for ATIS; in a practical system, we'd need some further module (rule-based or statistical) that maps such partial strings to their denotations, as this mapping is less obvious here than in the travel domain.

(2) "Pick up the W-shaped piece in the upper right corner"

$$\begin{bmatrix} \text{action} = \text{"pick up"} \\ \text{tile} = \text{"the W-shaped piece} \\ \quad \text{in the upper right corner"} \\ \text{field} = \text{empty} \\ \text{rotpar} = \text{empty} \\ \text{mirpar} = \text{empty} \end{bmatrix}$$

The corpus contains 1563 utterances, average length 5.42 words (sd 2.35), with a vocabulary of 222 distinct words. There are 964 distinct frames, with 775 unique frames.

In both datasets we use transcribed utterances and not ASR output, and hence our results present an upper bound on real-world performance.

4 Local Models: Support Vector Machines

In this section we report the results of our experiments with local classifiers, i.e. models which, given an input, predict one out of a set of classes as an answer. Such models are very naturally suited to the *prediction task*, where the semantics of the full utterance is treated as its class, which is to be predicted on the basis of what possibly is only a prefix of that utterance. We will also look at a simple modification, however, which enables such models to do something that is closer to the task of computing partial meanings.

4.1 Experimental Setup

For our experiments with local models, we used the implementations of support vector machines provided by the WEKA toolkit (Witten and Frank, 2005); as baseline we use a simple majority class predictor.[4]

We used the standard WEKA tools to convert the utterance strings into word vectors. Training was always done with the full utterance, but testing was done on prefixes of utterances; i.e., a sentence with 5 words would be one instance in training, but in a testing fold it would contribute 5 instances, one with one word, one with two words, and so on.[5] Because of this special way of testing the classifiers, and also because of the modifica-

[4] We tried other classifiers (C4.5, logistic regression, naive Bayes) as well, and found comparable performance on a development set. However, because of the high time costs (some models needed > 40 hours for training and testing on modern multi-CPU servers) we do not systematically compare performance and instead focus on SVMs. In any case, our interest here is not in comparing classification algorithms, but rather in exploring approaches to the novel problem of statistical incremental NLU.

[5] On a development set, we tried training on utterance prefixes, but that degraded performance, presumably due to increase in ambiguous training instances (same beginnings of what ultimately are very different utterances).

tions described below, we had to provide our own methods for cross-validation and evaluation. For the larger ATIS data set, we used 10 folds in cross validation, and for the Pentomino dataset 20 folds.

4.2 Results

To situate our results, we begin by looking at the performance of the models that predict a **full frame**, when given a **full utterance**; this is the normal, "non-incremental" statistical NLU task.[6]

(3)

classf.	metric	ATIS	Pento
maj	correctness	1.07	1.79
maj	f-score	35.98	16.15
SVM	correctness	16.21	38.77
SVM	f-score	68.17	63.23

We see that the results for ATIS are considerably lower than the state of the art in statistical NLU (Table 1). This need not concern us too much here, as we are mostly interested in the dynamics of the incremental process, but it indicates that there is room for improvement with more sophisticated models and feature design. (We will discuss an example of an improved model shortly.) We also see a difference between the corpora reflected in these results: being *exactly* right (good correctness) seems to be harder on the ATIS corpus, while being *somewhat* right (good f-score) seems to be harder on the pento corpus; this is probably due to the different sizes of the search space of possible frame types (large for ATIS, small for pento).

What we are really interested in, however, is the performance when given only a **prefix of an utterance**, and how this develops over the course of processing successively larger prefixes. We can investigate this with Figure 1. First, look at the solid lines. The black line shows the average f-score at various prefix lengths (in 10% steps) for the ATIS data, the grey line for the pento corpus. We see that both lines show a relatively steady incline, meaning that the f-score continues to improve when more of the utterance is seen. This is interesting to note, as both (DeVault et al., 2009) and (Atterer et al., 2009) found that in their data, all that is to be known can often be found somewhat before the end of the utterance. That this does not work so well here is most likely due to the difference in domain and the resulting utterances. Utterances giving details about travel plans

[6]The results for ATIS are based on half of the overall ATIS data, as cross-validating the model on all data took prohibitively long, presumably due to the large number of unique frames / classes.

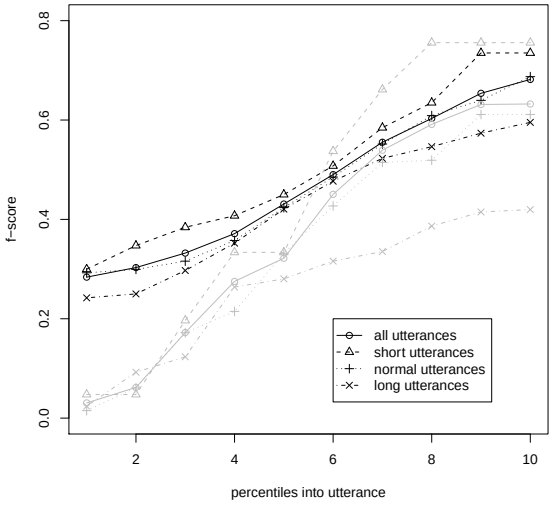


Figure 1: F-Score by Length of Prefix

are likely to present many important details, and some of them late into the utterance; cf. (1) above. The data from (DeVault et al., 2009) seems to be more conversational in nature, and, more importantly, presumable the expressible goals are less closely related to each other and hence can be read off of shorter prefixes.

As presented so far, the results are not very helpful for practical applications of incremental NLU. One thing one would like to know in a practical situation is how much the prediction of the model can be trusted for a given partial utterance. We would like to read this off graphs like those in the Figure—but of course, normally we cannot know what percentage of an utterance we have already seen! Can we trust this averaged curve if we do not know what length the incoming utterance will have?

To investigate this question, we have binned the test utterances into three classes, according to their length: "normal", for utterances that are of average length $\pm$ half a standard deviation, and "short" for all that are shorter, and "long" for all that are longer. The f-score curves for these classes are shown with the non-solid lines in Figure 1. We see that for ATIS there is not much variation compared to averaging over all utterances, and moreover, that the "normal" class very closely follows the general curve. On the pento data, the model seems to be comparably better for short utterances.

In a practical application, one could go with the assumption that the incoming utterance is going to be of normal length, and use the "normal"

curve for guidance; or one could devise an additional classifier that predicts the length-class of the incoming utterance, or more generally predicts whether a frame can already be trusted (DeVault et al., 2009). We leave this for future work.

As we have seen, the models that treat the semantic frame simply as a class label do not fare particularly well. This is perhaps not that surprising; as discussed above, in our corpora there aren't that many utterances with exact the same frame. Perhaps it would help to break up the task, and train **individual classifiers for each slot**?[7] This idea can be illustrated with (2) above. There we already included "unused" slots in the frame; if we now train classifiers for each slot, allowing them to predict "empty" in cases where a slot is unused, we can in theory reconstruct any frame from the ensemble of classifiers. To cover the pento data, the ensemble is small (there are 5 frames); it is considerably larger for ATIS, where there are so many distinct slots.

Again we begin by looking at the performance for **full utterances** (i.e., at 100% utterance length), but this time for **constructing the frame** from the reply of the classifier ensemble:

(4)

classf.	metric	ATIS	Pento
maj	correctness	0.16	0
maj	f-score	33.18	20.24
SVM	correctness	52.69	50.48
SVM	f-score	86.79	73.15

We see that this approach leads to an impressive improvement on the ATIS data (83.64 f-score instead of 68.17), whereas the improvement on the pento data is more modest (73.15 / 63.23).

Figure 2 shows the incremental development of the f-scores for the reconstructed frame. We see a similar shape in the curves; again a relatively steady incline for ATIS and a more dramatic shape for pento, and again some differences in behaviour for the different length classes of utterances. However, by just looking at the reconstructed frame, we are ignoring valuable information that the slot-classifier approach gives us. In some applications, we may already be able to do something useful with partial information; e.g., in the ATIS domain, we could look up an airport as soon as a FROMLOC becomes known. Hence, we'd want more fine-grained information, not just about when we can trust the whole frame, but rather about when

[7] A comparable approach is used for the non-incremental case for example by (Mairesse et al., 2009).

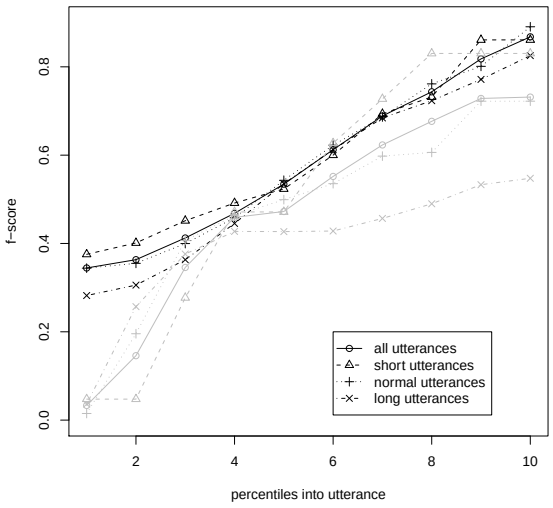


Figure 2: F-Score by Length of Prefix; Slot Classifiers

we can trust individual predicted slot values. (And so we move from the *prediction* task to the *partial representations* task.)

To explore this, we look at *First Occurrence* and *Final Decision* for some selected slots in Table 2. For some slots, the first occurrence (FO) of the correct value comes fairly early into the utterance (e.g., for the name of the airline it's at ca. 60%, for the departure city at ca. 63%, both with relatively high standard deviation, though) while others are found the first time rather late (goal city at 81%). This conforms well with intuitions about how such information would be presented in an utterance ("I'd like to fly on Lufthansa from Berlin to Tokyo").

We also see that the predictions are fairly stable: the number of cases where the slot value stays correct until the end is almost the same as that where it is correct at least once (FD *appl*icable vs. FO apl), and the average position is almost the same. In other words, the classifiers seem to go fairly reliably from "empty" (no value) to the correct value, and then seem to stay there. The overhead of unnecessary edits (EO) is fairly low for all slots shown in the table. (Ideally, EO is 0, meaning that there is no change except the one from "empty" to correct value.) All this is good news, as it means that a later module in a dialogue system can often begin to work with the partial results as soon as a slot-classifier makes a non-empty prediction. In an actual application, how trustworthy the individual classifiers are would then be read off statistics

slot name	avg FO	stdDev	apl	avg FD	stdDev	apl	avg EO	stdDev	apl
AIRLINE_NAME	0.5914	0.2690	506	0.5909	0.2698	501	0.5180	0.5843	527
DEPART_TIME.PERIOD_OF_DAY	0.7878	0.2506	530	0.7992	0.2476	507	0.2055	0.5558	579
FLIGHT_DAYS	0.4279	0.2660	37	0.4279	0.2660	37	0.0000	0.0000	37
FROMLOC.CITY_NAME	0.6345	0.1692	3633	0.6368	0.1692	3554	0.1044	0.4526	3718
ROUND_TRIP	0.5366	0.2140	287	0.5366	0.2140	287	0.0104	0.1015	289
TOLOC.CITY_NAME	0.8149	0.1860	3462	0.8162	0.1856	3441	0.2348	0.5723	3628
frames	0.9745	0.0811	2382	0.9765	0.0773	2361	0.7963	1.1936	4481

Table 2: FO/FD/EO for some selected slots; averaged over utterances of all lengths

like these, given a corpus from the domain.

To conclude this section, we have shown that classifiers that predict a complete frame based on utterance prefixes have a somewhat hard task here (harder, it seems, than in the corpus used in (Sagae et al., 2009), where they achieve an f-score of 87 on transcribed utterances), and the prediction results improve steadily throughout the whole utterance, rather than reaching their best value before its end. When the task is 'spread' over several classifiers, with each one responsible for only one slot, performance improves drastically, and also, the results become much more 'incremental'. We now turn to models that by design are more incremental in this sense.

5 Sequential Models: Conditional Random Fields

5.1 Experimental Setup

We use Conditional Random Fields (Lafferty et al., 2001) as our representative of the class of sequential models, as implemented in CRF++.[8] We use a simple template file that creates features based on a left context of three words.

Even though sequential models have the potential to be truly incremental (in the sense that they could produce a new output when fed a new increment, rather than needing to process the whole prefix again), CRF++ is targeted at tagging applications, and expects full sequences. We hence test in the same way as the SVMs from the previous section, by computing a new tag sequence for each prefix. Training again is done only on full utterances / tag sequences.

We compare the CRF results against two baselines. The simplest consists of just always choosing the most frequent tag, which is "O" (for *other*, marking material that does not contribute directly to the relevant meaning of the utterance, such as "please" in "I'd like to return on Monday, please."). The other baseline tags each word with its most frequent training data tag.

[8] http://crfpp.sourceforge.net/

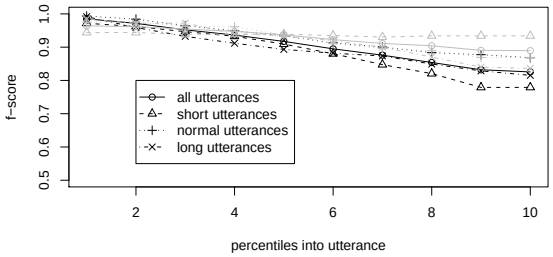


Figure 3: F-Score by Length of Prefix

ATIS	Corr.	Tag F-Score	Frame F-Score
CRF	93.38	82.56	76.10
Maj	85.14	60.86	48.08
O	63.43	00.31	00.31
Pento	Corr.	Tag F-Score	Frame F-Score
CRF	89.19	88.95	76.94
Maj	80.20	80.13	65.94
O	5.90	0.19	0.19

Table 3: Results of CRF models

5.2 Results

We again begin by looking at the limiting case, the results for **full utterances** (i.e., at the 100% mark).

Table 3 show three sets of results for each corpus. *Correctness* looks at the proportion of tags in a sequence that were correct. This measure is driven up by correct recognition of the dummy tag "o"; as we can see, this is quite frequently correct in ATIS, which drives up the "always use O"-baseline. Tag F-Score values the important tags higher; we see here, though, that the majority baseline (each word tagged with its most frequent tag) is surprisingly good. It is solidly beaten for the ATIS data, though. On the pento data, with its much smaller tagset (5 as opposed to 95), this baseline comes very high, but still the learner is able to get some improvement. The last metric evaluates reconstructed frames. It is stricter, because it offers less potential to be right (a sequence of the same tag will be translated into one slot value, turning several opportunities to be right into

only one).

The incremental dynamics looks quite different here. Since the task is not one of prediction, we do not expect to get better with more information; rather, we start at an optimal point (when nothing is said, nothing can be wrong), and hope that we do not amass too many errors along the way. Figure 3 confirms this, showing that the classifier is better able to keep the quality for the pento data than for the ATIS data. Also, there is not much variation depending on the length of the utterance.

6 Conclusions

We have shown how sequential and local statistical models can be used for two variants of the incremental NLU task: prediction, based on incomplete information, and assignment of partial representations to partial input. We have shown that breaking up the prediction task by using an ensemble of classifiers improves performance, and creates a hybrid task that sits between prediction and incremental interpretation.

While the objective quality as measured by our metrics is quite good, what remains to be shown is how such models can be integrated into a dialogue system, and how what they offer can be turned into improvements on interactivity. This is what we are turning to next.

Acknowledgements Funded by ENP grant from DFG.

References

G.S. Aist, J. Allen, E. Campana, L. Galescu, C.A. Gomez Gallo, S. Stoness, M. Swift, and M Tanenhaus. 2006. Software architectures for incremental understanding of human speech. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Pittsburgh, PA, USA, September.

James Allen, George Ferguson, and Amanda Stent. 2001. An architecture for more realistic conversational systems. In *Proceedings of the conference on intelligent user interfaces*, Santa Fe, USA, June.

Michaela Atterer and David Schlangen. 2009. RUBISC – a robust unification-based incremental semantic chunker. In *Proceedings of the 2nd International Workshop on Semantic Representation of Spoken Language (SRSL 2009)*, Athens, Greece, March.

Michaela Atterer, Timo Baumann, and David Schlangen. 2009. No sooner said than done? testing incrementality of semantic interpretations of spontaneous speech. In *Proceedings of Interspeech 2009*, Brighton, UK, September.

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 *Proceedings of the workshop on Human Language Technology*, pages 43–48, Plainsboro, NJ, USA.

David DeVault, Kenji Sagae, and David Traum. 2009. Can i finish? learning when to respond to incremental interpretation results in interactive dialogue. In *Proceedings of the 10th Annual SIGDIAL Meeting on Discourse and Dialogue (SIGDIAL'09)*, London, UK, September.

Raquel Fernández and David Schlangen. 2007. Referring under restricted interactivity conditions. In Simon Keizer, Harry Bunt, and Tim Paek, editors, *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 136–139, Antwerp, Belgium, September.

Yulan He and Steve Young. 2005. Semantic processing using the hidden vector state model. *Computer Speech and Language*, 19(1):85–106.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*, pages 282–289.

F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, and S. Young. 2009. Spoken language understanding from unaligned data using discriminative classification models. In *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, Taipei, Taiwan, April.

Ivan Meza, Sebastian Riedel, and Oliver Lemon. 2008. Accurate statistical spoken language understanding from limited development resources. In *In Proceedings of ICASSP*.

David Milward and Robin Cooper. 1994. Incremental interpretation: Applications, theory, and relationships to dynamic semantics. In *Proceedings of COLING 1994*, pages 748–754, Kyoto, Japan, August.

Brian Roark. 2001. *Robust Probabilistic Predictive Syntactic Processing: Motivations, Models, and Applications*. Ph.D. thesis, Department of Cognitive and Linguistic Sciences, Brown University.

Kenji Sagae, Gwen Christian, David DeVault, and David Traum. 2009. Towards natural language understanding of partial speech recognition results in dialogue systems. In *Short paper proceedings of the North American chapter of the Association for Computational Linguistics - Human Language Technologies conference (NAACL-HLT'09)*, Boulder, Colorado, USA, June.

David Schlangen and Gabriel Skantze. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, pages 710–718, Athens, Greece, March.

David Schlangen, Timo Baumann, and Michaela Atterer. 2009. Incremental reference resolution: The task, metrics for evaluation, and a bayesian filtering model that is sensitive to disfluencies. In *Proceedings of SIGdial 2009, the 10th Annual SIGDIAL Meeting on Discourse and Dialogue*, London, UK, September.

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 Association for Computational Linguistics (EACL 2009)*, pages 745–753, Athens, Greece, March.

Andreas Stolcke. 1995. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, 21(2):165–201.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, USA, 2nd edition.

Luke S. Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of EMNLP-CoNLL*.

Invited Talk

Dynamic Adaptation in Dialog Systems

Marilyn Walker

University of California, Santa Cruz

A hallmark of human robust intelligence is the ability to flexibly and dynamically adapt behavior to the current situation. For dialog behavior, this entails adaptation to features of both the dialog partner (e.g., relationship, age, personality) and the dialog situation (e.g., task context, asynchronous interaction, limited modalities such as voice-only communication). We don't completely understand how humans do this, nor do we have the ability to produce such dynamically adaptable behavior in human computer dialog interaction. In this talk I will discuss our recent work on dynamic adaptation to the user, and present some experimental results showing that it is possible to automatically generate both verbal and nonverbal system behaviors that are perceived by the user as reliably expressing particular system personalities. I will describe two of my current projects at UCSC that are integrating these capabilities into mobile dialogue systems: SpyFeet, a role playing augmented reality game for encouraging girls to exercise, and Skipper, a dialogue system that gives pedestrians directions in both urban and campus environments.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, page 17,
The University of Tokyo, September 24-25, 2010.

Modeling User Satisfaction Transitions in Dialogues from Overall Ratings

Ryuichiro Higashinaka[†], **Yasuhiro Minami**[‡], **Kohji Dohsaka**[‡], and **Toyomi Meguro**[‡]
[†] NTT Cyber Space Laboratories, NTT Corporation
[‡] NTT Communication Science Laboratories, NTT Corporation
higashinaka.ryuichiro@lab.ntt.co.jp
{minami,dohsaka,meguro}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes a novel approach for predicting user satisfaction transitions during a dialogue only from the ratings given to entire dialogues, with the aim of reducing the cost of creating reference ratings for utterances/dialogue-acts that have been necessary in conventional approaches. In our approach, we first train hidden Markov models (HMMs) of dialogue-act sequences associated with each overall rating. Then, we combine such rating-related HMMs into a single HMM to decode a sequence of dialogue-acts into state sequences representing to which overall rating each dialogue-act is most related, which leads to our rating predictions. Experimental results in two dialogue domains show that our approach can make reasonable predictions; it significantly outperforms a baseline and nears the upper bound of a supervised approach in some evaluation criteria. We also show that introducing states that represent dialogue-act sequences that occur commonly in all ratings into an HMM significantly improves prediction accuracy.

1 Introduction

In recent years, there has been intensive work on the automatic evaluation of dialogues (Walker et al., 1997; Möller et al., 2008). Automatic evaluation makes it possible to predict the performance of dialogue systems without the costly process of performing surveys with human subjects, leading to a rapid improvement cycle for dialogue systems. It is also useful for detecting problematic situations in an ongoing dialogue (Walker et al., 2002; Herm et al., 2008; Kim, 2007). In these studies, the typical approach is to train a prediction model, such as a regression or classification model, using features representing the whole or a part of a dialogue together with human reference labels (e.g., reference ratings). However, creating such reference labels by hand can be extremely costly when we want to predict user satisfaction transitions during a dialogue because we need to create reference labels after each utterance/dialogue-act in the training data (Engelbrecht et al., 2009).

This paper proposes a novel approach for predicting user satisfaction transitions during a dialogue only from the dialogues with overall ratings. The approach makes it possible to avoid creating reference labels for utterances/dialogue-acts and only requires a single reference label for each dialogue. More specifically, we predict the user satisfaction rating after each dialogue-act in a dialogue only by using dialogues with dialogue-level (overall) user satisfaction ratings as training data. Our basic approach is to train hidden Markov models (HMMs) of dialogue-act sequences associated with each overall rating and combine such rating-related HMMs into a single HMM. We use this combined HMM to decode a sequence of dialogue-acts by the Viterbi algorithm (Rabiner, 1990) into state sequences that indicate from which rating-related HMM each dialogue-act is most likely to have been generated, leading to our rating predictions for the dialogue-acts. This paper experimentally examines the validity of our approach and explores several model topologies for possible improvement.

In Section 2, we review related work on automatic evaluation of dialogues. In Section 3, we describe our approach in detail. In Section 4, we describe the experiment we performed to verify our approach and present the results. In Section 5, we summarize and mention future work.

2 Related Work

Regression models are typically utilized for evaluating the quality of an entire dialogue. Most famously, the PARADISE framework (Walker et al., 1997) learns from data a linear regression model that predicts dialogue-level user satisfaction from various objective characteristics of a dialogue that concern task success and dialogue costs. This framework is widely used today and a number of extensions have been proposed to improve the prediction performance (Möller et al., 2008); how-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 18–27,
The University of Tokyo, September 24-25, 2010.

ever, it is not aimed at predicting user satisfaction transitions.

Classification models are widely employed to detect problematic situations in an ongoing dialogue. Walker et al. (2002) developed the Problematic Dialogue Predictor for the "How May I Help You" system (Gorin et al., 1997) to robustly transfer problematic calls to human operators in call routing tasks. They derive speech recognition, language understanding, and dialogue management features from the first few turns of a dialogue and apply a decision tree classifier to detect problematic calls. For a similar task, Hirschberg et al. (2004) and Herm et al. (2008) used prosodic and emotional features. Kim (2007) recently proposed an approach for online call quality monitoring so that problematic calls can be transferred to human operators as quickly as possible rather than waiting for the first few turns.

N-grams and HMM-based approaches have also been actively studied. Hara et al. (2010) proposed predicting the most likely user satisfaction level of a dialogue by using N-grams of dialogues for each satisfaction level in the music navigation domain. Isomura et al. (2009) used HMMs to evaluate the naturalness of a dialogue in their interview system. They trained HMMs that model dialogue-act sequences between human subjects and used them to evaluate human-machine dialogues by the output probabilities of the HMMs. Recently, there have been approaches to predict user satisfaction transitions by evaluating the quality of individual utterances in a dialogue. For example, Engelbrecht et al. (2009) predicted user satisfaction ratings after each user utterance by HMMs trained from utterance-level features and utterance-level reference ratings.

The problem with these approaches is that they require a lot of training data, especially when we want to predict the quality of smaller units such as utterances. Our aim is to reduce such cost. Our work is similar to Engelbrecht's work (Engelbrecht et al., 2009) in that we use HMMs to predict user satisfaction transitions during a dialogue but different in that we only use dialogue-level ratings to model dialogue-act-level user satisfaction transitions.

3 Approach

We aim to predict user satisfaction transitions only from dialogues with overall ratings. More formally, given a dialogue d_i of a set of dialogues D ($= \{d_1 \ldots d_N\}$), we want to predict the user satisfaction rating after each dialogue-act in d_i, namely, $r'(da(d_i, 1)) \ldots r'(da(d_i, m_i))$, using D with their dialogue-level ratings $r(d_1) \ldots r(d_N)$.

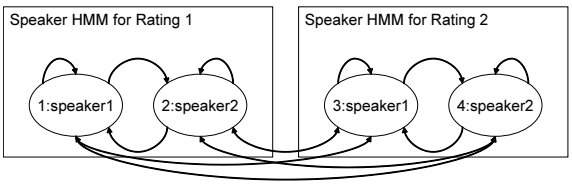


Figure 1: SHMMs connected ergodically. In the figure, an oval marked with speaker1/speaker2 indicates a state that emits speaker1/speaker2's dialogue-acts. Arrows denote transitions and numbers before speaker1/speaker2 are state IDs. Boxes group together the states related to a particular overall rating.

Here, $da(d_i, l)$ denotes the l-th dialogue-act in d_i, N the total number of dialogues, and m_i the total number of dialogue-acts in d_i.

Our basic idea is to train HMMs representing dialogue-act sequences of dialogues for each overall rating and combine these rating-related HMMs into a single HMM that can assign ratings for dialogue-acts by estimating from which HMM each dialogue-act has most likely to have been generated by the Viterbi decoding. We use HMMs because they can deal with sequences that evolve over time and have been successfully utilized to model and evaluate dialogue-act sequences (Shirai, 1996; Isomura et al., 2009; Engelbrecht et al., 2009). The generative feature of an HMM is also useful when we want to build a probabilistic dialogue manager that produces the most likely dialogue-act sequences (Hori et al., 2008) or that aims to maximize a reward function in partially observable Markov decision processes (Williams and Young, 2007; Minami et al., 2009).

When there are K levels of user satisfaction as overall ratings, we create K HMMs each of which is trained using the dialogue-act sequences in dialogues $D_k \subset D$, where $D_k = \{\forall d_i, |r(d_i) = k\}$. We use the EM-algorithm to train HMMs. Here, we assume that each HMM has two states, each of which emits dialogue-acts of one of the conversational participants. This type of HMM is called a speaker HMM (**SHMM**) and has been successfully utilized to model two-party conversation (Meguro et al., 2009).

As an illustrative example, Fig. 1 shows two SHMMs for ratings 1 and 2 that are connected ergodically. We can simply use these connected SHMMs (namely, states 1, 2, 3, and 4) to decode a sequence of dialogue-acts into state sequences and thereby obtain rating predictions. For example, if the optimal state sequence obtained by the Viterbi decoding is $\{4, 2, 1, 3, 2\}$, we can convert it into ratings <2, 1, 1, 2, 1> using the ratings associated with the states.

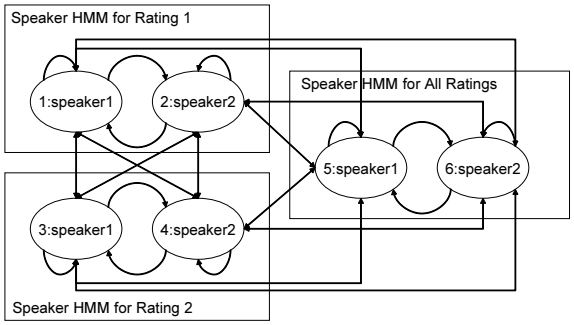


Figure 2: SHMMs with an additional SHMM trained from all dialogues.

Introducing Common States: The simple ergodic model may not be sufficient for appropriately assigning ratings to input dialogue-act sequences because it is often the case that there are dialogue-act sequences, such as greetings and question-answer pairs, that commonly occur in every dialogue. If we forcefully assign a rating for such dialogue-act sequences, it may result in degrading the prediction accuracy. Therefore, in addition to the simple ergodic model, we introduce another SHMM that represents dialogue-act sequences of dialogues for all ratings (see Fig. 2). This additional SHMM models dialogue-act sequences that occur commonly in all dialogues and it can simply be trained using all dialogues. Hence, we call the states in this SHMM **common states**. When this SHMM is added to the ergodic model, it may be possible to reduce the possibility of our having to forcefully assign inappropriate scores to common dialogue-act sequences. In this model, when the optimal state sequence is {1, 4, 5, 6, 2}, the predicted ratings become <1, 2, 0, 0, 1>. Here, we assume that the SHMM for all ratings corresponds to rating 0, which is reasonable because common dialogue-acts should not affect ratings. The obtained ratings can also be interpreted as <1, 2, 2, 2, 1> when we assume that the rating of a dialogue-act is taken over from the previous turn.

Using Concatenated Training: We have so far presented two model topologies, one with K SHMMs connected ergodically and the other with $K+1$ SHMMs having an additional SHMM representing all ratings. However, we still have a problem; that is, we need to find optimal transition probabilities between the SHMMs of different ratings. Our solution is to use concatenated training (Lee, 1989). The procedure for concatenated training is illustrated in Fig. 3 and has the following three steps.

step 1 Train an SHMM M_k ($M_k \in M, 1 \leq k \leq K$) using dialogues D_k, where $D_k = \{\forall d_i | r(d_i) = k\}$, and an SHMM M_0 using all dialogues; i.e., D. Here, K means the maximum level of user satisfaction and $r(d_i)$ the rating assigned to d_i.

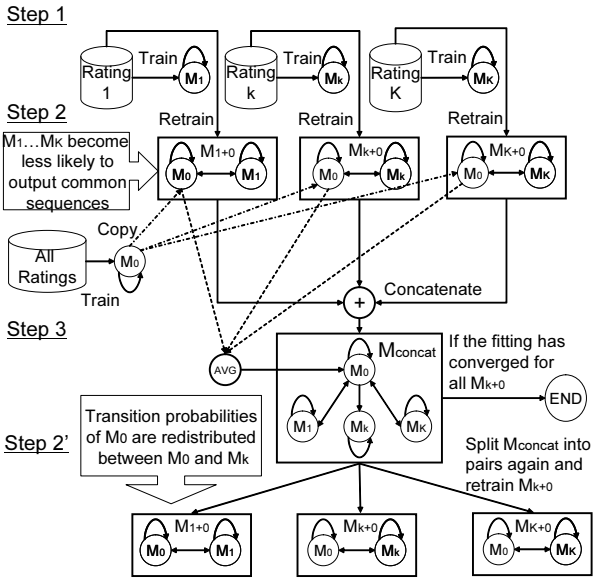


Figure 3: Three steps to combine SHMMs using concatenated training.

step 2 Connect each $M_k \in M$ with a copy of M_0 using equal initial and transition probabilities (we call this connected model M_{k+0}) and retrain M_{k+0} with $\forall d_i \in D_k$, where $r(d_i) = k$.

step 3 Merge all models M_{k+0} ($1 \leq k \leq K$) to produce one concatenated HMM (M_{concat}). Here, the output probabilities of the copies of M_0 are averaged over K when all models are merged to create a combined model. If the fitting of all M_{k+0} models has converged against the training data, exit this procedure; otherwise, go to step 2 by connecting a copy of M_0 and M_k for all k. Here, the transition probabilities from M_0 to $M_l (l \neq k)$ are summed and equally distributed between the copied M_0's self-loop and transitions to the states in M_k.

In concatenated training, the transition and output probabilities can be optimized between M_0 and M_k, meaning that the output probabilities of dialogue-act sequences that are common and also found in M_k can be moved from M_k to M_0. This makes the distribution of M_k sharp (not broad/uniform), making it likely to output only the dialogue-acts specific to a rating k. As regards M_0, its distribution of output probabilities can also be sharpened for dialogue-acts that occur commonly in all ratings. This sharpening of distributions is likely to be helpful in assigning

appropriate ratings to dialogue-act sequences. In the next section, we experimentally examine how these proposed HMMs perform in modeling and predicting user satisfaction transitions in dialogue.

4 Experiment

To verify our approach, we first prepared dialogue data. Then, we trained our HMMs and compared them with a random baseline and an upper bound that uses a supervised approach; that is, an HMM is trained using reference labels on the dialogue-act level.

4.1 Dialogue Data

We used dialogues in two domains; the animal discussion (**AD**) domain and the attentive listening (**AL**) domain. All dialogues are in Japanese. In both domains, the data we used were text dialogues. We did not use spoken dialogue data because we wanted to avoid particular problems of voice, such as filled pauses and overlaps, although we aim to deal with spoken dialogue in the future.

4.1.1 Animal Discussion

We used the dialogue data in the AD domain that we previously collected (Higashinaka et al., 2008). In this domain, the system and user talk about likes and dislikes about animals via a text chat interface. The data consist of 1000 dialogues between a dialogue system and 50 human users. Each user conversed with the system 20 times, including two example dialogues at the beginning. All user/system utterances have been annotated with dialogue-acts. There are 29 dialogue-act types including those related to self-disclosure, question, response, and greetings. For example, a dialogue-act DISC-P denotes one's self-disclosure about a proposition `P`. Here, `P` is either `like(X,A)` or `dislike(X,A)` where `X` is a conversational participant and `A` a certain animal. DISC-R denotes one's self-disclosure of a reason for a proposition. See (Higashinaka et al., 2008) for the details of the dialogue-acts.

For our experiment, we created two subsets of the data. We first extracted 180 dialogues by taking all 18 non-example dialogues for the initial ten users sorted by user ID (**AD-SUB1**; 4147 user dialogue-acts and 6628 system dialogue-acts). Then, from AD-SUB1, we randomly extracted nine dialogues per user to form another subset of 90 dialogues (**AD-SUB2**; 2050 user dialogue-acts and 3290 system dialogue-acts). An annotator, who was not one of the authors, labeled AD-SUB1 with dialogue-level user satisfaction ratings and AD-SUB2 with utterance-level ratings. More specifically, each dialogue/utterance was given three different user satisfaction ratings related to "Smoothness of the conversation", "Closeness perceived by the user towards the system", and "Willingness to continue the conversation". The ratings ranged from 1 to 7, where 1 is the worst and 7 the best (see Fig. 4 for examples of utterance-level and overall ratings given by the annotator for an excerpt of a dialogue). In a manner similar to (Evanini et al., 2008), we used a third-person's user satisfaction rating for the sake of consistency.

	Utterance (dialogue-acts)	Sm	Cl	Wi
SYS	Do you like rabbits? (DA: Q-DISC-P)	6	6	6
USR	I like rabbits. They are cute. (DA: DISC-P, DISC-R)			
SYS	Indeed they are cute. (DA: REPEAT)	6	6	6
SYS	Tell me why you like rabbits. (DA: Q-DISC-R-OTHER)	6	5	6
USR	I like them because they are small and warm. (DA: DISC-P-R)			
SYS	You like them because they are warm. (DA: REPEAT)	7	5	7
	Overall rating for the dialogue	7	5	6

Figure 4: Excerpt of a dialogue with utterance-level user satisfaction ratings for smoothness (Sm), closeness (Cl), and willingness (Wi) in the AD domain. SYS and USR denote system and user, respectively. The dialogue was translated by the authors.

For utterance-level ratings, the annotator carefully read each utterance and gave ratings after each system utterance according to how she would have felt after receiving each system utterance if she had been the user in the dialogue. To make the situation more realistic, she was not allowed to look down at the dialogue after the current utterance. At the beginning of a dialogue, the ratings always started from four (neutral). When the annotator gave dialogue-level ratings, she looked through the entire dialogue and rated its quality (smoothness, closeness, and willingness) according to how she would have felt after having had the dialogue in question.

4.1.2 Attentive Listening

We collected human-human listening-oriented dialogues in a manner similar to (Meguro et al., 2009). In this AL domain, a listener attentively listens to the other in order to satisfy the speaker's desire to speak and to make himself/herself heard. We collected such listening-oriented dialogues using a website where users taking the roles of listeners and speakers were matched up to have conversations. There were ten listeners who always stayed at the website and 37 speakers who could talk to them anytime the listeners were available. They were all paid for their participation. A conversation was done through a text-chat interface.

The use of facial and other non-linguistic expressions were not allowed for analysis purposes. The participants were instructed to end the conversation approximately after ten minutes. Within a three-week period, each speaker was instructed to have at least two conversations a day, resulting in our collecting 1260 listening-oriented dialogues.

Two independent annotators labeled each utterance with 40 dialogue-act types, including those related to self-disclosure, question, internal argument, sympathy, and information giving. The inter-annotator agreement was reasonable, with 0.57 in Cohen's κ. Although we cannot describe the full details of our dialogue-acts for lack of space, we have dialogue-acts DISC-EVAL-POS for one's self-disclosure of his/her positive evaluation towards a certain entity, DISC-EXP for one's self-disclosure of his/her experience, and SELF-Q-DESIRE for one's internal argument about his/her desire (e.g., "Have I ever wanted to go abroad?"). We used the dialogue-act annotation of one of the annotators in this work.

An annotator gave dialogue-level user satisfaction ratings to all 1260 dialogues (**AL-ALL**; 31779 speaker dialogue-acts and 28681 listener dialogue-acts). Then, we made a subset of the data by randomly selecting ten dialogues for each of the ten listeners to obtain 100 dialogues (**AL-SUB1**; 2453 speaker dialogue-acts and 2197 listener dialogue-acts). Finally, the annotator gave utterance-level ratings to AL-SUB1. The utterance-level ratings were given only after listeners' utterances. The annotator gave three ratings as in the AD domain; namely, smoothness, closeness, and good listening. Instead of willingness, we have a "good listener" criterion asking for how good the annotator thinks the listener is from the viewpoint of attentive listening; for example, how well the listener is making it easy for the speaker to speak. All ratings ranged from 1 to 7. See Fig. 5 for a sample dialogue in the AL domain with utterance-level and overall ratings given by the annotator.

	Utterance (dialogue-acts)	Sm	Cl	GL
LIS	You know, in spring, Japanese food tastes delicious. (DA: DISC-EVAL-POS)	5	5	5
SPK	This time every year, I make a plan to go on a healthy diet. But . . . (DA: DISC-HABIT)			
LIS	Uh-huh (DA: ACK)	6	5	6
SPK	The temperature goes up suddenly! (DA: INFO)			
SPK	It's always too late! (DA: DISC-EVAL-NEG)			
LIS	Clothing worn gets less and less while not being able to lose weight. (DA: DISC-FACT)	6	6	6
SPK	Well, people around me soon get used to my body shape though. (DA: DISC-FACT)			
	Overall rating for the dialogue	7	7	7

Figure 5: Excerpt of a dialogue with utterance-level user satisfaction ratings for smoothness (Sm), closeness (Cl), and good listener (GL) in the AL domain. SPK and LIS denote speaker and listener, respectively. Both the speaker and listener are human.

4.2 Training HMMs

From the dialogue data and their dialogue-level ratings, we created our proposed HMMs. We had five topology variations:

ergodic0: The simple ergodic model with no additional SHMM for all ratings. See Fig. 1 for the topology. This HMM has 7 SHMMs connected ergodically with equal initial/transition probabilities.

ergodic1: The simple ergodic model with an additional SHMM for all ratings. See Fig. 2 for the topology. This HMM has 8 (7 + 1) SHMMs connected ergodically with equal initial/transition probabilities.

ergodic2: Same as ergodic1 except that the number of common states is doubled so that common dialogue-act sequences can be more accurately modeled. Note that without concatenated training, SHMMs for each rating may also have sharp distributions for common sequences. One possible solution to avoid this is to sharpen the distributions of common states by increasing its number of states.

concat1: 8 (7 + 1) SHMMs combined using concatenated training. See Fig. 3 for the topology.

concat2: Same as concat1 except that the number of common states is doubled.

[See Appendices A and B for the actual examples of the obtained models]

4.2.1 Baseline and Upper Bound

We created the following baseline (random) and upper bound (supervised) models for comparison:

random: This outputs ratings 1–7 at random.

supervised: This is an HMM trained in a manner similar to (Engelbrecht et al., 2009). This model is the same as ergodic0 in topology but different in that the initial, transition, and output probabilities are trained in a supervised manner using the dialogue-acts and dialogue-act-level reference ratings in AD-SUB2 and AL-SUB1. Since we only have ratings for system/listener utterances in the corpora, in order to make training data, we assumed that the ratings for dialogue-acts corresponding to user/speaker utterances were the same as

those after the previous system/listener utterances. This model simulates the ideal situation where we possess user satisfaction ratings for all dialogue-acts in the data.

4.3 Evaluation Procedure

We performed a ten-fold cross validation. We first separated utterance-level labeled data (i.e., AD-SUB2 or AL-SUB1) into 10 disjoint sets. Then, for each set S, we used dialogue-level labeled data (i.e., AD-SUB1 or AL-ALL) excluding S for training HMMs. Here, 'supervised' only used the utterance-level labeled data excluding S for training. Then, we made the models (i.e., ergodic0, ergodic1, ergodic2, concat1, concat2, random and supervised) output rating sequences for the dialogue-acts in S and evaluated them with the reference ratings in S. We repeated this process ten times to evaluate the overall performance.

Since utterance-level ratings are provided only after system/listener utterances, we only evaluated ratings after dialogue-acts corresponding to system/listener utterances. When a system/listener utterance contained multiple dialogue-acts, the dialogue-acts were assumed to have the same rating as that utterance. When the output rating sequences contain 0, which can be the case for ergodic1–2 and concat1–2, the 0 is replaced by the most previous non-zero rating. When 0 is found at the beginning of a dialogue, it remained 0. Although our reference ratings always started with four (cf. Section 4.1.1), we did not use this information to fill initial zeros because we wanted to evaluate the prediction accuracy when we do not have any prior knowledge. Since some models may benefit from avoiding evaluating dialogue-acts at the beginning because of these zeros, we simply compared the rating sequences where all models produced non-zero values. For example, when we have three output rating sequences <0,5,6,0,4>, <0,0,1,2,0>, and <1,2,3,4,5> for a given dialogue-act sequence, the zeros that follow non-zero values are first filled with their preceeding values, and thereby we obtain <0,5,6,6,4>, <0,0,1,2,2>, and <1,2,3,4,5>. Then, by cropping the common non-zero span, we obtain <6,6,4>, <1,2,2>, and <3,4,5>, and use these rating sequences for evaluation.

4.3.1 Evaluation Criteria

We used two kinds of evaluation criteria: one for evaluating individual matches and the other for evaluating distributions.

Evaluating Individual Matches: We used the match rate and mean absolute error to evaluate the matching of reference and hypothesis rating sequences. They are derived by the equations shown below. In the equations, R ($= \{R_1 \dots R_L\}$) and H ($= \{H_1 \dots H_L\}$) denote reference and hypothesis rating sequences for a dialogue, respectively. L is the length of R and H (Note that they have the same length).

• **Match Rate (MR)**

$$\mathrm{MR}(R, H) = \frac{1}{L} \sum_{i=1}^{L} \mathrm{match}(R_i, H_i), \quad (1)$$

where 'match' returns 1 or 0 depending on whether a rating in R matches that in H.

• **Mean Absolute Error (MAE)**

$$\mathrm{MAE}(R, H) = \frac{1}{L} \sum_{i=1}^{L} |R_i - H_i|. \quad (2)$$

Evaluating Distributions: In generative models, it is important that the output distribution matches that of the reference. Therefore, we additionally use Kullback-Leibler divergence, match rate per rating, and mean absolute error per rating. The Kullback-Leibler divergence evaluates the shape of output distributions. The match rate per rating and mean absolute error per rating evaluate how accurately each individual rating can be predicted; namely, the accuracy for predicting dialogue-acts with one rating is equally valued with those for other ratings irrespective of the distribution of ratings in the reference. It is important to use these metrics in the practical as well as information theoretic sense because it is no use predicting only easy-to-guess ratings; we should be able to correctly predict rare but still important cases. For example, rating 1 in human-human dialogue is quite rare; however, predicting it is very important for detecting problematic situations in a dialogue.

• **Kullback-Leibler Divergence (KL)**

$$\mathrm{KL}(\mathbf{R}, \mathbf{H}) = \sum_{r=1}^{K} \mathrm{P}(\mathbf{H}, r) \cdot \log\left(\frac{\mathrm{P}(\mathbf{H}, r)}{\mathrm{P}(\mathbf{R}, r)}\right), \quad (3)$$

where K is the maximum user satisfaction rating (i.e. 7 in this experiment), $\mathbf{R}$ and $\mathbf{H}$ denote the sequentially concatenated reference/hypothesis rating sequences of the entire dialogues, and $\mathrm{P}(*, r)$ denotes the occurrence probability that a rating r is found in an arbitrary rating sequence.

• **Match Rate per rating (MR/r)**

$$\mathrm{MR/r}(\mathbf{R}, \mathbf{H}) = \frac{1}{K} \sum_{\mathrm{r}=1}^{K} \frac{\sum_{i \in \{i | \mathbf{R}_i = r\}} \mathrm{match}(\mathbf{R}_i, \mathbf{H}_i)}{\sum_{i \in \{i | \mathbf{R}_i = r\}} 1}, \quad (4)$$

	Criterion	random	ergodic0	ergodic1	ergodic2	concat1	concat2	supervised
Smoothness	MR	0.142_{e0e1}	0.111	0.111	0.157_{e0e1}	0.153	$\mathbf{0.199}_{e0e1r}$	$0.275_{c1e0e1e2r}$
	MAE	1.988_{e0e1}	2.212	2.212	1.980	1.936_{e0e1}	$\mathbf{1.870}_{e0e1}$	$1.420_{c1c2e0e1e2r}$
	KL	0.287	0.699	0.699	0.562	**0.280**	0.369	0.162
	MR/r	0.143	0.137	0.137	0.176	0.136	**0.177**	0.217
	MAE/r	2.286	2.414	2.414	**2.152**	2.301	2.206	1.782
Closeness	MR	0.143	0.129	0.129	0.171_{e0e1}	0.174	$\mathbf{0.189}_{e0e1}$	$0.279_{c1c2e0e1e2r}$
	MAE	2.028	2.066	2.066	1.964	$\mathbf{1.798}_{e0e1r}$	1.886	$1.431_{c1c2e0e1e2r}$
	KL	0.195	0.449	0.449	0.261	**0.138**	0.263	0.092
	MR/r	0.143	0.156	0.156	**0.170**	0.155	0.164	0.231
	MAE/r	2.283	2.236	2.236	2.221	2.079	**2.067**	1.702
Willingness	MR	0.143_{e0e1}	0.112	0.112	0.180_{e0e1}	0.152	$\mathbf{0.183}_{e0e1}$	$0.283_{c1c2e0e1e2r}$
	MAE	2.005	2.133	2.133	1.962	$\mathbf{1.801}_{e0e1r}$	1.882	$1.403_{c1c2e0e1e2r}$
	KL	**0.225**	0.568	0.568	0.507	0.238	0.255	0.125
	MR/r	0.143	0.152	0.152	**0.192**	0.181	0.167	0.224
	MAE/r	2.286	2.258	2.258	2.107	**1.958**	2.164	1.705

Table 1: The match rate (MR), mean absolute error (MAE), Kullback-Leibler divergence (KL), match rate per rating (MR/r) and mean absolute error per rating (MAE/r) for our proposed HMMs, the random baseline, and the upper bound (supervised) for the AD domain. 'e0–e2', 'c1–c2', and 'r' indicate the statistical significance ($p<0.01$) over ergodic0–2, concat1–2, and random, respectively. **Bold font** indicates the best value within each row (except for 'supervised').

where $\mathbf{R}_i$ and $\mathbf{H}_i$ denote ratings at i-th positions.

• **Mean Absolute Error per rating (MAE/r)**

$$\mathrm{MAE/r}(\mathbf{R},\mathbf{H}) = \frac{1}{K}\sum_{\mathrm{r}=1}^{K}\frac{\sum_{i\in\{i|\mathbf{R}_i=r\}}|\mathbf{R}_i-\mathbf{H}_i|}{\sum_{i\in\{i|\mathbf{R}_i=r\}}1}. \quad (5)$$

4.4 Evaluation Results

Tables 1 and 2 show the evaluation results for the AD and AL domains, respectively. The MR and MAE values are averaged over all dialogues. To compare the means of the MR and MAE, we performed a non-parametric multiple comparison test [Steel-Dwass test (Dwass, 1960)]. We did not perform a statistical test for other criteria because it was difficult to perform sample-wise comparison for distributions. Naturally, 'supervised' is the best performing model for all criteria in both domains. Therefore, we focus on how much our proposed models differ from the baseline (random) and the upper bound (supervised).

In the AD domain, we find that ergodic0 and ergodic1 performed rather poorly and concat1 and concat2 performed fairly well, significantly outperforming the random baseline. However, it is also clear that we still need a great deal of improvement for our models to reach the level of 'supervised'. A promising sign is that concat2 is not significantly different from 'supervised' in smoothness. Here, ergodic0 and ergodic1 returned the exact same results. This means that the state transition paths did not go through the common states at all in ergodic1, suggesting that the common states in ergodic1 have very broad output distributions and the optimal path could not go through the common states, instead preferring other states having sharper distributions. However, this phenomenon was rightly avoided by introducing more common states as seen in the results for ergodic2; nonetheless, as the results for concat1 and concat2 indicate, the transition probabilities have to be trained appropriately to obtain better results.

In the AL domain, although the tendency of the evaluation results is the same as that for the AD domain, concat2 is clearly the best performing model. It outperformed other models in almost all cases except for "Good Listener" for which concat1 performed better. In fact, the MR/r and MAE/r of concat1 are quite close to those of 'supervised', suggesting the potential of our approach.

Overall, although we still need further improvement in order for our models to be closer to the upper bound, we showed that we can, to some extent, predict user satisfaction transitions in a dialogue only from overall ratings of dialogues using our proposed HMMs. We also showed that model topologies and learning methods can make significant differences. Especially, we found the introduction of common states to be crucial in making appropriate models for prediction. Since our models, especially concat2, significantly outperformed the baseline, we believe that our approach can be one of the viable options for automatically predicting user satisfaction transitions when there exist only overall rating data.

5 Summary and Future Work

We presented a novel approach for modeling user satisfaction transitions only from dialogues with overall ratings. The experimental results show that it is possible to predict user satisfaction transi-

	Criterion	random	ergodic0	ergodic1	ergodic2	concat1	concat2	supervised
Smoothness	MR	0.143_{e0e1e2}	0.069	0.069	0.131_{e0e1}	0.173_{e0e1}	$\mathbf{0.243}_{c1e0e1e2r}$	$0.439_{c1c2e0e1e2r}$
	MAE	1.868_{e0e1e2}	2.519	2.519	2.433	$1.687_{e0e1e2r}$	$\mathbf{1.594}_{e0e1e2r}$	$0.802_{c1c2e0e1e2r}$
	KL	0.989	2.253	2.253	2.319	0.851	**0.753**	0.087
	MR/r	0.141	0.118	0.118	0.156	0.161	**0.167**	0.231
	MAE/r	2.289	2.500	2.500	2.492	2.093	**2.077**	1.868
Closeness	MR	0.143_{e0e1}	0.050	0.050	0.175_{e0e1}	0.158_{e0e1}	$\mathbf{0.263}_{c1e0e1e2r}$	$0.425_{c1c2e0e1e2r}$
	MAE	1.849_{e0e1e2}	2.357	2.357	2.316	1.778_{e0e1e2}	$\mathbf{1.562}_{e0e1e2r}$	$0.890_{c1c2e0e1e2r}$
	KL	1.022	2.137	2.137	2.220	1.155	**0.909**	0.109
	MR/r	0.143	0.090	0.090	0.122	0.117	**0.159**	0.237
	MAE/r	2.281	2.577	2.577	2.811	2.260	**2.039**	1.972
Good Listener	MR	0.143_{e0e1}	0.075	0.075	0.145_{e0e1}	0.199_{e0e1}	$\mathbf{0.206}_{e0e1e2}$	$0.422_{c1c2e0e1e2r}$
	MAE	1.890_{e0e1e2}	2.237	2.237	2.150	$\mathbf{1.634}_{e0e1e2r}$	$\mathbf{1.634}_{e0e1e2r}$	$0.852_{c1c2e0e1e2r}$
	KL	0.945	1.738	1.738	1.782	0.924	**0.824**	0.087
	MR/r	0.143	0.121	0.121	0.184	**0.224**	0.200	0.227
	MAE/r	2.284	2.358	2.358	2.236	**1.911**	2.083	1.769

Table 2: Evaluation results for the AL domain. See Table 1 for the notations in the table.

tions to some extent by our approach and that introducing common states and concatenated training can significantly improve prediction accuracy. For improvement, we plan to explore new dialogic features for emissions, different topologies, and other optimization functions, such as discriminative ones. We also need to validate our approach using dialogue-act recognition results instead of hand-labeled dialogue-acts. We also want to apply our approach to sequence mining in dialogues where we have categories instead of ratings for dialogues. It is also necessary to test whether our HMMs can be generalized over different raters, since user satisfaction ratings may differ greatly among individuals. Although there remain such issues, we believe we have presented a new direction in automatic evaluation of dialogues and the experimental results show that our approach is promising.

References

Meyer Dwass. 1960. Some k-sample rank-order tests. In Ingram Olkin et al., editor, *Contributions to Probability and Statistics*, pages 198–202. Stanford University Press.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller. 2009. Modeling user satisfaction with hidden Markov models. In *Proc. SIGDIAL*, pages 170–177.

Keelan Evanini, Phillip Hunter, Jackson Liscombe, David Suendermann, Krishna Dayanidhi, and Roberto Pieraccini. 2008. Caller experience: A method for evaluating dialog systems and its automatic prediction. In *Proc. SLT*, pages 129–132.

Allen L. Gorin, Giuseppe Riccardi, and Jerry H. Wright. 1997. How may I help you? *Speech Communication*, 23(1-2):113–127.

Sunao Hara, Norihide Kitaoka, and Kazuya Takeda. 2010. Estimation method of user satisfaction using N-gram-based dialog history model for spoken dialog system. In *Proc. LREC*, pages 78–83.

Ota Herm, Alexander Schmitt, and Jackson Liscombe. 2008. When calls go wrong: How to detect problematic calls based on log-files and emotions? In *Proc. INTERSPEECH*, pages 463–466.

Ryuichiro Higashinaka, Kohji Dohsaka, and Hideki Isozaki. 2008. Effects of self-disclosure and empathy in human-computer dialogue. In *Proc. SLT*, pages 109–112.

Julia Hirschberg, Diane Litman, and Marc Swerts. 2004. Prosodic and other cues to speech recognition failures. *Speech Communication*, 43:155–175.

Chiori Hori, Kiyonori Ohtake, Teruhisa Misu, Hideki Kashioka, and Satoshi Nakamura. 2008. Dialog management using weighted finite-state transducers. In *Proc. INTERSPEECH*, pages 211–214.

Naoki Isomura, Fujio Toriumi, and Kenichiro Ishii. 2009. Evaluation method of non-task-oriented dialogue system using HMM. *IEICE Transactions on Information and Systems*, J92-D(4):542–551.

Woosung Kim. 2007. Online call quality monitoring for automating agent-based call centers. In *Proc. INTERSPEECH*, pages 130–133.

Kai-Fu Lee. 1989. *Automatic speech recognition: the development of the SPHINX system*. Kluwer Academic Publishers.

Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, Yasuhiro Minami, and Hideki Isozaki. 2009. Analysis of listening-oriented dialogue for building listening agents. In *Proc. SIGDIAL*, pages 124–127.

Yasuhiro Minami, Akira Mori, Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, and Eisaku Maeda. 2009. Dialogue control algorithm for ambient intelligence based on partially observable Markov decision processes. In *Proc. IWSDS*, pages 254–263.

Sebastian Möller, Klaus-Peter Engelbrecht, and Robert Schleicher. 2008. Predicting the quality and usability of spoken dialogue services. *Speech Communication*, 50(8-9):730–744.

Lawrence R. Rabiner. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. *Readings in speech recognition*, 53(3):267–296.

Katsuhiko Shirai. 1996. Modeling of spoken dialogue with and without visual information. In *Proc. ICSLP*, volume 1, pages 188–191.

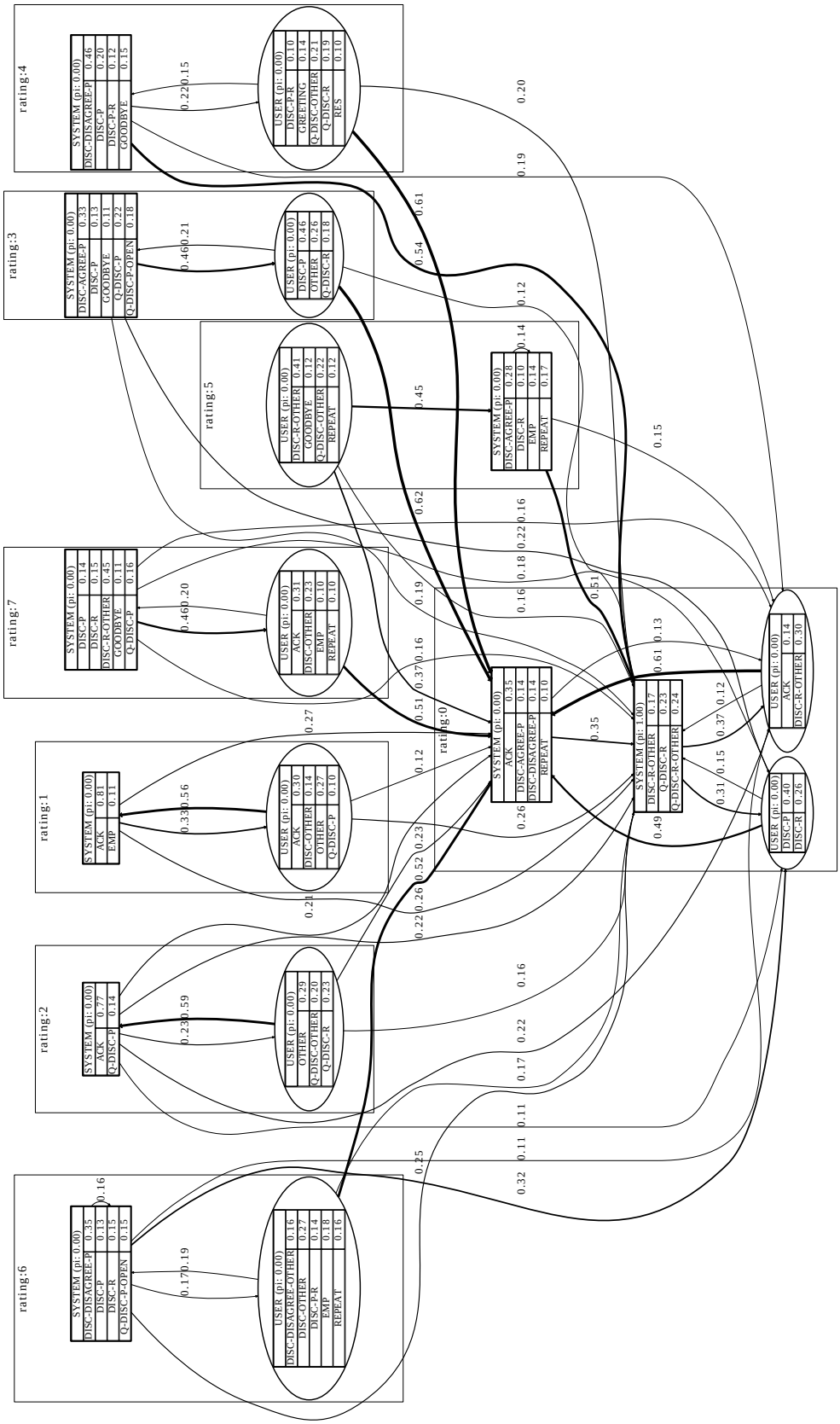
Marilyn A. Walker, Diane Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *Proc. EACL*, pages 271–280.

Marilyn A. Walker, Irene Langkilde-Geary, Helen Wright Hastie, Jerry Wright, and Allen Gorin. 2002. Automatically training a problematic dialogue predictor for a spoken dialogue system. *Journal of Artificial Intelligence Research*, 16(1):293–319.

Jason D. Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422.

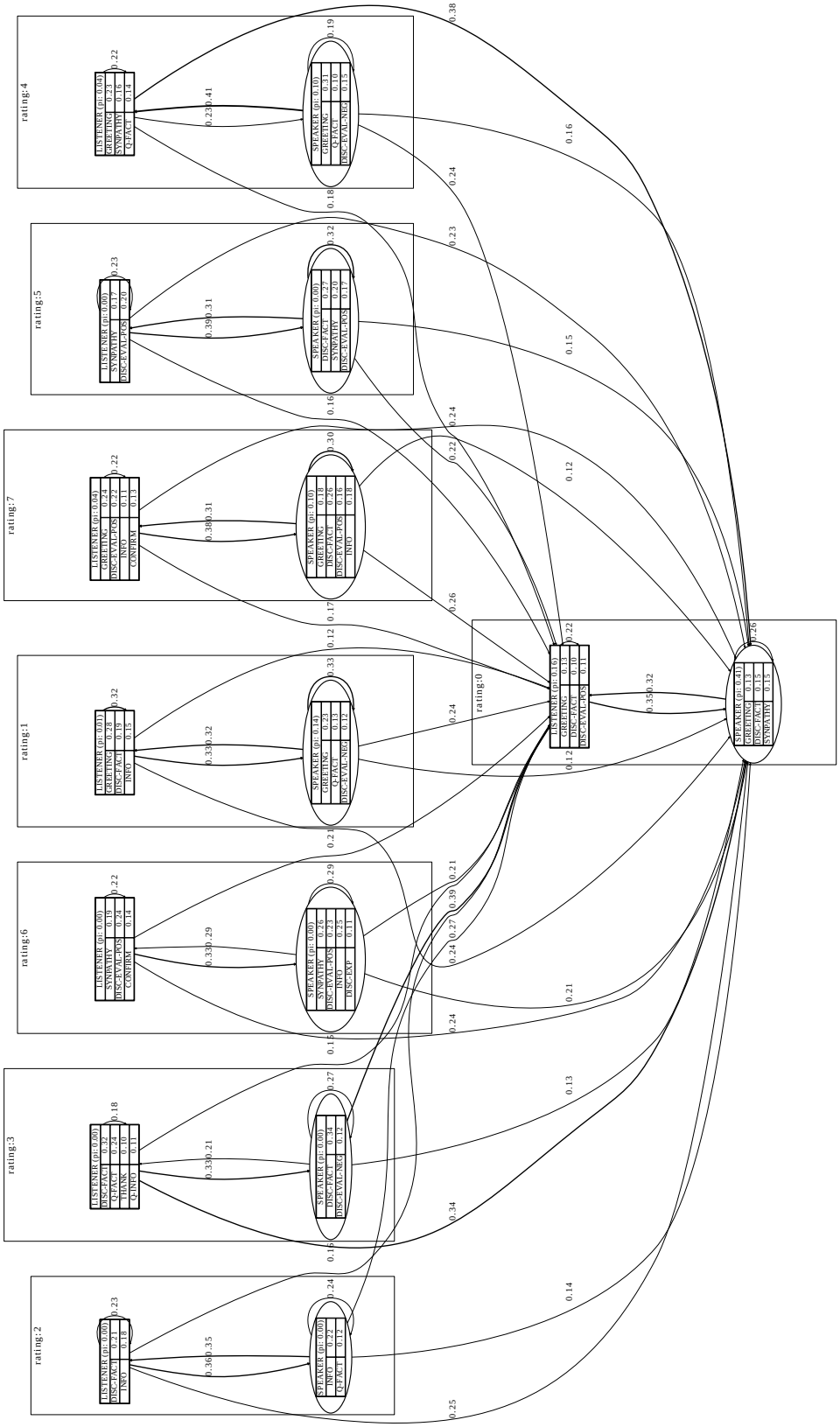
Appendix A. HMM obtained by concat2 for Willingness rating in the AD domain.

This HMM is the model obtained for one of the folds in the experiment. Square and oval states emit a system's dialogue-act and a user's dialogue-act, respectively. Emissions (dialogue-acts) are shown in each state as a table with their probabilities. Only the emissions and transitions over the probability of 0.1 are displayed for the sake of brevity. Here, 'pi' denotes initial probability.



Appendix B. HMM obtained by concat1 for Good Listener rating in the AL domain.

This HMM is the model obtained for one of the folds in the experiment. Square and oval states emit a listener's dialogue-act and a speaker's dialogue-act, respectively. We find DICS-EVAL-NEG (self-disclosure of one's evaluation with a negative polarity) in the rating score 1 and DICS-EVAL-POS in the rating score 7, indicating that it may be better to make speakers talk about positive evaluations to be a good listener.



Evaluation Metrics For End-to-End Coreference Resolution Systems

Jie Cai and **Michael Strube**
Heidelberg Institute for Theoretical Studies gGmbH
Schloß-Wolfsbrunnenweg 35
69118 Heidelberg, Germany
(jie.cai|michael.strube)@h-its.org

Abstract

Commonly used coreference resolution evaluation metrics can only be applied to key mentions, i.e. already annotated mentions. We here propose two variants of the B^3 and *CEAF* coreference resolution evaluation algorithms which can be applied to coreference resolution systems dealing with system mentions, i.e. automatically determined mentions. Our experiments show that our variants lead to intuitive and reliable results.

1 Introduction

The coreference resolution problem can be divided into two steps: (1) determining *mentions*, i.e., whether an expression is referential and can take part in a coreferential relationship, and (2) deciding whether mentions are coreferent or not. Most recent research on coreference resolution simplifies the resolution task by providing the system with *key mentions*, i.e. already annotated mentions (Luo et al. (2004), Denis & Baldridge (2007), Culotta et al. (2007), Haghighi & Klein (2007), inter alia; see also the task description of the recent SemEval task on coreference resolution at `http://stel.ub.edu/semeval2010-coref`), or ignores an important part of the problem by evaluating on key mentions only (Ponzetto & Strube, 2006; Bengtson & Roth, 2008, inter alia). We follow here Stoyanov et al. (2009, p.657) in arguing that such evaluations are "an unrealistic surrogate for the original problem" and ask researchers to evaluate end-to-end coreference resolution systems.

However, the evaluation of end-to-end coreference resolution systems has been inconsistent making it impossible to compare the results. Nicolae & Nicolae (2006) evaluate using the *MUC* score (Vilain et al., 1995) and the *CEAF* algorithm (Luo, 2005) without modifications. Yang et al. (2008) use only the *MUC* score. Bengtson & Roth (2008) and Stoyanov et al. (2009) derive variants from the B^3 algorithm (Bagga & Baldwin, 1998). Rahman & Ng (2009) propose their own variants of B^3 and *CEAF*. Unfortunately, some of the metrics' descriptions are so concise that they leave too much room for interpretation. Also, some of the metrics proposed are too lenient or are more sensitive to mention detection than to coreference resolution. Hence, though standard corpora are used, the results are not comparable.

This paper attempts to fill that desideratum by analysing several variants of the B^3 and *CEAF* algorithms. We propose two new variants, namely B^3_{sys} and $CEAF_{sys}$, and provide algorithmic details in Section 2. We describe two experiments in Section 3 showing that B^3_{sys} and $CEAF_{sys}$ lead to intuitive and reliable results. Implementations of B^3_{sys} and $CEAF_{sys}$ are available open source along with extended examples[1].

2 Coreference Evaluation Metrics

We discuss the problems which arise when applying the most prevalent coreference resolution evaluation metrics to end-to-end systems and propose our variants which overcome those problems. We provide detailed analyses of illustrative examples.

2.1 *MUC*

The MUC score (Vilain et al., 1995) counts the minimum number of links between mentions to be inserted or deleted when mapping a system response to a gold standard key set. Although pairwise links capture the information in a set, they cannot represent singleton entities, i.e. entities, which are mentioned only once. Therefore, the MUC score is not suitable for the ACE data (`http://www.itl.nist.`

[1] `http://www.h-its.org/nlp/download`

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 28–36,
The University of Tokyo, September 24-25, 2010.

gov/iad/mig/tests/ace/), which includes singleton entities in the keys. Moreover, the MUC score does not give credit for separating singleton entities from other chains. This becomes problematic in a realistic system setup, when mentions are extracted automatically.

2.2 B^3

The B^3 algorithm (Bagga & Baldwin, 1998) overcomes the shortcomings of the MUC score. Instead of looking at the links, B^3 computes precision and recall for all mentions in the document, which are then combined to produce the final precision and recall numbers for the entire output.

For each mention, the B^3 algorithm computes a precision and recall score using equations 1 and 2:

$$Precision(m_i) = \frac{|R_{m_i} \cap K_{m_i}|}{|R_{m_i}|} \quad (1)$$

$$Recall(m_i) = \frac{|R_{m_i} \cap K_{m_i}|}{|K_{m_i}|} \quad (2)$$

where R_{m_i} is the response chain (i.e. the system output) which includes the mention m_i, and K_{m_i} is the key chain (manually annotated gold standard) with m_i. The overall precision and recall are computed by averaging them over all mentions.

Since B^3's calculations are based on mentions, singletons are taken into account. However, a problematic issue arises when system mentions have to be dealt with: B^3 assumes the mentions in the key and in the response to be identical. Hence, B^3 has to be extended to deal with system mentions which are not in the key and key mentions not extracted by the system, so called *twinless mentions* (Stoyanov et al., 2009).

2.2.1 Existing B^3 variants

A few variants of the B^3 algorithm for dealing with system mentions have been introduced recently. Stoyanov et al. (2009) suggest two variants of the B^3 algorithm to deal with system mentions, B^3_0 and B^3_{all}[2]. For example, a key and a response are provided as below:

Key : {a b c}
Response: {a b d}

B^3_0 discards all twinless system mentions (i.e. mention d) and penalizes recall by setting $recall_{m_i} = 0$ for all twinless key mentions (i.e. mention c). The B^3_0 precision, recall and F-score

[2]Our discussion of B^3_0 and B^3_{all} is based on the analysis of the source code available at http://www.cs.utah.edu/nlp/reconcile/.

		Set 1		
System 1	key response	{a b c} {a b d}		
		P	R	F
B^3_0		1.0	0.444	0.615
B^3_{all}		0.556	0.556	0.556
$B^3_{r\&n}$		0.556	0.556	0.556
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572
System 2	key response	{a b c} {a b d e}		
		P	R	F
B^3_0		1.0	0.444	0.615
B^3_{all}		0.375	0.556	0.448
$B^3_{r\&n}$		0.375	0.556	0.448
B^3_{sys}		0.5	0.556	0.527
$CEAF_{sys}$		0.4	0.667	0.500

Table 1: Problems of B^3_0

(i.e. $F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$) for the example are calculated as:

$Pr_{B^3_0} = \frac{1}{2}(\frac{2}{2} + \frac{2}{2}) = 1.0$
$Rec_{B^3_0} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + 0) \doteq 0.444$
$F_{B^3_0} = 2 \times \frac{1.0 \times 0.444}{1.0 + 0.444} \doteq 0.615$

B^3_{all} retains twinless system mentions. It assigns $1/|R_{m_i}|$ to a twinless system mention as its precision and similarly $1/|K_{m_i}|$ to a twinless key mention as its recall. For the same example above, the B^3_{all} precision, recall and F-score are given by:

$Pr_{B^3_{all}} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3}) \doteq 0.556$
$Rec_{B^3_{all}} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3}) \doteq 0.556$
$F_{B^3_{all}} = 2 \times \frac{0.556 \times 0.556}{0.556 + 0.444} \doteq 0.556$

Tables 1, 2 and 3 illustrate the problems with B^3_0 and B^3_{all}. The rows labeled *System* give the original keys and system responses while the rows labeled B^3_0, B^3_{all} and B^3_{sys} show the performance generated by Stoyanov et al.'s variants and the one we introduce in this paper, B^3_{sys} (the row labeled $CEAF_{sys}$ is discussed in Subsection 2.3).

In Table 1, there are two system outputs (i.e. *System 1* and *System 2*). Mentions *d* and *e* are the twinless system mentions erroneously resolved and *c* a twinless key mention. *System 1* is supposed to be slightly better with respect to precision, because *System 2* produces one more spurious resolution (i.e. for mention *e*). However, B^3_0 computes exactly the same numbers for both systems. Hence, there is no penalty for erroneous coreference relations in B^3_0, if the mentions do not appear in the key, e.g. putting mentions *d* or *e* in *Set 1* does not count as precision errors. — B^3_0 is too lenient by only evaluating the correctly extacted mentions.

		Set 1	Singletons	
System 1	key response	{a b c} {a b d}		
		P	R	F
B^3_{all}		0.556	0.556	0.556
$B^3_{r\&n}$		0.556	0.556	0.556
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572
System 2	key response	{a b c} {a b d}	{c}	
		P	R	F
B^3_{all}		0.667	0.556	0.606
$B^3_{r\&n}$		0.667	0.556	0.606
B^3_{sys}		0.667	0.556	0.606
$CEAF_{sys}$		0.5	0.667	0.572

Table 2: Problems of B^3_{all} (1)

		Set 1	Singletons	
System 1	key response	{a b} {a b d}		
		P	R	F
B^3_{all}		0.556	1.0	0.715
$B^3_{r\&n}$		0.556	1.0	0.715
B^3_{sys}		0.556	1.0	0.715
$CEAF_{sys}$		0.667	1.0	0.800
System 2	key response	{a b} {a b d}	{i} {j} {k}	
		P	R	F
B^3_{all}		0.778	1.0	0.875
$B^3_{r\&n}$		0.556	1.0	0.715
B^3_{sys}		0.556	1.0	0.715
$CEAF_{sys}$		0.667	1.0	0.800

Table 3: Problems of B^3_{all} (2)

B^3_{all} deals well with the problem illustrated in Table 1, the figures reported correspond to intuition. However, B^3_{all} can output different results for identical coreference resolutions when exposed to different mention taggers as shown in Tables 2 and 3. B^3_{all} manages to penalize erroneous resolutions for twinless system mentions, however, it ignores twinless key mentions when measuring precision. In Table 2, *System 1* and *System 2* generate the same outputs, except that the mention tagger in *System 2* also extracts mention *c*. Intuitively, the same numbers are expected for both systems. However, B^3_{all} gives a higher precision to *System 2*, which results in a higher F-score.

B^3_{all} retains all twinless system mentions, as can be seen in Table 3. *System 2*'s mention tagger tags more mentions (i.e. the mentions *i, j* and *k*), while both *System 1* and *System 2* have identical coreference resolution performance. Still, B^3_{all} outputs quite different results for precision and thus for F-score. This is due to the credit B^3_{all} takes from unresolved singleton twinless system mentions (i.e. mention *i, j, k* in *System 2*). Since the metric is expected to evaluate the end-to-end coreference system performance rather than the mention tagging quality, it is not satisfying to observe that B^3_{all}'s numbers actually fluctuate when the system is exposed to different mention taggers.

Rahman & Ng (2009) apply another variant, denoted here as $B^3_{r\&n}$. They remove only those twinless system mentions that are singletons before applying the B^3 algorithm. So, a system would not be rewarded by the the spurious mentions which are correctly identified as singletons during resolution (as has been the case with B^3_{all}'s higher precision for *System 2* as can be seen in Table 3).

We assume that Rahman & Ng apply a strategy similar to B^3_{all} after the removing step (this is not clear in Rahman & Ng (2009)). While it avoids the problem with singleton twinless system mentions, $B^3_{r\&n}$ still suffers from the problem dealing with twinless key mentions, as illustrated in Table 2.

2.2.2 B^3_{sys}

We here propose a coreference resolution evaluation metric, B^3_{sys}, which deals with system mentions more adequately (see the rows labeled B^3_{sys} in Tables 1, 2, 3, 4 and 5). We put all twinless key mentions into the response as singletons which enables B^3_{sys} to penalize non-resolved coreferent key mentions without penalizing non-resolved singleton key mentions, and also avoids the problem B^3_{all} and $B^3_{r\&n}$ have as shown in Table 2. All twinless system mentions which were deemed not coreferent (hence being singletons) are discarded. To calculate B^3_{sys} precision, all twinless system mentions which were mistakenly resolved are put into the key since they are spurious resolutions (equivalent to the assignment operations in B^3_{all}), which should be penalized by precision. Unlike B^3_{all}, B^3_{sys} does not benefit from unresolved twinless system mentions (i.e. the twinless singleton system mentions). For recall, the algorithm only goes through the original key sets, similar to B^3_{all} and $B^3_{r\&n}$. Details are given in Algorithm 1.

For example, a coreference resolution system has the following key and response:

Key : {a b c}
Response: {a b d} {i j}

To calculate the precision of B^3_{sys}, the key and response are altered to:

Key_p : {a b c} {d} {i} {j}
Response_p: {a b d} {i j} {c}

Algorithm 1 B^3_{sys}

Input: key sets key, response sets $response$
Output: precision P, recall R and F-score F

```
Discard all the singleton twinless system mentions in response;
Put all the twinless annotated mentions into response;
if calculating precision then
    Merge all the remaining twinless system mentions with key to form key_p;
    Use response to form response_p
    Through key_p and response_p;
    Calculate B^3 precision P.
end if
if calculating recall then
    Discard all the remaining twinless system mentions in response to from response_r;
    Use key to form key_r
    Through key_r and response_r;
    Calculate B^3 recall R
end if
Calculate F-score F
```

So, the precision of B^3_{sys} is given by:

$$Pr_{B^3_{sys}} = \frac{1}{6}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{2} + \frac{1}{2} + 1) \doteq 0.611$$

The modified key and response for recall are:

Key$_r$: {a b c}
Response$_r$: {a b} {c}

The resulting recall of B^3_{sys} is:

$$Rec_{B^3_{sys}} = \frac{1}{3}(\frac{2}{3} + \frac{2}{3} + \frac{1}{3}) \doteq 0.556$$

Thus the F-score number is calculated as:

$$F_{B^3_{sys}} = 2 \times \frac{0.611 \times 0.556}{0.611 + 0.556} \doteq 0.582$$

B^3_{sys} indicates more adequately the performance of end-to-end coreference resolution systems. It is not easily tricked by different mention taggers[3].

2.3 *CEAF*

Luo (2005) criticizes the B^3 algorithm for using entities more than one time, because B^3 computes precision and recall of mentions by comparing entities containing that mention. Hence Luo proposes the *CEAF* algorithm which aligns entities in key and response. *CEAF* applies a similarity metric (which could be either mention based or entity based) for each pair of entities (i.e. a set of mentions) to measure the goodness of each possible alignment. The best mapping is used for calculating *CEAF* precision, recall and F-measure.

Luo proposes two entity based similarity metrics (Equation 3 and 4) for an entity pair (K_i, R_j) originating from key, K_i, and response, R_j.

$$\phi_3(K_i, R_j) = |K_i \cap R_j| \quad (3)$$

$$\phi_4(K_i, R_j) = \frac{2|K_i \cap R_j|}{|K_i| + |R_j|} \quad (4)$$

[3]Further example analyses can be found in Appendix A.

The *CEAF* precision and recall are derived from the alignment which has the best total similarity (denoted as $\Phi(g^*)$), shown in Equations 5 and 6.

$$Precision = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \quad (5)$$

$$Recall = \frac{\Phi(g^*)}{\sum_i \phi(K_i, K_i)} \quad (6)$$

If not specified otherwise, we apply Luo's $\phi_3(\star, \star)$ in the example illustrations. We denote the original *CEAF* algorithm as *CEAF*$_{orig}$.

Detailed calculations are illustrated below:

Key : {a b c}
Response: {a b d}

The *CEAF*$_{orig}$ $\phi_3(\star, \star)$ are given by:

$\phi_3(K_1, R_1) = 2$ $(K_1 : \{abc\}; R_1 : \{abd\})$
$\phi_3(K_1, K_1) = 3$
$\phi_3(R_1, R_1) = 3$

So the *CEAF*$_{orig}$ evaluation numbers are:

$Pr_{CEAF_{orig}} = \frac{2}{3} = 0.667$
$Rec_{CEAF_{orig}} = \frac{2}{3} = 0.667$
$F_{CEAF_{orig}} = 2 \times \frac{0.667 \times 0.667}{0.667 + 0.667} = 0.667$

2.3.1 Problems of *CEAF*$_{orig}$

CEAF$_{orig}$ was intended to deal with key mentions. Its adaptation to system mentions has not been addressed explicitly. Although *CEAF*$_{orig}$ theoretically does not require to have the same number of mentions in key and response, it still cannot be directly applied to end-to-end systems, because the entity alignments are based on mention mappings.

As can be seen from Table 4, *CEAF*$_{orig}$ fails to produce intuitive results for system mentions. *System 2* outputs one more spurious entity (containing mention *i* and *j*) than *System 1* does, however, achieves a same *CEAF*$_{orig}$ precision. Since twinless system mentions do not have mappings in key, they contribute nothing to the mapping similarity. So, resolution mistakes for system mentions are not calculated, and moreover, the precision is easily skewed by the number of output entities. *CEAF*$_{orig}$ reports very low precision for system mentions (see also Stoyanov et al. (2009)).

2.3.2 Existing *CEAF* variants

Rahman & Ng (2009) briefly introduce their *CEAF* variant, which is denoted as *CEAF*$_{r\&n}$ here. They use $\phi_3(\star, \star)$, which results in equal *CEAF*$_{r\&n}$ precision and recall figures when using true mentions. Since Rahman & Ng's experiments using system mentions produce unequal precision and recall figures, we assume that, after removing

		Set 1	Set 2	Singletons
System 1	key	{a b c}		
	response	{a b}		{c} {i} {j}
		P	R	F
$CEAF_{orig}$		0.4	0.667	0.500
B^3_{sys}		1.0	0.556	0.715
$CEAF_{sys}$		0.667	0.667	0.667
System 2	key	{a b c}		
	response	{a b}	{i j}	{c}
		P	R	F
$CEAF_{orig}$		0.4	0.667	0.500
B^3_{sys}		0.8	0.556	0.656
$CEAF_{sys}$		0.6	0.667	0.632

Table 4: Problems of $CEAF_{orig}$

		Set 1	Set 2	Set 3	Singletons
System 1	key	{a b c}			
	response	{a b}	{i j}	{k l}	{c}
		P	R	F	
$CEAF_{r\&n}$		0.286	0.667	0.400	
B^3_{sys}		0.714	0.556	0.625	
$CEAF_{sys}$		0.571	0.667	0.615	
System 2	key	{a b c}			
	response	{a b}	{i j k l}		{c}
		P	R	F	
$CEAF_{r\&n}$		0.286	0.667	0.400	
B^3_{sys}		0.571	0.556	0.563	
$CEAF_{sys}$		0.429	0.667	0.522	

Table 5: Problems of $\text{CEAF}_{r\&n}$

twinless singleton system mentions, they do not put any twinless mentions into the other set. In the example in Table 5, $CEAF_{r\&n}$ does not penalize adequately the incorrectly resolved entities consisting of twinless sytem mentions. So $CEAF_{r\&n}$ does not tell the difference between *System 1* and *System 2*. It can be concluded from the examples that the same number of mentions in key and response is needed for computing the *CEAF* score.

2.3.3 $CEAF_{sys}$

We propose to adjust *CEAF* in the same way as we did for B^3_{sys}, resulting in $CEAF_{sys}$. We put all twinless key mentions into the response as singletons. All singleton twinless system mentions are discarded. For calculating $CEAF_{sys}$ precision, all twinless system mentions which were mistakenly resolved are put into the key. For computing $CEAF_{sys}$ recall, only the original key sets are considered. That way $CEAF_{sys}$ deals adequately with system mentions (see Algorithm 2 for details).

Algorithm 2 $CEAF_{sys}$

Input: key sets key, response sets $response$
Output: precision P, recall R and F-score F

1: Discard all the singleton twinless system mentions in $response$;
2: Put all the twinless annotated mentions into $response$;
3: **if** calculating precision **then**
4: Merge all the remaining twinless system mentions with key to form key_p;
5: Use $response$ to form $response_p$
6: Form Map $g^\star$ between key_p and $response_p$
7: Calculate $CEAF$ precision P using $\phi_3(\star,\star)$
8: **end if**
9: **if** calculating recall **then**
10: Discard all the remaining twinless system mentions in $response$ to form $response_r$;
11: Use key to form key_r
12: Form Map $g^\star$ between key_r and $response_r$
13: Calculate $CEAF$ recall R using $\phi_3(\star,\star)$
14: **end if**
15: Calculate F-score F

Taking *System 2* in Table 4 as an example, key and response are altered for precision:

Key$_p$: {a b c} {i} {j}
Response$_p$: {a b d} {i j} {c}

So the $\phi_3(\star,\star)$ are as below, only listing the best mappings:

$\phi_3(K_1, R_1) = 2\ (K_1 : \{abc\}; R_1 : \{abd\})$
$\phi_3(K_2, R_2) = 1\ (K_2 : \{i\}; R_2 : \{ij\})$
$\phi_3(\emptyset, R_3) = 0\ (R_3 : \{c\})$
$\phi_3(R_1, R_1) = 3$
$\phi_3(R_2, R_2) = 2$
$\phi_3(R_3, R_3) = 1$

The precision is thus give by:

$Pr_{CEAF_{sys}} = \frac{2+1+0}{3+2+1} = 0.6$

The key and response for recall are:

Key$_r$: {a b c}
Response$_r$: {a b} {c}

The resulting $\phi_3(\star,\star)$ are:

$\phi_3(K_1, R_1) = 2(K_1 : \{abc\}; R_1 : \{ab\})$
$\phi_3(\emptyset, R_2) = 0(R_2 : \{c\})$
$\phi_3(K_1, K_1) = 3$
$\phi_3(R_1, R_1) = 2$
$\phi_3(R_2, R_2) = 1$

The recall and F-score are thus calculated as:

$Rec_{CEAF_{sys}} = \frac{2}{3} = 0.667$
$F_{CEAF_{sys}} = 2 \times \frac{0.6 \times 0.667}{0.6+0.667} = 0.632$

However, one additional complication arises with regard to the similarity metrics used by *CEAF*. It turns out that only $\phi_3(\star,\star)$ is suitable for dealing with system mentions while $\phi_4(\star,\star)$ produces uninituitive results (see Table 6).

$\phi_4(\star,\star)$ computes a normalized similarity for each entity pair using the summed number of mentions in the key and the response. *CEAF* precision then distributes that similarity evenly over the response set. Spurious system entities, such as the one with mention *i* and *j* in Table 6, are not penalized. $\phi_3(\star,\star)$ calculates unnormalized similarities. It compares the two systems in Table 6 adequately. Hence we use only $\phi_3(\star,\star)$ in $CEAF_{sys}$.

		Set 1	Singletons	
System 1	key	{a b c}		
	response	{a b}	{c} {i} {j}	
		P	R	F
$\phi_4(\star,\star)$		0.4	0.8	0.533
$\phi_3(\star,\star)$		0.667	0.667	0.667
System 2	key	{a b c}		
	response	{a b} {i j}	{c}	
		P	R	F
$\phi_4(\star,\star)$		0.489	0.8	0.607
$\phi_3(\star,\star)$		0.6	0.667	0.632

Table 6: Problems of $\phi_4(\star,\star)$

When normalizing the similarities by the number of entities or mentions in the key (for recall) and the response (for precision), the *CEAF* algorithm considers all entities or mentions to be equally important. Hence *CEAF* tends to compute quite low precision for system mentions which does not represent the system performance adequately. Here, we do not address this issue.

2.4 *BLANC*

Recently, a new coreference resolution evaluation algorithm, *BLANC*, has been introduced (Recasens & Hovy, 2010). This measure implements the *Rand index* (Rand, 1971) which has been originally developed to evaluate clustering methods. The *BLANC* algorithm deals correctly with singleton entities and rewards correct entities according to the number of mentions. However, a basic assumption behind *BLANC* is, that the sum of all coreferential and non-coreferential links is constant for a given set of mentions. This implies that *BLANC* assumes identical mentions in key and response. It is not clear how to adapt *BLANC* to system mentions. We do not address this issue here.

3 Experiments

While Section 2 used toy examples to motivate our metrics B^3_{sys} and *CEAF*$_{sys}$, we here report results on two larger experiments using ACE2004 data.

3.1 Data and Mention Taggers

We use the ACE2004 (Mitchell et al., 2004) English training data which we split into three sets following Bengtson & Roth (2008): Train (268 docs), Dev (76), and Test (107). We use two in-house mention taggers. The first (*SM1*) implements a heuristic aiming at high recall. The second (*SM2*) uses the *J48* decision tree classifier (Witten & Frank, 2005). The number of detected mentions, head coverage, and accuracy on testing data are shown in Table 7.

		SM1	*SM2*
training	mentions	31,370	16,081
	twin mentions	13,072	14,179
development	mentions	8,045	–
	twin mentions	3,371	–
test	mentions	8,387	4,956
	twin mentions	4,242	4,212
	head coverage	79.3%	73.3%
	accuracy	57.3%	81.2%

Table 7: Mention Taggers on ACE2004 Data

3.2 Artificial Setting

For the artificial setting we report results on the development data using the *SM1* tagger. To illustrate the stability of the evaluation metrics with respect to different mention taggers, we reduce the number of twinless system mentions in intervals of 10%, while correct (non-twinless) ones are kept untouched. The coreference resolution system used is the BART (Versley et al., 2008) reimplementation of Soon et al. (2001). The results are plotted in Figures 1 and 2.

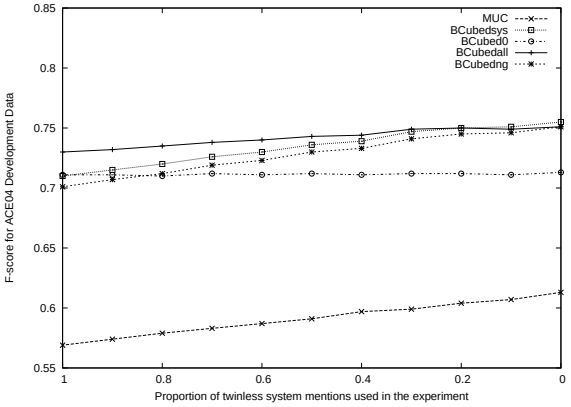


Figure 1: Artificial Setting B^3 Variants

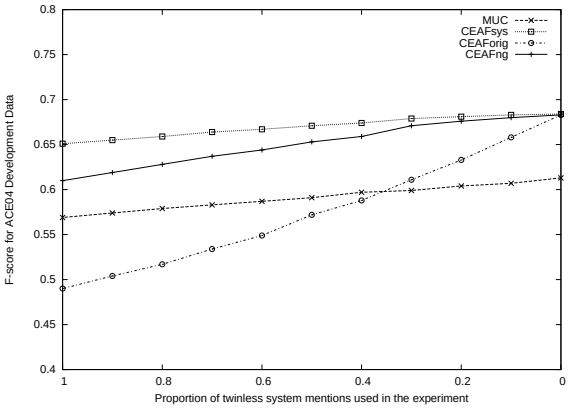


Figure 2: Artificial Setting *CEAF* Variants

	MUC		
	R	Pr	F
Soon (SM1)	51.7	53.1	52.4
Soon (SM2)	49.1	69.9	**57.7**

Table 8: Realistic Setting *MUC*

	B^3_{sys}			B^3_0		
	R	Pr	F	R	Pr	F
Soon (SM2)	64.1	87.3	73.9	54.7	91.3	68.4
Bengtson	66.1	81.9	73.1	69.5	74.7	72.0

Table 11: Realistic Setting

Omitting twinless system mentions from the training data while keeping the number of correct mentions constant should improve the coreference resolution performance, because a more precise coreference resolution model is obtained. As can be seen from Figures 1 and 2, the *MUC*-score, B^3_{sys} and *CEAF*$_{sys}$ follow this intuition.

B^3_0 is almost constant. It does not take twinless mentions into account. B^3_{all}'s curve, also, has a lower slope in comparison to B^3_{sys} and MUC (i.e. B^3_{all} computes similar numbers for worse models). This shows that the B^3_{all} score can be tricked by using a high recall mention tagger, e.g. in cases with the worse models (i.e. ones on the left side of the figures) which have much more twinless system mentions. The original *CEAF* algorithm, *CEAF*$_{orig}$, is too sensitive to the input system mentions making it less reliable. *CEAF*$_{sys}$ is parallel to B^3_{sys}. Thus both of our metrics exhibit the same intuition.

3.3 Realistic Setting

3.3.1 Experiment 1

For the realistic setting we compare *SM1* and *SM2* as preprocessing components for the BART (Versley et al., 2008) reimplementation of Soon et al. (2001). The coreference resolution system with the *SM2* tagger performs better, because a better coreference model is achieved from system mentions with higher accuracy.

The *MUC*, B^3_{sys} and *CEAF*$_{sys}$ metrics have the same tendency when applied to systems with different mention taggers (Table 8, 9 and 10 and the bold numbers are higher with a p-value of 0.05, by a paired-t test). Since the *MUC* scorer does not evaluate singleton entities, it produces too low numbers which are not informative any more.

As shown in Table 9, B^3_{all} reports counter-intuitive results when a system is fed with system mentions generated by different mention taggers. B^3_{all} cannot be used to evaluate two different end-to-end coreference resolution systems, because the mention tagger is likely to have bigger impact than the coreference resolution system. B^3_0 fails to generate the right comparison too, because it is too lenient by ignoring all twinless mentions.

The *CEAF*$_{orig}$ numbers in Table 10 illustrate the big influence the system mentions have on precision (e.g. the very low precision number for *Soon (SM1)*). The big improvement for *Soon (SM2)* is largely due to the system mentions it uses, rather than to different coreference models.

Both $B^3_{r\&n}$ and *CEAF*$_{r\&n}$ show no serious problems in the experimental results. However, as discussed before, they fail to penalize the spurious entities with twinless system mentions adequately.

3.3.2 Experiment 2

We compare results of Bengtson & Roth's (2008) system with our *Soon (SM2)* system. Bengtson & Roth's embedded mention tagger aims at high precision, generating half of the mentions *SM1* generates (explicit statistics are not available to us).

Bengtson & Roth report a B^3 F-score for system mentions, which is very close to the one for true mentions. Their B^3-variant does not impute errors of twinless mentions and is assumed to be quite similar to the B^3_0 strategy.

We integrate both the B^3_0 and B^3_{sys} variants into their system and show results in Table 11 (we cannot report significance, because we do not have access to results for single documents in Bengtson & Roth's system). It can be seen that, when different variants of evaluation metrics are applied, the performance of the systems vary wildly.

4 Conclusions

In this paper, we address problems of commonly used evaluation metrics for coreference resolution and suggest two variants for B^3 and *CEAF*, called B^3_{sys} and *CEAF*$_{sys}$. In contrast to the variants proposed by Stoyanov et al. (2009), B^3_{sys} and *CEAF*$_{sys}$ are able to deal with end-to-end systems which do not use any gold information. The numbers produced by B^3_{sys} and $CEAF_{sys}$ are able to indicate the resolution performance of a system more adequately, without being tricked easily by twisting preprocessing components. We believe that the explicit description of evaluation metrics, as given in this paper, is a precondition for the re-

	B^3_{sys}			B^3_0			B^3_{all}			$B^3_{r\&n}$		
	R	Pr	F	R	Pr	F	R	Pr	F	R	Pr	F
Soon (SM1)	65.7	76.8	70.8	57.0	91.1	**70.1**	65.1	85.8	74.0	65.1	78.7	71.2
Soon (SM2)	64.1	87.3	**73.9**	54.7	91.3	68.4	64.3	87.1	73.9	64.3	84.9	**73.2**

Table 9: Realistic Setting B^3 Variants

	CEAF$_{sys}$			*CEAF*$_{orig}$			*CEAF*$_{r\&n}$		
	R	Pr	F	R	Pr	F	R	Pr	F
Soon (SM1)	66.4	61.2	63.7	62.0	39.9	48.5	62.1	59.8	60.9
Soon (SM2)	67.4	65.2	**66.3**	60.0	56.6	**58.2**	60.0	66.2	62.9

Table 10: Realistic Setting *CEAF* Variants

liabe comparison of end-to-end coreference resolution systems.

Acknowledgements. This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a HITS PhD. scholarship. We would like to thank Éva Mújdricza-Maydt for implementing the mention taggers.

References

Bagga, Amit & Breck Baldwin (1998). Algorithms for scoring coreference chains. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, Granada, Spain, 28–30 May 1998, pp. 563–566.

Bengtson, Eric & Dan Roth (2008). Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Waikiki, Honolulu, Hawaii, 25-27 October 2008, pp. 294–303.

Culotta, Aron, Michael Wick & Andrew McCallum (2007). First-order probabilistic models for coreference resolution. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, N.Y., 22–27 April 2007, pp. 81–88.

Denis, Pascal & Jason Baldridge (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, N.Y., 22–27 April 2007, pp. 236–243.

Haghighi, Aria & Dan Klein (2007). Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 23–30 June 2007, pp. 848–855.

Luo, Xiaoqiang (2005). On coreference resolution performance metrics. In *Proceedings of the Human Language Technology Conference and the 2005 Conference on Empirical Methods in Natural Language Processing*, Vancouver, B.C., Canada, 6–8 October 2005, pp. 25–32.

Luo, Xiaoqiang, Abe Ittycheriah, Hongyan Jing, Nanda Kambhatla & Salim Roukos (2004). A mention-synchronous coreference resolution algorithm based on the Bell Tree. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain, 21–26 July 2004, pp. 136–143.

Mitchell, Alexis, Stephanie Strassel, Shudong Huang & Ramez Zakhary (2004). *ACE 2004 Multilingual Training Corpus*. LDC2005T09, Philadelphia, Penn.: Linguistic Data Consortium.

Nicolae, Cristina & Gabriel Nicolae (2006). BestCut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 22–23 July 2006, pp. 275–283.

Ponzetto, Simone Paolo & Michael Strube (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, New York, N.Y., 4–9 June 2006, pp. 192–199.

Rahman, Altaf & Vincent Ng (2009). Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore, 6-7 August 2009, pp. 968–977.

Rand, William R. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Recasens, Marta & Eduard Hovy (2010). *BLANC: Implementing the Rand index for coreference evaluation*. Submitted.

Soon, Wee Meng, Hwee Tou Ng & Daniel Chung Yong Lim (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544.

Stoyanov, Veselin, Nathan Gilbert, Claire Cardie & Ellen Riloff (2009). Conundrums in noun phrase coreference resolution: Making sense of the state-of-the-art. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing*, Singapore, 2–7 August 2009, pp. 656–664.

Versley, Yannick, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang & Alessandro Moschitti (2008). BART: A modular toolkit for coreference resolution. In *Companion Volume to the Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, Columbus, Ohio, 15–20 June 2008, pp. 9–12.

Vilain, Marc, John Burger, John Aberdeen, Dennis Connolly & Lynette Hirschman (1995). A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference (MUC-6)*, pp. 45–52. San Mateo, Cal.: Morgan Kaufmann.

Witten, Ian H. & Eibe Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques* (2nd ed.). San Francisco, Cal.: Morgan Kaufmann.

Yang, Xiaofeng, Jian Su & Chew Lim Tan (2008). A twin-candidate model for learning-based anaphora resolution. *Computational Linguistics*, 34(3):327–356.

A B^3_{sys} Example Output

Here, we provide additional examples for analyzing the behavior of B^3_{sys} where we systematically vary system outputs. Since we proposed B^3_{sys} for dealing with end-to-end systems, we consider only examples also containing twinless mentions. The systems in Table 12 and 14 generate different twinless key mentions while keeping the twinless system mentions untouched. In Table 13 and 15, the number of twinless system mentions changes through different responses and the number of twinless key mentions is fixed.

In Table 12, B^3_{sys} recall goes up when more key mentions are resolved into the correct set. And the precision stays the same, because there is no change in the number of the erroneous resolutoins (i.e. the spurious cluster with mentions i and j). For the examples in Tables 13 and 15, B^3_{sys} gives worse precision to the outputs with more spurious resolutions, and the same recall if the systems resolve key mentions in the same way. Since the set of key mentions intersects with the set of twinless system mentions in Table 14, we do not have an intuitive explanation for the decrease in precision from response$_1$ to response$_4$. However, both the F-score and the recall still show the right tendency.

	Set 1	Set 2	B^3_{sys}		
key	{a b c d e}		P	R	F
response$_1$	{a b}	{i j}	0.857	0.280	0.422
response$_2$	{a b c}	{i j}	0.857	0.440	0.581
response$_3$	{a b c d}	{i j}	0.857	0.68	0.784
response$_4$	{a b c d e}	{i j}	0.857	1.0	0.923

Table 12: Analysis of B^3_{sys} 1

	Set 1	Set 2	B^3_{sys}		
key	{a b c d e}		P	R	F
response$_1$	{a b c}	{i j}	0.857	0.440	0.581
response$_2$	{a b c}	{i j k}	0.75	0.440	0.555
response$_3$	{a b c}	{i j k l}	0.667	0.440	0.530
response$_4$	{a b c}	{i j k l m}	0.6	0.440	0.508

Table 13: Analysis of B^3_{sys} 2

	Set 1	B^3_{sys}		
key	{a b c d e}	P	R	F
response$_1$	{a b i j}	0.643	0.280	0.390
response$_2$	{a b c i j}	0.6	0.440	0.508
response$_3$	{a b c d i j}	0.571	0.68	0.621
response$_4$	{a b c d e i j}	0.551	1.0	0.711

Table 14: Analysis of B^3_{sys} 3

	Set 1	B^3_{sys}		
key	{a b c d e}	P	R	F
response$_1$	{a b c i j}	0.6	0.440	0.508
response$_2$	{a b c i j k}	0.5	0.440	0.468
response$_3$	{a b c i j k l}	0.429	0.440	0.434
response$_4$	{a b c i j k l m}	0.375	0.440	0.405

Table 15: Analysis of B^3_{sys} 4

Probabilistic Ontology Trees for Belief Tracking in Dialog Systems

Neville Mehta
Oregon State University
mehtane@eecs.oregonstate.edu

Rakesh Gupta
Honda Research Institute
rgupta@hra.com

Antoine Raux
Honda Research Institute
araux@hra.com

Deepak Ramachandran
Honda Research Institute
dramachandran@hra.com

Stefan Krawczyk
Stanford University
stefank@cs.stanford.edu

Abstract

We introduce a novel approach for robust belief tracking of user intention within a spoken dialog system. The space of user intentions is modeled by a probabilistic extension of the underlying domain ontology called a probabilistic ontology tree (POT). POTs embody a principled approach to leverage the dependencies among domain concepts and incorporate corroborating or conflicting dialog observations in the form of interpreted user utterances across dialog turns. We tailor standard inference algorithms to the POT framework to efficiently compute the user intentions in terms of m-best most probable explanations. We empirically validate the efficacy of our POT and compare it to a hierarchical frame-based approach in experiments with users of a tourism information system.

1 Introduction

A central function of a spoken dialog system (SDS) is to estimate the user's intention based on the utterances. The information gathered across multiple turns needs to be combined and understood in context after automatic speech recognition (ASR). Traditionally, this has been addressed by dialog models and data structures such as forms (Goddeau et al., 1996) and hierarchical task decomposition (Rich and Sidner, 1998). To formalize knowledge representation within the SDS and enable the development of reusable software and resources, researchers have investigated the organization of domain concepts using IS-A/HAS-A ontologies (van Zanten, 1998; Noh et al., 2003).

Because the SDS only has access to noisy observations of what the user really uttered due to speech recognition and understanding errors, belief tracking in speech understanding has received particular attention from proponents of probabilistic approaches to dialog management (Bohus and Rudnicky, 2006; Williams, 2006). The mechanism for belief tracking often employs a Bayesian network (BN) that represents the joint probability space of concepts while leveraging conditional independences among them (Paek and Horvitz, 2000). Designing a domain-specific BN requires significant effort and expert knowledge that is not always readily available. Additionally, real-world systems typically yield large networks on which inference is intractable without major assumptions and approximations. A common workaround to mitigate the intensive computation of the joint distribution over user intentions is to assume full conditional independence between concepts which violates the ground truth in most domains (Bohus and Rudnicky, 2006; Williams, 2006).

We propose a novel approach to belief tracking for an SDS that solves both the design and tractability issues while making more realistic conditional independence assumptions. We represent the space of user intentions via a *probabilistic ontology tree* (POT) which is a tree-structured BN whose structure is directly derived from the hierarchical concept structure of the domain specified via an IS-A/HAS-A ontology. The specialization (IS-A) and composition (HAS-A) relationships between the domain concepts are intuitive and provide a systematic way of representing ontological knowledge for a wide range of domains.

The remainder of the paper is structured as follows. We begin by describing the construction of the POT given a domain ontology. We show how a POT employs null semantics to represent consistent user intentions based on the specialization and composition constraints of the domain. We then show how standard inference algorithms can be tailored to exploit the characteristics of the POT to efficiently infer the m-best list of probable explanations of user intentions given the observa-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 37–46,
The University of Tokyo, September 24-25, 2010.

tions. The POT and the associated inference algorithm empower a dialog manager (DM) to account for uncertainty while avoiding the design complexity, intractability issues, and other restrictive assumptions that characterize state-of-the-art systems. The section on empirical evaluation describes experiments in a tourist information domain that compare the performance of the POT system to a frame-based baseline system. The paper concludes with a discussion of related work.

2 Problem Formulation

Let $\{X_1, X_2, \ldots, X_N\}$ be a set of N concepts. Every concept X_i takes its value from its finite discrete domain $\mathcal{D}(X_i)$ which includes a special null element for the cases where X_i is irrelevant. The user intention space is defined as $\mathcal{U} = \mathcal{D}(X_1) \times \mathcal{D}(X_2) \times \cdots \times \mathcal{D}(X_N)$. At each dialog turn t, the system makes a noisy observation o_t about the true user intention $u \in \mathcal{U}$. o_t consists of a set of *slots*. A slot is a tuple $\langle v, d, c \rangle$ where $v \in \{X_1, \ldots, X_N\}$, $d \in \mathcal{D}(v)$ is a value of v, and $c \in \mathbb{R}$ is the confidence score assigned to that concept-value combination by the speech understanding (SU) system.

The *goal* of *belief tracking* is to maintain $\Pr(X_1, \ldots, X_N | o_1, \ldots, o_t)$, a distribution over the N-dimensional space $\mathcal{U}$ conditioned on all the observations made up to turn t. At each turn, the belief is updated based on the new observations to estimate the true, unobserved, user intention.

3 Probabilistic Ontology Trees

We model the space of the user intentions via a POT. A POT is a tree-structured BN that extends a domain ontology by specifying probability distributions over its possible instantiations based on specializations and compositions.

3.1 Domain Ontology

To ensure that the corresponding POTs are tree-structured, we consider a restricted class of domain ontologies over concepts.

Definition 1. *A* domain ontology *is a labeled directed acyclic graph. The set of vertices (corresponding to the domain concepts) is partitioned into $\{V_0\}$, V_S, and V_C, where V_0 is the only root node, V_S is the set of* specialization *nodes (related via* IS-A *to their parents), and V_C is the set of* composition *nodes (related via* HAS-A *to their parents). The set of edges satisfy the constraints that a specialization node has exactly one parent and a composition node may only have more than one parent if they are all specialization nodes with a common parent.*

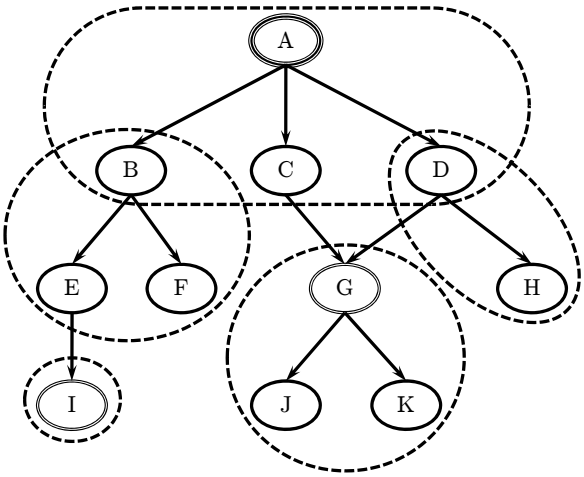


Figure 1: The ontology for a sample domain where B IS-A A, C IS-A A, D IS-A A, E IS-A B, F IS-A B, C HAS-A G (essential), D HAS-A G (nonessential), H IS-A D, E HAS-A I (essential), J IS-A G, and K IS-A G. Specialization nodes are drawn single-lined, composition nodes are drawn double-lined, and the root node is drawn triple-lined. Specialization subtrees are marked by dashed ovals.

Specialization nodes represent refinements of their parent concepts. Specializations of a concept are disjoint, that is, for any particular instance of the parent exactly one specialization is applicable and the rest are inapplicable. For example, if Dog IS-A Animal and Cat IS-A Animal, then Cat is inapplicable when Dog is applicable, and vice versa. Composition nodes represent attributes of their parents and may be essential or nonessential, e.g., Dog HAS-A Color (essential), Dog HAS-A Tail (nonessential). These definitions correspond with the standard semantics in the knowledge representation community (Noh et al., 2003). An example ontology is shown in Figure 1.

Definition 2. *A* specialization subtree (spec-tree) *in the ontology is a subtree consisting of a node with its specialization children (if any).*

3.2 POT Construction

We now describe how a POT may be constructed from a domain ontology. The purpose of the POT is to maintain a distribution of possible instantiations of the ontology such that the ontological structure is respected.

Given an ontology G, the corresponding POT is a tree-structured BN defined as follows:

Variables. Let T be a spec-tree in G with root R. Unless R is a (non-root) specialization node with no specialization children, T is represented in the POT by a variable X with the domain

$$\mathcal{D}(X) = \begin{cases} \{\mathsf{exists}, \mathsf{null}\}, \text{ if } \mathrm{Children}_T(R) = \varnothing \\ \mathrm{Children}_T(R), \text{ if } R = V_0 \\ \mathrm{Children}_T(R) \cup \{\mathsf{null}\}, \text{ otherwise.} \end{cases}$$

Edges. Let POT variables X and Y correspond to distinct spec-trees T_X and T_Y in G. There is a directed edge from X to Y if and only if either

- A leaf of T_X is the root of T_Y.
- There is an edge from a leaf in T_X to the non-specialization root of T_Y.
- There is an edge from the non-specialization root of T_X to that of T_Y.

Conditional Probability Tables (CPTs). If X (corresponding to spec-tree T_X) is the parent of Y (corresponding to spec-tree T_Y) in the POT, then Y's CPT is conditioned as follows:

- If T_Y is rooted at one of the leaves of T_X, then

$$\Pr(Y = \mathsf{null} | X = \mathsf{Y}) = 0$$
$$\Pr(Y = \mathsf{null} | X \neq \mathsf{Y}) = 1$$

  where Y is the domain value of X corresponding to child Y.
- If R is the root of T_X, and T_Y has a composition root node that is attached only to nodes in $S \subset \mathrm{Children}_{T_X}(R)$, then

$$\Pr(Y = \mathsf{null} | X = \mathsf{V}) = 1$$

  for any domain value V of X corresponding to a node $V \in \mathrm{Children}_{T_X}(R) - S$.
- If the root of T_Y is an essential composition node attached to a leaf V of T_X, then

$$\Pr(Y = \mathsf{null} | X = \mathsf{V}) = 0$$

  where V is the domain value of X corresponding to the leaf V.

We label a POT variable with that of the root of the corresponding spec-tree for convenience. The domain of a POT variable representing a spec-tree comprises the specialization children (node names in sanserif font) and the special value null; the null value allows us to render any node (except the root) inapplicable. Spec-trees comprising single nodes have the domain value exists to switch between being applicable and inapplicable. The CPT entries determine the joint probabilities over possible valid instantiations of the ontology and could be based on expert knowledge or learned from data. The conditions we impose on them (*null semantics*) ensure that inconsistent instantiations of the ontology have probability 0 in the POT. While the ontology might have undirected cycles involving the children of spec-trees, the corresponding POT is a tree because spec-trees in the ontology collapse into single POT nodes. The POT for the example domain is shown in Figure 2.

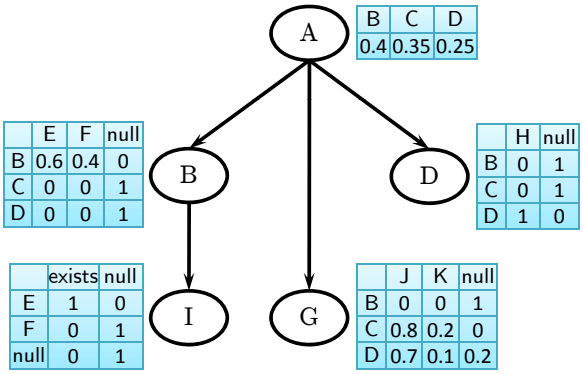


Figure 2: The POT for the example domain. If a node represents a spec-tree in the ontology, then it is labeled by the root of the spec-tree; otherwise, it is labeled with the name of the corresponding ontology node. $\mathcal{D}$(A) = {B, C, D}, $\mathcal{D}$(B) = {E, F, null}, $\mathcal{D}$(D) = {H, null}, and Pr(A), Pr(B|A) and Pr(D|A) represent some distributions over the respective specializations. $\mathcal{D}$(I) = {exists, null} and $\mathcal{D}$(G) = {J, K, null}. Note that a composition node (G) can be shared between multiple specializations (C and D) in the ontology while the resulting POT remains tree-structured.

3.3 Tourist Information POT

For the empirical analysis, we designed a POT for a tourist information system that informs the user about places to shop, eat, get service, and displays relevant information such as the distance to an intended location. The user can also provide conversational commands such as stop, reset, undo, etc. The full ontology for the tourist information domain is shown in Figure 3 and the POT is in Figure 4. In the POT, Action is the root node, with $\mathcal{D}$(Action) = {Venue, Command}, and $\mathcal{D}$(Venue)

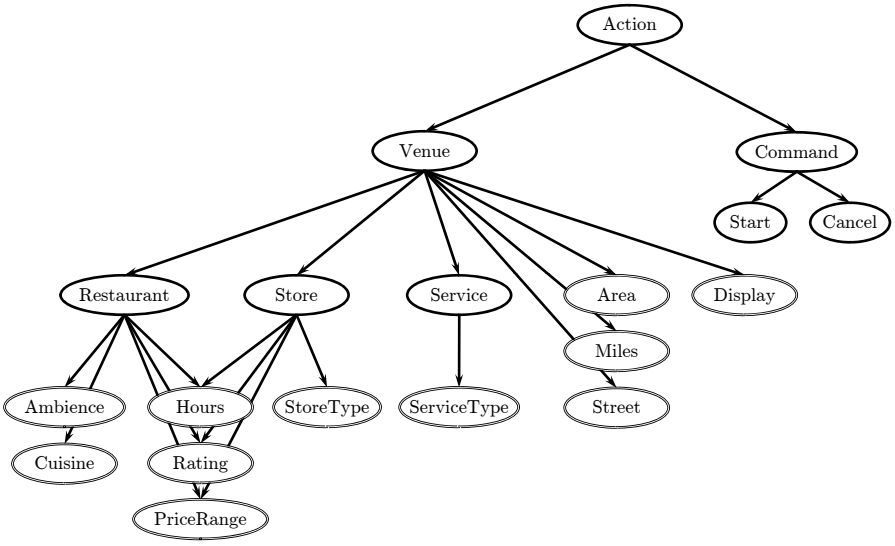


Figure 3: The ontology for the tourist information domain. All the composition nodes have specializations of their own (such as Japanese and Greek for Cuisine), but have not been shown for the sake of compactness.

= {Restaurant, Store, Service, null}. All the composition (or attribute) nodes such as Hours and Rating are made children of Venue by construction. Since a Command is inapplicable when the Action is a Venue, we have $\Pr(\text{Command} = \text{null} \mid \text{Action} = \text{Venue}) = 1$. The composition nodes (Cuisine, Street, etc.) have specializations of their own ({Japanese, Greek, ...}, {Castro, Elm, ...}, etc.), but are not shown for the sake of clarity. Since Cuisine is an essential attribute of Restaurant, $\Pr(\text{Cuisine} = \text{null} \mid \text{Venue} = \text{Restaurant}) = 0$; moreover, $\Pr(\text{Cuisine} = \text{null} \mid \text{Venue} = \text{Service}) = 1$ because Cuisine is not relevant for Service.

4 Inferring User Intention

We have seen how the POT provides the probabilistic machinery to represent domain knowledge. We now discuss how the POT structure can be leveraged to infer user intention based on the slots provided by the SU.

4.1 Soft Evidence

Every slot retrieved from the SU needs to be incorporated as observed evidence in the POT. We can set the associated node within the POT directly to its domain value as hard evidence when we know these values with certainty. Instead, we employ probabilistic observations to soften the evidence entered into the POT. We assume that the confidence score $c \in [0, 100]$ of a slot corresponds to the degree of certainty in the observation. For an observed slot variable X, we create an observation node $\hat{X}$ on the fly with the same domain as X and make it a child of X. If x is the observed value for slot X, then the CPT of $\hat{X}$ is constructed from the slot's confidence score as follows:

$$\Pr(\hat{X}|X = \mathsf{x}) = \begin{cases} \frac{c(|\mathcal{D}(X)|-1)/100+1}{|\mathcal{D}(X)|}, & \hat{X} = \mathsf{x} \\ \frac{1-c/100}{|\mathcal{D}(X)|}, & \hat{X} \neq \mathsf{x} \end{cases}$$

The probability values are generated by linearly interpolating between the uniform probability value and 1 based on the confidence score. For the remaining values,

$$\Pr(\hat{X}|X \neq \mathsf{x}) = \begin{cases} 1 - \varepsilon(|\mathcal{D}(X)| - 1), & \hat{X} = X \\ \varepsilon, & \hat{X} \neq X \end{cases}$$

where $\varepsilon > 0$.[1] Since the confidence score gives an indication of the probability for the observed value of a slot but says nothing about the remaining values, the diagonal elements for the remaining values are near 1. We cannot make them exactly 1 because the observation node needs to coexist with possibly conflicting observations in the POT.

If the user confirms the current POT hypothesis, then observations corresponding to the current hypothesis (with CPTs proportional to the score of the confirmation) are added to the POT to enforce the belief. If the user denies the current hypothesis, then all observations corresponding to the current hypothesis are removed from the POT.

[1] In our experiments, we use $\varepsilon = 10^{-10}$.

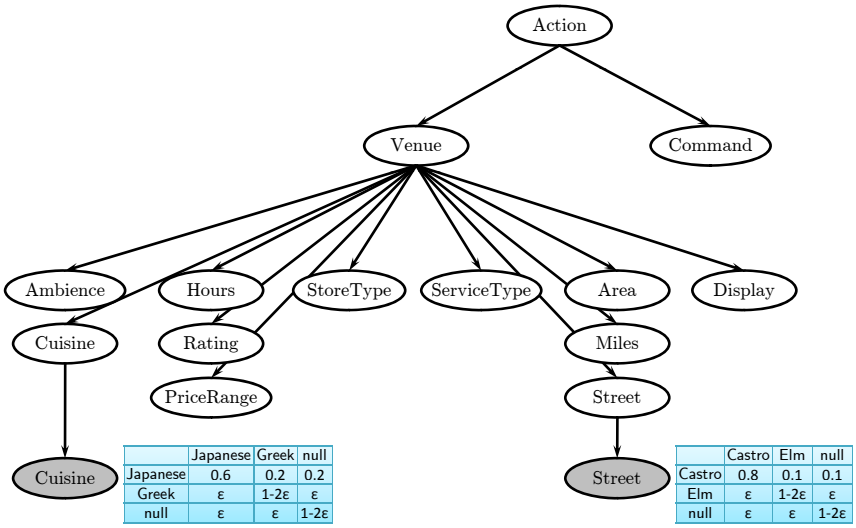


Figure 4: The POT for the tourist information domain. Assuming that $\mathcal{D}$(Cuisine) = {Japanese, Greek, null} and $\mathcal{D}$(Street) = {Castro, Elm, null}, the shaded observation nodes represent the soft evidence for input slots ⟨Cuisine, Japanese, 40⟩ and ⟨Street, Castro, 70⟩.

The POT for the tourist information domain after getting two slots as input is shown in Figure 4. The attached nodes are set to the observed slot values and the evidence propagates through the POT as explained in the next section.

4.2 POT Inference

A probable explanation (PE) or hypothesis is an assignment of values to the variables in the POT, and the most probable explanation (MPE) within the POT is the explanation that maximizes the joint probability conditioned on the observed variables. The top m estimates of the user's intentions correspond to the m-best MPEs. The design of the POT ensures that the m-best MPEs are all *consistent* across specializations, that is, exactly one specialization is applicable per node in any PE; all inconsistent explanations have a probability of 0.

The m-best MPEs could be found naively using the Join-Tree algorithm to compute the joint distribution over all variables and then use that to find the top m explanations. The space required to store the joint distribution alone is $O(n^N)$, where N is the number of nodes and n the number of values per node. Because the run time complexity is at least as much as this, it is impractical for any reasonably sized tree. However, we can get a significant speedup for a fixed m by using the properties of the POT.

Algorithm 1 uses a message-passing protocol, similar to many in the graphical models literature (Koller and Friedman, 2009), to simulate a dynamic programming procedure across the levels of the tree (see Figure 5). In Algorithm 2, an MPE message is computed at each node X using messages from the children, and sent to the parent. The message from X is the function (or table) $\psi_X(x, \vec{z})$ that represents the probabilities of the top m explanations, $\vec{z}$, of the subtree rooted at X for a particular value of $X = x$. At the root node X_0 we try all values of x_0 to find the top m MPEs for the entire tree. Note that in step 7, we

Algorithm 1 COMPUTE-PE
Input: POT T with root X_0, number of MPEs m, evidence E
Output: m MPEs for T

1: **for** $X \in T$ in reverse topological order **do**
2: Collect messages ψ_{Y_i} from all children Y_i of X
3: ψ_X = COMPUTE-MPE-MESSAGE$(X, m, \{\psi_{Y_i}\})$
4: **end for**
5: **return** top m elements of $\Pr(X_0|E)\psi_{X_0}(\cdot)$ without E

Algorithm 2 COMPUTE-MPE-MESSAGE
Input: POT node X, number of MPEs m, messages from children ψ_{Y_i}
Output: Message $\psi_X(\cdot)$

1: **if** X is a leaf node **then**
2: $\psi_X(x) \leftarrow 1, \forall x \in \mathcal{D}(X)$
3: **return** ψ_X
4: **end if**
5: **for** $x \in \mathcal{D}(X)$ **do**
6: **for** $\vec{z} = ((y_1, \vec{z_1}), \ldots, (y_k, \vec{z_k})) \in \{\mathcal{D}(\psi_{Y_1}) \times \ldots \times \mathcal{D}(\psi_{Y_k}) : \Pr(Y_i = \text{null}|X = x, E) < 1\}$ **do**
7: $\psi'_X(x, \vec{z}) \leftarrow \prod_i [\Pr(Y_i = y_i|X = x, E)\psi_{Y_i}(y_i, \vec{z_i})]$
8: **end for**
9: $\psi_X(x) \leftarrow$ top m elements of $\psi'_X(x)$.
10: **end for**
11: **return** ψ_X

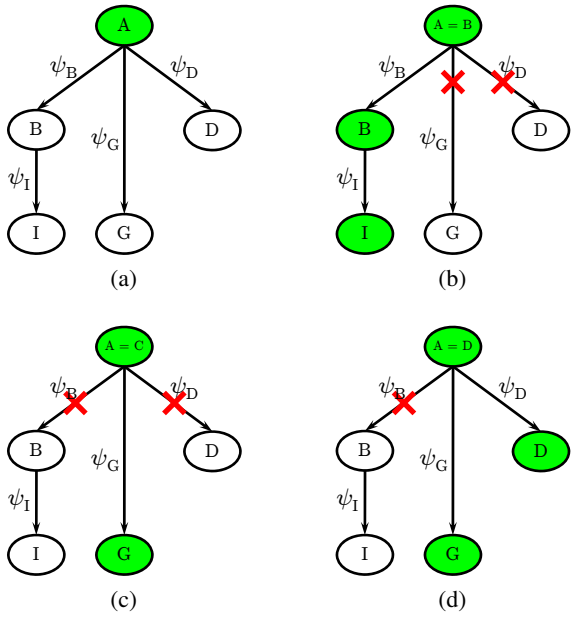


Figure 5: COMPUTE-MPE applied to the example POT. (a) Inference starts with the messages being passed up from the leaves to the root A. Every message ψ_X is an $m \times n$ table that contains the probabilities for the m-best MPEs of the subtree rooted at X for all the n domain values of X. (b) At the root, A is set to its first element B, and its marginal $\Pr(A = B)$ is combined with the message ψ_B. The semantics of the POT ensures that the other messages can be safely ignored because those subtrees are known to be null with probability 1. (c) A is set to C and only the essential attribute G is non-null. (d) A is set to its final element D, and consequently both the node D and the nonessential attribute G are non-null and their messages are mutually independent.

need the marginal $P(Y|X, E)$ which can be efficiently computed by a parallel message-passing method. Evidence nodes can only appear as leaves because of our soft evidence representation, and are encompassed by the base case. The algorithm leverages the fact that the joint of any entire subtree rooted at a node that is null with probability 1 can be safely assumed to be null with probability 1. The validity of Algorithm 1 is proven in Appendix A.

4.3 Complexity Analysis

At a POT node with at most n values and branching factor k, we do n maximizations over the product space of k nm-sized lists. Thus, the time complexity of Algorithm 1 on a POT with N nodes is $O(N(nm)^k)$ and the space complexity is $O(Nnmk)$. (Insertion sort maintains a sorted list truncated at m elements to keep track of the top m elements at any time.) However, the algorithm is significantly faster on specialization nodes because only one child is applicable and needs to be considered in the maximization (step 7). In the extreme case of a specialization-only POT, the time and space complexities both drop to $O(Nmn)$.

A similar algorithm for incrementally finding m-best MPEs in a general BN is given in Srinivas and Nayak (1996). However, our approach has the ability to leverage the null semantics in POTs resulting in significant speedup as described above. This is crucial because the run-time complexity of enumerating MPEs is known to be P^{PP}-Complete for a general BN (Kwisthout, 2008).

5 Empirical Evaluation

To test the effectiveness of our POT approach, we compare it to a frame-based baseline system for inferring user intentions.

The baseline system uses a hierarchical frame-based approach. Each frame maps to a particular user intention, and the frames are filled concurrently from the dialog observations. The slots from a turn overwrite matching slots received in previous turns. The baseline system uses the same ontology as the POT to insure that it only produces consistent hypotheses, e.g., it never produces "Venue=Service, Cuisine=Japanese" because Service does not have a Cuisine attribute. When several hypotheses compete, the system selects the one with the maximum allocated slots. We implemented the POT engine based on the Probabilistic Network Library (Intel, 2005). It takes a POT specification as input, receives the ASR slots, and returns its m-best MPEs.

Using a tourism information spoken dialog system, we collected a corpus of 375 dialogs from 15 users with a total of 720 turns (details in Appendix B). Evaluation is performed by running these collected dialogs in batch and providing the ASR slots of each turn to both the baseline and POT belief-tracking systems.[2] After each turn, both systems return their best hypothesis of the overall user intention in the form of a set of concept-value pairs. These hypothe-

[2]Speech recognition and understanding was performed using the Nuance Speech Recognition System v8.5 running manual and statistical grammars with robust interpretation.

System		Precision	Recall	F1
POT	Top hypothesis	0.84	0.81	0.83
	Top 2 hypotheses	0.87	0.84	0.85
	Top 3 hypotheses	0.89	0.85	0.87
	Top 4 hypotheses	0.91	0.86	0.89
	Top 5 hypotheses	0.92	0.86	0.89
Baseline		0.84	0.79	0.81

Table 1: Precision/recall results comparing the baseline system against the POT-based system on the 25-scenario experiment. Results are averaged over all 15 users.

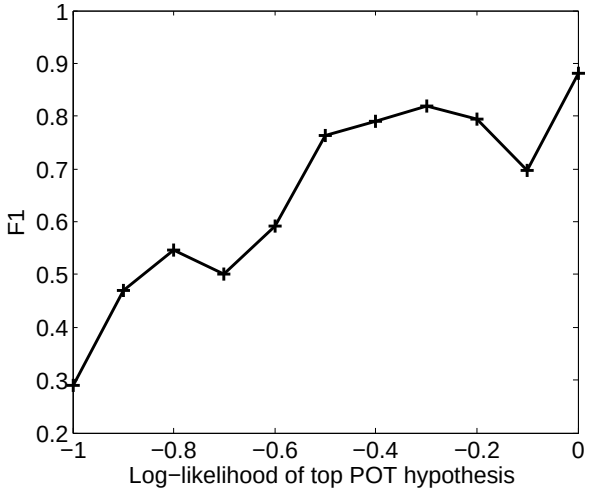


Figure 6: F1 score as a function of the log-likelihood of the top hypothesis for the user's goal.

ses are compared to the true user intention expressed so far in the dialog (e.g., if the user wants a cheap restaurant but has not mentioned it yet, PriceRange=Cheap is not considered part of the ground truth). This offline approach allows us to compare both versions on the same input.

Table 1 shows the precision/recall results for the experiment based on comparing the set of true user intention concepts to the inferred hypotheses of the POT and baseline systems. The average word error rate for all users is 29.6%. The POT system shows a 2% improvement in recall and F1 over the baseline. Additionally, leveraging the m-best hypotheses beyond just the top one could help enhance performance or guide useful clarification questions as shown by the improved performance when using the top 2–5 hypotheses; we assume an oracle for selecting the hypothesis with highest F1 among the top *m* hypotheses. All of the CPTs in the POT (besides the structural constraints) are uniformly distributed. Thus, the performance of the POT could be further improved by training the CPTs on real data.

To assess the quality of likelihood returned by the POT as a belief confidence measure, we binned dialog turns according to the log-likelihood of the top hypothesis and then computed the F1 score of each bin. Figure 6 shows that belief log-likelihood is indeed a good predictor of the F1 score. This information could be very useful to a dialog manager to trigger confirmation or clarification questions for example.

6 Discussion

The definition and construction of POTs provide a principled and systematic way to construct probabilistic models for an SDS. While any BN can be used to model the space of user intentions, designing an effective network is not an easy task for system designers not well versed in graphical models. In previous belief tracking work, researchers describe their networks with little indication on how they arrived at the specific structure (Paek and Horvitz, 2000; Thomson and Young, 2009). Prior work on ontologies for SDSs (van Zanten, 1998; Noh et al., 2003) as well as the prominence of concept hierarchies in other areas such as object-oriented programming and knowledge engineering make them a natural and intuitive way of representing SDS domains. The development of POTs builds on past research on constructing BNs based on ontological knowledge (Helsper and van der Gaag, 2002; Pfeffer et al., 1999).

While most approaches to belief tracking in the dialog systems community make a strict independence assumption between concepts (Bohus and Rudnicky, 2006; Williams, 2006), POTs model the dependencies between concepts connected by specialization and composition relationships while remaining significantly more tractable than general BNs and being very straightforward to design. The null semantics allow a POT to capture disjoint values and the applicability of attributes which are common aspects of concept ontologies. Obviously, a POT cannot capture all types of concept relationships since each concept can have only one parent. However, this restriction allows us to perform efficient exact computation of the m-best MPEs which is a significant advantage. Statistical Relational Learning approaches such as Markov Logic Networks (Richardson and Domingos, 2006) have been developed for more general relational models than strict ontologies, but they lack the parsimony and efficiency of POTs.

Thomson and Young (2009) describe an approach to dialog management based on a partially observable Markov decision process (POMDP) whose policy depends only on individual concepts' marginal distributions rather than on the overall user intention. Because their system performs belief tracking with a dynamic Bayesian network (DBN) rather than a static BN, the exact marginal computation is intractable and the authors use loopy belief propagation to compute the marginals. Even then, they indicate that the dependencies of the subgoals must be limited to enable tractability. In practice, all concepts are made independent except for the binary validity nodes that deterministically govern the dependence between nodes (similar to the null semantics of a POT). Williams (2007) also represents the user goal as a DBN for a POMDP-based DM. They perform belief updating using particle filtering and approximate the joint probability over the user intention with the product of the concept marginals. This could lead to inaccurate estimation for conditionally dependent concepts.

Among authors who have used m-best lists of dialog states for dialog management, Higashinaka et al. (2003) have shown empirically that maintaining multiple state hypotheses facilitates shorter dialogs. Their system scores each dialog state using a linear combination of linguistic and discourse features, and this score is used by a handcrafted dialog policy. While illustrating the advantages of m-best lists, this scoring approach lacks theoretical justification and ability to include prior knowledge that POTs inherit from BNs.

7 Conclusion

We have presented the POT framework for belief tracking in an SDS. We have shown how a POT can be constructed from the domain ontology and provided an exact algorithm to infer the user's intention in real-time. POTs strike a balance between representing rich concept dependencies and facilitating efficient tracking of the m-best user intentions based on exact joint probabilities rather than approximations such as concept marginals.

References

D. Bohus and A. Rudnicky. 2006. A K Hypotheses + Other Belief Updating Model. In *AAAI Workshop on Statistical and Empirical Approaches to Spoken Dialogue Systems*.

D. Goddeau, H. Meng, J. Polifroni, S. Seneff, and S. Busayapongchai. 1996. A Form-Based Dialogue Manager for Spoken Language Applications. In *ICSLP*.

E. Helsper and L. van der Gaag. 2002. Building Bayesian Networks through Ontologies. In *European Conference on Artificial Intelligence*.

R. Higashinaka, M. Nakano, and K. Aikawa. 2003. Corpus based Discourse Understanding on Spoken Dialog Systems. In *Annual Meeting on Association for Computational Linguistics*.

Intel. 2005. Probabilistic Network Library. `http://sourceforge.net/projects/openpnl/`.

D. Koller and N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

J. Kwisthout. 2008. Complexity Results for Enumerating MPE and Partial MAP. In *European Workshop on Probabilistic Graphical Models*.

H. Noh, C. Lee, and G. Lee. 2003. Ontology-based Inference for Information-seeking in Natural Language Dialog Systems. In *IEEE International Conference on Industrial Informatics*.

T. Paek and E. Horvitz. 2000. Conversation as Action under Uncertainty. In *Uncertainty in Artificial Intelligence*.

A. Pfeffer, D. Koller, B. Milch, and K. T. Takusagawa. 1999. Spook: A system for probabilistic object-oriented knowledge representation. In *Uncertainty in Artifical Intelligence*.

C. Rich and C. Sidner. 1998. COLLAGEN: a Collaboration Manager for Software Interface Agents. *An International Journal: User Modeling and User Adapted Interaction*, 8.

M. Richardson and P. Domingos. 2006. Markov Logic Networks. *Machine Learning*, 62:107–136.

S. Srinivas and P. Nayak. 1996. Efficient Enumeration of Instantiations in Bayesian Networks. In *UAI*.

B. Thomson and S. Young. 2009. Bayesian Update of Dialogue State: A POMDP Framework for Spoken Dialogue Systems. *Computer Speech and Language*.

G. van Zanten. 1998. Adaptive Mixed-Initiative Dialogue Management. In *IEEE Workshop on Interactive Voice Technology for Telecommunications Applications*.

J. Williams. 2006. Partially Observable Markov Decision Processes for Dialog Management.

J. Williams. 2007. Using Particle Filters to Track Dialogue State. In *IEEE Workshop on Automatic Speech Recognition & Understanding*.

A Analysis of the Inference Algorithm

Theorem 1. *Algorithm 1 returns the top m MPEs of the POT along with their joint probabilities.*

Proof. We first prove this for the special case of $m = 1$ to simplify notation. For the base case of a node with no children, Algorithm 2 simply returns a message with all probabilities at 1 for all values of that node. Now, consider a node X with children $Y_1, \ldots, Y_k$. Let $\text{Desc}(Y)$ be the descendants of node Y. Since Algorithm 2 given node X returns exactly one explanation, $\boldsymbol{z}$ for each $x \in \mathcal{D}(X)$, we will define $\psi_X(x) = \psi_X(x, \boldsymbol{z})$. Now, to show that $\psi_X(x) = \max_{\text{Desc}(X)} \Pr(\text{Desc}(X)|X = x, E)$, that is, Algorithm 2 returns the top explanation of the entire subtree rooted at X for every value in $\mathcal{D}(X)$, we use structural induction on the tree.

$$
\begin{aligned}
&\max_{\text{Desc}(X)} \Pr(\text{Desc}(X)|X = x, E) \\
&= \max_{Y_{1:k}, \text{Desc}(Y_{1:k})} \Pr(Y_{1:k}, \text{Desc}(Y_{1:k})|X = x, E) \\
&= \max_{Y_{1:k}, \text{Desc}(Y_{1:k})} \prod_i \Pr(Y_i|X = x, E) \Pr(\text{Desc}(Y_i)|Y_i, E) \\
&= \prod_i \max_{Y_i, \text{Desc}(Y_i)} \Big[\Pr(Y_i|X = x, E) \Pr(\text{Desc}(Y_i)|Y_i, E) \Big] \\
&= \prod_i \max_{Y_i} \Big[\Pr(Y_i|X = x, E) \max_{\text{Desc}(Y_i)} \Pr(\text{Desc}(Y_i)|Y_i, E) \Big] \\
&= \prod_i \max_{Y_i} \Big[\Pr(Y_i|X = x, E) \psi_{Y_i}(y_i) \Big] \quad \{\text{Inductive step}\} \\
&= \psi_X(x).
\end{aligned}
$$

The proof for $m > 1$, where every maximization returns a list of the top m elements, is similar. □

B Dialogs in the Tourist Information Domain

Each user conducted 25 dialogs according to prescribed scenarios for the tourist information domain. The order of scenarios was randomized for each user. Sample scenarios:

1. Find a good and cheap Mexican restaurant in Mountain View.
2. There is a medical emergency and you need to get to the hospital. Find a route.
3. You need to find your favorite coffee franchise. You have 10 minutes to get coffee.
4. Find a place to buy some fruits and vegetables.
5. Find a Chinese restaurant in Santa Clara with good ambiance, and display travel distance.
6. Find an ATM on Castro Street in Mountain View.

Figure 7 shows a typical interaction with the system for the first scenario along with a possible hypothesis inferred by the system at every turn of the dialog. Figure 8 shows an example where the POT system is able to discard an incorrect observation about a restaurant based on the accumulated belief about bookstores over multiple turns. Figure 9 shows how the POT is able to leverage the ontological structure to pick out higher-level concepts with lower confidence scores over spurious low-level concepts with higher confidence scores.

User	Find a Mexican restaurant in Mountain View.
Hypothesis	*[venue restaurant] [area mountain view] [cuisine italian]*
	{Note: Mexican is misrecognized as Italian.}
User	No, Mexican.
Hypothesis	*[venue restaurant] [area mountain view] [cuisine mexican]*
User	Show me ones with at least four star rating.
Hypothesis	*[venue restaurant] [area mountain view] [cuisine mexican] [rating four star]*
User	I want a cheap place.
Hypothesis	*[venue restaurant] [area mountain view] [cuisine mexican] [rating four star] [price cheap]*
User	Is there anything on Castro?
Hypothesis	*[venue restaurant] [area mountain view] [street castro] [cuisine mexican] [rating four star] [price cheap]*

Figure 7: A sample dialog in the tourism information domain showing the inferred hypothesis of the user's intention at every turn. The information response from the system's back-end is based on its current hypothesis.

User utterance	Where is the bookstore?
ASR	`where is the bookstore`
True hypothesis	*[action venue] [venue store] [sell book]*
Baseline hypothesis	*[action venue] [venue store] [sell book]*
POT hypothesis	*[action venue] [venue store] [sell book]*
User utterance	Store on Market Street.
ASR	`store on market street`
True hypothesis	*[action venue] [venue store] [sell book] [street market]*
Baseline hypothesis	*[action venue] [venue store] [sell book] [street market]*
POT hypothesis	*[action venue] [venue store] [sell book] [street market]*
User utterance	In downtown.
ASR	`dennys`
True hypothesis	*[action venue] [venue store] [sell book] [street market] [area downtown]*
Baseline hypothesis	*[action venue] [venue restaurant] [brand dennys]*
POT hypothesis	*[action venue] [venue store] [sell book] [street market]*

Figure 8: A dialog showing the ASR input for the user's utterance, and the corresponding true, baseline, and POT hypotheses. The POT is able to correctly discard the inconsistent observation in the third turn with the observations in previous turns.

User utterance	Where should I go to buy Lego for my kid?
SU slots	⟨Venue Store 38⟩ ⟨ServiceType GolfCourse 60⟩
True hypothesis	*[action venue] [venue store] [storetype toy]*
Baseline hypothesis	*[action venue] [venue service] [servicetype golf course]*
POT hypothesis	*[action venue] [venue store]*

Figure 9: A single dialog turn showing the SU slots for the user's utterance, and the corresponding baseline, POT, and true hypotheses. Any system that looks at the individual confidence scores will base its hypothesis on the ⟨ServiceType GolfCourse 60⟩ slot. Instead, the POT hypothesis is influenced by ⟨Venue Store 38⟩ because its score in combination with the concept's location in the POT makes it more likely than the other slot.

‘How was your day?’ An architecture for multimodal ECA systems

Raúl Santos de la Cámara
Telefónica I+D
C/ Emilio Vargas 6
28043 Madrid, Spain
e.rsai@tid.es

Markku Turunen
Univ. of Tampere
Kanslerinrinne 1
FI-33014, Finland
mturunen@cs.uta.fi

Jaakko Hakulinen
Univ. of Tampere
Kanslerinrinne 1
FI-33014, Finland
jh@cs.uta.fi

Debora Field
Computer Science
Univ. of Sheffield
S1 4DP, UK
d.field@shef.ac.uk

Abstract

Multimodal conversational dialogue systems consisting of numerous software components create challenges for the underlying software architecture and development practices. Typically, such systems are built on separate, often pre-existing components developed by different organizations and integrated in a highly iterative way. The traditional dialogue system pipeline is not flexible enough to address the needs of highly interactive systems, which include parallel processing of multimodal input and output. We present an architectural solution for a multimodal conversational social dialogue system.

1 Introduction

Multimodal conversational dialogue applications with embodied conversational agents (ECas) are complex software systems consisting of multiple software components. They require much of architectural solutions and development approaches compared to traditional spoken dialogue systems. These systems are mostly assembled from separate, often pre-existing components developed by different organizations. For such systems, the simple pipeline architecture is not a viable choice. When multimodal systems are built, software architecture should be flexible enough to enable the system to support natural interaction with features such as continuous and timely multimodal feedback and interruptions by both participants. Such features require parallel processing components and flexible communication between the components. Furthermore, the architecture should provide an open sandbox, where the components can be efficiently combined and experimented with during the iterative development process.

The HWYD (‘How was your day?’) Companion system is a multimodal virtual companion capable of affective social dialogue and for which we have developed a custom novel architecture. The application features an ECA which exhibits facial expressions and bodily movements and gestures. The system is rendered on a HD screen with the avatar being presented as roughly life-size. The user converses with the ECA using a wireless microphone. A demonstration video of the virtual companion in action is available online[1].

The application is capable of long social conversations about events that take place during a user’s working day. The system monitors the user’s emotional state on acoustic and linguistic levels, generates affective spoken responses, and attempts to positively influence the user’s emotional state. The system allows for user initiative, it asks questions, makes comments and suggestions, gives warnings, and offers advice.

2 Communications framework

The HWYD Companion system architecture employs Inamode, a loosely coupled multi-hub framework which facilitates a loose, non-hierarchical connection between any number of components. Every component in the system is connected to a repeating hub which broadcasts all messages sent to it to all connected components. The hub and the components connected to it form a single domain. Facilitators are used to forward messages between different domains according to filtering rules. During development, we have experimented with a number of Facilitators to create efficient and simple domains to overcome problems associated with single-hub systems. For example, multiple hubs allow the

[1] http://www.youtube.com/watch?v=BmDMNguQUmM

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 47–50,
The University of Tokyo, September 24-25, 2010.

reduction of broadcast messages, which is for example used in the audio processing pipeline, where a dedicated hub allows very rapid message broadcast (nearly 100 messages per second are exchanged) without compromising the stability of the system by flooding the common pipeline.

For communication between components, a lightweight communication protocol is used to support components implemented in various programming languages. A common XML message "envelope" specifies the basic format of message headers as seen in Figure 1.

```
<message
  sender              = "eca"
  id                  = "1234563862"

  msg_type            = "eca_interrupt_data"
  turn                = "12"
  msg_cause           = "interruption_occured"
  msg_sequence        = "IM-DM-ECA" >
      <payload>
            <ECAdata> data </ECAdata>
      </payload>
</message>
```

Figure 1: System message XML format

.

Mandatory elements in the envelope (top block) are necessary so other modules can identify the purpose of the message and its contents upon a shallow inspection. These include the *sender* component and a unique *message id*. Additional envelope fields elements include: *message type*, *turn id*, *dialogue segment identifier*, *recipient identifier*, and a list of message identifiers corresponding to the previous messages in the current processing sequence.

For system-wide and persistent knowledge management, a central XML-database allows the system to have inter-session and intra-session 'memory' of past events and dialogues. This database (KB) includes information such the user and dialogue models, processing status of modules, and other system-wide information.

3 Data flow in the architecture

To maximize the naturalness of the ECA's interaction, the system implements parallel processing paths. It also makes use of a special module, the **Interruption Manager (IM)**, to control components in situations where regular processing procedure must be deviated from. In addition, there are 'long' and 'short' processing sequences from user input to system output. Both 'loops' operate simultaneously. The Main Dialogue ('long') Loop, which is the normal processing path, is indicated by the bold arrows in Fig. 2, and includes all system components except the IM. The dotted arrows signal the deviations to this main path that are introduced by the

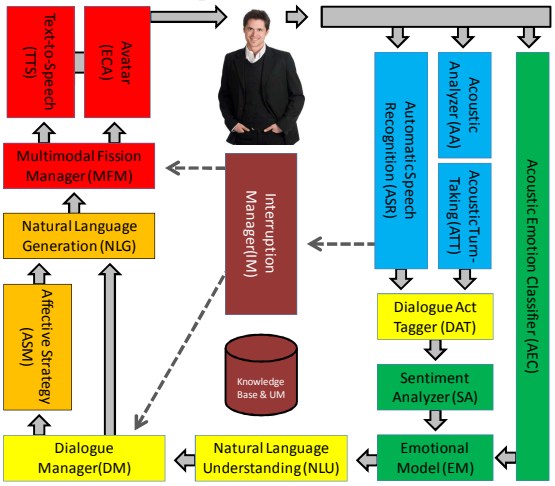


Figure 2:HWYD Companion main modules

interruption management and feedback loops. The system has an activity detector in the input subsystem that is active permanently and analyses user input in real-time. If there is a detection of user input at the same time as the ECA is talking, this module triggers a signal that is captured by the IM. The IM, which tracks the activity of the rest of the modules in the system, has a set of heuristics that are examined each time this triggering signal is detected. If any heuristic matches, the system decides there has been a proper user interruption and decides upon a series of actions to recover from the interruption.

4 Module Processing Procedure

The first stage in the processing is the acoustic processing. User speech is processed by the Acoustic Analyzer, the Automatic Speech Recognizer, and the Acoustic Emotion Classifier simultaneously for maximum responsiveness.

The **Acoustic Analyzer (AA)** extracts low-level features (pitch, intensity and the probability that the input was from voiced speech) from the acoustic signal at frequent time intervals (typically 10 milliseconds). Features are passed to the Acoustic Turn-Taking Detector in larger buffers (a few hundred milliseconds) together with timestamps. AA is implemented in TCL using Snack toolkit (http://www.speech.kth.se/snack/).

The **Acoustic Turn-Taking detector (ATT)** is a Java module, which estimates when the user has finished a turn by comparing intensity pause lengths and pitch information of user speech to configurable empirical thresholds. ATT also decides whether the user has interrupted the system

('barge-in'), while ignoring shorter backchannelling phrases (Crook et al. (2010)). Interruption messages are passed to the Interruption Manager. ATT receives a message from the ECA module when the system starts or stops speaking.

Dragon Naturally Speaking **Automatic Speech Recognition (ASR)** system is used to provide real-time large vocabulary speech recognition. Per-user acoustic adaptation is used to improve recognition rates. ASR provides N-best lists, confidence scores, and phrase hypotheses.

The **Acoustic Emotion Classifier (AEC)** component (EmoVoice (Vogt et al. (2008)) categorizes segments of user speech into five valence+arousal categories, also applying a confidence score. The Interruption Manager monitors the messages of the AEC to include emotion-related information into feedback loop messages sent to the ECA subsystem. This allows rapid reactions to the user mood.

The Sentiment Analyzer (SA) labels ASR output strings with sentiment information at word and sentence levels using valence categories *positive*, *neutral* and *negative*. The SA uses the AFFECTiS Sentiment Server, which is a general purpose .NET SOAP XML service for analysis and scoring of author sentiment.

The **Emotional Model (EM)**, written in Lisp, fuses information from the AEC and SA. It stores a globally accessible emotional representation of the user for other system modules to make use of. Affective fusion is rule-based, prefers the SA's valence information, and outputs the same five valence+arousal categories as used in the AEC. The EM can also serve as a basis for temporal integration (mood representation) as part of the affective content of the User Model. It also combines the potentially different segmentations by the ASR and AEC.

The **User Model (UM)** stores facts about the user as objects and associated attributes. The information contained in the User Model is used by other system modules, in particular by Dialogue Manager and Affective Strategy Module.

The **Dialogue Act Tagger and Segmenter (DAT)**, written in C under Linux, uses the ATT results to compile all ASR results corresponding to each user turn. DAT then segments the combined results into semantic units and labels each with a dialogue act (DA) tag (from a subset of SWBD-DAMSL (Jurafsky et al. (2001)). A Stochastic Machine Learning model combining Hidden Markov Model (HMM) and N-grams is used in a manner analogous to Martínez-Hinarejos et al. (2006). The N-grams yield the probability of a possible DA tag given the previous ones. The Viterbi algorithm is used to find the most likely sequence of DA tags.

The **Natural Language Understanding (NLU)** component, implemented in Prolog, produces a logical form representing the semantic meaning of a user turn. The NLU consists of a part-of-speech tagger, a Noun Phrase and Verb Group chunker, a named-entity classification component (rule-based), and a set of pattern-matching rules which recognize major grammatical relationships (subject, direct object, etc.) The resulting shallow-parsed text is further processed using pattern-matching rules. These recognize configurations of entity and relation relevant to the templates needed by the Dialogue Manager, the EM, and the Affective Strategy Module.

The **Dialogue Manager (DM)**, written in Java and Prolog, combines the SA and NLU results, decides on the system's next utterance and identifies salient objects for the Affective Strategy Module. The DM maintains an information state containing information about concepts under discussion, as well as the system's agenda of current conversational goals.

One of the main features of the HWYD Companion is its ability to positively influence the user's mood through its **Affective Strategy Module (ASM)**. This module appraises the user's situation, considering the events reported in the user turn and its (bi-modal) affective elements. From this appraisal, the ASM generates a long multi-utterance turn. Each utterance implements communicative acts constitutive of the strategy. ASM generates influence operators which are passed to the Natural Language Generation module. ASM output is triggered when the system has learned enough about a particular event to warrant affective influence. As input, ASM takes information extraction templates describing events, together with the emotional data attached. ASM is a Hierarchical Task Network (HTN) Planner implemented in Lisp.

The **Natural Language Generator (NLG)**, written in Lisp, produces linguistic surface forms from influence operators produced by the ASM. These operators correspond to communicative actions taking the form of performatives. NLG uses specific rhetorical structures and constructs associated with humour, and uses emotional TTS expressions through specific lexical choice.

5 Multimodal ECA Control

Multimodal control of the ECA, which consists of a tightly-synchronized naturalistic avatar and affective **Text-To-Speech (TTS)** generation, is highly challenging from an architectural viewpoint, since the coordinating component needs to be properly synchronized with the rest of the system, including both the main dialogue loop and the feedback and interruption loops.

The system Avatar is in charge of generating a three-dimensional, human-like character to serve as the system's 'face'. The avatar is connected to the TTS, and the speech is synchronized with the lip movements. The prototype is currently using the Haptek™ 3D avatar engine running inside a web browser. The Haptek engine provides a talking head and torso along with a low level API to control its interaction with any SAPI-compliant TTS subsystem, and also allows some manipulation of the character animation. An intermediate layer consisting of a Java applet and Javascript code embeds the rendered avatar in a web page and provides connectivity with the Multimodal Fission Manager. We intend to replace the current avatar with a photorealistic avatar under development within the project consortium.

Loquendo™ TTS SAPI synthesizer is used to vocalize system turns. The TTS engine works in close connection with the ECA software using the SAPI interface. TTS includes custom paralinguistic events for producing expressive speech. TTS is based on the concatenative technique with variable length acoustic units.

The Multimodal Fission Manager (MFM) controls the Avatar and the TTS engine, enabling the system to construct complex communicative acts that chain together series of utterances and gestures. It offers FML-standard-based syntax to make the avatar perform a series of body and facial gestures.

The system features a template-based input mode in which a module can call ECA to perform actions without having to build a full FML-based XML message. This is intended to be used in the feedback loops, for example, to convey the impression that the ECA is paying attention.

6 Conclusions

We have presented an advanced multimodal dialogue system that challenges the usual pipeline-based implementation. To do so, it leverages on an architecture that provides the means for a flexible component interconnection, that can accomodate the needs of a system using more than one processing path for its data. We have shown how this has enabled us to implement complex behavior such as interrupt and short loop handling. We are currently expanding coverage and will carry out an evaluation with real users this September.

Acknowledgements

This work was funded by Companions, a European Commission Sixth Framework Programme Information Society Technologies Integrated Project (IST-34434).

References

Vogt, T., André, E. and Bee, N. 2008. EmoVoice – A framework for online recognition of emotions from voice. In: *Proc. Workshop on Perception and Interactive Technologies for Speech-Based Systems*, Springer, Kloster Irsee, Germany.

Cavazza, M., Smith, C., Charlton, D., Crook, N., Boye, J., Pulman, S., Moilanen, K., Pizzi, D., Santos de la Camara, R., Turunen, M. 2010 *Persuasive Dialogue based on a Narrative Theory: an ECA Implementation*, Proc. 5th Int. Conf. on Persuasive Technology (to appear).

Hernández, A., López, B., Pardo, D., Santos, R., Hernández, L., Relaño Gil, J. and Rodríguez, M.C. 2008 Modular definition of multimodal ECA communication acts to improve dialogue robustness and depth of intention. In: Heylen, D., Kopp, S., Marsella, S., Pelachaud, C., and Vilhjálmsson, H. (Eds.), *AAMAS 2008 Workshop on Functional Markup Language.*

Crook, N., Smith, C., Cavazza, M., Pulman, S., Moore, R., and Boye, J. 2010 Handling User Interruptions in an Embodied Conversational Agent. In *Proc. AAMAS 2010.*

Wagner J., André, E., and Jung, F. 2009 Smart sensor integration: A framework for multimodal emotion recognition in real-time. In *Affective Computing and Intelligent Interaction 2009.*

Cavazza, M., Pizzi, D., Charles, F., Vogt, T. André, E. 2009 Emotional input for character‐based interactive storytelling *AAMAS (1) 2009*: 313-320.

Jurafsky, D. Shriberg, E., Biasca, D. 2001 *Switchboard swbd‐damsl shallow‐discourse‐function annotation coders manual.* Tech. Rep. 97‐01, University of Colorado Institute of Cognitive Science

Martínez‐Hinarejos, C.D., Granell, R., Benedí, J.M. 2006. *Segmented and unsegmented dialogue‐act annotation with statistical dialogue models*. Proc. COLING/ACL Sydney, Australia, pp. 563‐570.

Middleware for Incremental Processing in Conversational Agents

David Schlangen[*], **Timo Baumann**[*], **Hendrik Buschmeier**[†], **Okko Buß**[*]
Stefan Kopp[†], **Gabriel Skantze**[‡], **Ramin Yaghoubzadeh**[†]
[*]University of Potsdam Germany
[†]Bielefeld University Germany
[‡]KTH, Stockholm Sweden
david.schlangen@uni-potsdam.de

Abstract

We describe work done at three sites on designing conversational agents capable of incremental processing. We focus on the 'middleware' layer in these systems, which takes care of passing around and maintaining incremental information between the modules of such agents. All implementations are based on the abstract model of incremental dialogue processing proposed by Schlangen and Skantze (2009), and the paper shows what different instantiations of the model can look like given specific requirements and application areas.

1 Introduction

Schlangen and Skantze (2009) recently proposed an abstract model of incremental dialogue processing. While this model introduces useful concepts (briefly reviewed in the next section), it does not talk about how to actually implement such systems. We report here work done at three different sites on setting up conversational agents capable of incremental processing, inspired by the abstract model. More specifically, we discuss what may be called the 'middleware' layer in such systems, which takes care of passing around and maintaining incremental information between the modules of such agents. The three approaches illustrate a range of choices available in the implementation of such a middle layer. We will make our software available as development kits in the hope of fostering further research on incremental systems.[1]

In the next section, we briefly review the abstract model. We then describe the implementations created at Uni Bielefeld (BF), KTH Stockholm (KTH) and Uni Potsdam (UP). We close with a brief discussion of similarities and differences, and an outlook on further work.

[1]Links to the three packages described here can be found at `http://purl.org/net/Middlewares-SIGdial2010`.

2 The IU-Model of Incremental Processing

Schlangen and Skantze (2009) model incremental systems as consisting of a network of processing *modules*. Each module has a *left buffer*, a *processor*, and a *right buffer*, where the normal mode of processing is to take input from the left buffer, process it, and provide output in the right buffer, from where it goes to the next module's left buffer. (Top-down, expectation-based processing would work in the opposite direction.) Modules exchange *incremental units* (IUs), which are the smallest 'chunks' of information that can trigger connected modules into action. IUs typically are part of larger units; e.g., individual words as parts of an utterance, or frame elements as part of the representation of an utterance meaning. This relation of being part of the same larger unit is recorded through *same level links*; the information that was used in creating a given IU is linked to it via *grounded in* links. Modules have to be able to react to three basic situations: that IUs are *added* to a buffer, which triggers processing; that IUs that were erroneously hypothesised by an earlier module are *revoked*, which may trigger a revision of a module's own output; and that modules signal that they *commit* to an IU, that is, won't revoke it anymore (or, respectively, expect it to not be revoked anymore).

Implementations of this model then have to realise the actual details of this information flow, and must make available the basic module operations.

3 Sociable Agents Architecture

BF's implementation is based on the 'D-Bus' message bus system (Pennington et al., 2007), which is used for remote procedure calls and the bi-directional synchronisation of IUs, either locally between processes or over the network. The bus system provides *proxies*, which make the interface of a local object accessible remotely without copying data, thus ensuring that any access is guaranteed to yield up-to-date information. D-Bus bindings exist for most major programming languages, allowing

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 51–54,
The University of Tokyo, September 24-25, 2010.

for interoperability across various systems.

IUs exist as objects implementing a D-Bus interface, and are made available to other modules by publishing them on the bus. Modules are objects comprising a main thread and right and left buffers for holding own IUs and foreign IU proxies, respectively. Modules can co-exist in one process as threads or occupy one process each—even distributed across a network.

A dedicated *Relay* D-Bus object on the network is responsible for module administration and update notifications. At connection time, modules register with the relay, providing a list of IU categories and/or module names they are interested in. Category interests create loose functional links while module interests produce more static ones. Whenever a module chooses to publish information, it places a new IU in its right buffer, while removal of an IU from the right buffer corresponds to retraction. The relay is notified of such changes and in turn invokes a notification callback in all interested modules synchronising their left buffers by immediately and transparently creating or removing proxies of those IUs.

IUs consist of the fields described in the abstract model, and an additional category field which the relay can use to identify the set of interested modules to notify. They furthermore feature an optional custom lifetime, on the expiration of which they are automatically retracted.

Incremental changes to IUs are simply realised by changing their attributes: regardless of their location in either a right or left buffer, the same setter functions apply (e.g., `set_payload`). These generate relay-transported update messages which communicate the ID of the changed IU. Received update messages concerning self-owned and remotely-owned objects are discerned automatically to allow for special treatment of own IUs. The complete process is illustrated in Figure 1.

Current state and discussion. Our support for bi-directional IU editing is an extension to the concepts of the general model. It allows higher-level modules with a better knowledge of context to revise uncertain information offered by lower levels. Information can flow both ways, bottom-up and top-down, thus allowing for diagnostic and causal networks linked through category interests.

Coming from the field of embodied conversational agents, and being especially interested in modelling human-like communication, for example for on-line production of listener backchannel feedback, we constantly have to take incrementally changing uncertain input into account. Using the presented framework consistently as a network communication layer, we are currently modelling an entire cognitive architecture for virtual agents, based on the principle of incremental processing.

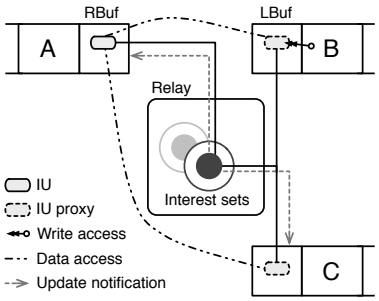


Figure 1: Data access on the IU proxies is transparently delegated over the D-Bus; module A has published an IU. B and C are registered in the corresponding interest set, thus receiving a proxy of this IU in their left buffer. When B changes the IU, A and C receive update notifications.

The decision for D-Bus as the transportation layer has enabled us to quickly develop versions for Python, C++ and Java, and produced straightforward-to-use libraries for the creation of IU-exchanging modules: the simplest fully-fledged module might only consist of a periodically invoked main loop callback function and any subset of the four handlers for IU events (added, removed, updated, committed).

4 Inpro Toolkit

The InproTK developed at UP offers flexibility on how tightly or loosely modules are coupled in a system. It provides mechanisms for sending IU updates between processes via a messaging protocol (we have used OAA [Cheyer and Martin, 2001], but other communication layers could also be used) as well as for using shared memory within one (Java) process. InproTK follows an event-based model, where modules create events, for which other modules can register as Listeners. Module networks are configured via a system configuration file which specifies which modules *listen* to which.

Modules *push* information to their right, hence the interface for inter-module communication is called *PushBuffer*. (At the moment, InproTK only implements left-to-right IU flow.) The `PushBuffer` interface defines a *hypothesis-change* method which a module will call for all its listening modules. A hypothesis change is (redundantly) characterised by passing both the complete current buffer state (a list of *IU*s) as well as the *delta* between

the previous and the current state, leaving listening modules a choice of how to implement their internal update.

Modules can be fully event-driven, only triggered into action by being notified of a hypothesis change, or they can run persistently, in order to create endogenous events like time-outs. Event-driven modules can run concurrently in separate threads or can be called sequentially by a push buffer (which may seem to run counter the spirit of incremental processing, but can be advantageous for very quick computations for which the overhead of creating threads should be avoided).

IUs are typed objects, where the base class `IU` specifies the links (same-level, grounded-in) that allow to create the IU network and handles the assignment of unique IDs. The payload and additional properties of an IU are specified for the IU's *type*. A design principle here is to make all relevant information available, while avoiding replication. For instance, an IU holding a bit of semantic representation can query which interval of input data it is based on, where this information is retrieved from the appropriate IUs by automatically following the grounded-in links. IU networks ground out in `BaseData`, which contains user-side input such as speech from the microphone, derived ASR feature vectors, camera feeds from a webcam, derived gaze information, etc., in several streams that can be accessed based on their timing information.

Besides IU communication as described in the abstract model, the toolkit also provides a separate communication track along which *signals*, which are any kind of information that is not seen as incremental hypotheses about a larger whole but as information about a single current event, can be passed between modules. This communication track also follows the observer/listener model, where processors define interfaces that listeners can implement.

Finally, InproTK also comes with an extensive set of monitoring and profiling modules which can be linked into the module network at any point and allow to stream data to disk or to visualise it online through a viewing tool (ANON 2009), as well as different ways to simulate input (e.g., typed or read from a file) for bulk testing.

Current state and discussion. InproTK is currently used in our development of an incremental multimodal conversational system. It is usable in its current state, but still evolves. We have built and integrated modules for various tasks (post-processing of ASR output, symbolic and statistical natural language understanding [ANON 2009a,b,c]). The configuration system and the availability of monitoring and visualisation tools enables us to quickly test different setups and compare different implementations of the same tasks.

5 Jindigo

Jindigo is a Java-based framework for implementing and experimenting with incremental dialogue systems currently being developed at KTH. In Jindigo, all modules run as separate threads within a single Java process (although the modules themselves may of course communicate with external processes). Similarly to InproTK, IUs are modelled as typed objects. The modules in the system are also typed objects, but buffers are not. Instead, a buffer can be regarded as a set of IUs that are connected by (typed) same-level links. Since all modules have access to the same memory space, they can follow the same-level links to examine (and possibly alter) the buffer. Update messages between modules are relayed based on a system specification that defines which types of update messages from a specific module go where. Since the modules run asynchronously, update messages do not directly invoke methods in other modules, but are put on the input queues of the receiving modules. The update messages are then processed by each module in their own thread.

Jindigo implements a model for updating buffers that is slightly different than the two previous approaches. In this approach, IUs are connected by *predecessor* links, which gives each IU (words, widest spanning phrases from the parser, communicative acts, etc), a position in a (chronologically) ordered stream. Positional information is reified by super-imposing a network of position nodes over the IU network, with the IUs being associated with edges in that network. These positional nodes then give us names for certain update stages, and so revisions can be efficiently encoded by reference to these nodes. An example can make this clearer. Figure 2 shows five update steps in the right buffer of an incremental ASR module. By reference to positional nodes, we can communicate easily (a) what the newest committed IU is (indicated in the figure as a shaded node) and (b) what the newest non-revoked or active IU is (i.e., the 'right edge' (RE); indicated in the figure as a node with a dashed line). So, the change between the state at time t_1 and t_2 is signalled by RE taking on a different value. This

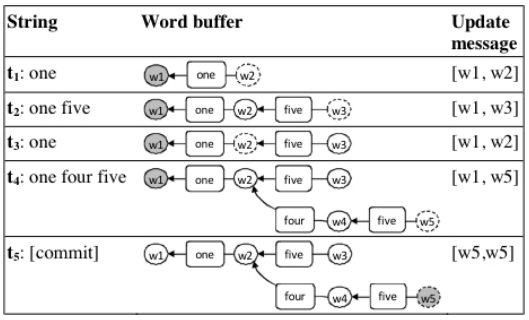


Figure 2: The right buffer of an ASR module, and update messages at different time-steps.

value (w3) has not been seen before, and so the consuming module can infer that the network has been extended; it can find out which IUs have been added by going back from the new RE to the last previously seen position (in this case, w2). At t_3, a retraction of a hypothesis is signalled by a return to a previous state, w2. All consuming modules have to do now is to return to an internal state linked to this previous input state. Commitment is represented similarly through a pointer to the rightmost committed node; in the figure, that is for example w5 at t_5.

Since information about whether an IU has been revoked or committed is not stored in the IU itself, all IUs can (if desirable) be defined as immutable objects. This way, the pitfalls of having asynchronous processes altering and accessing the state of the IUs may be avoided (while, however, more new IUs have to be created, as compared to altering old ones). Note also that this model supports parallel hypotheses as well, in which case the positional network would turn into a lattice.

The framework supports different types of update messages and buffers. For example, a parser may incrementally send NPs to a reference resolution (RR) module that has access to a domain model, in order to prune the chart. Thus, information may go both left-to-right and right-to-left. In the buffer between these modules, the order between the NPs that are to be annotated is not important and there is no point in revoking such IUs (since they do not affect the RR module's state).

Current state and discussion. Jindigo uses concepts from (Skantze, 2007), but has been rebuilt from ground up to support incrementality. A range of modules for ASR, semantic interpretation, TTS, monitoring, etc., have been implemented within the framework, allowing us to do experiments with complete systems interacting with users. We are currently using the framework to implement a model of incremental speech production.

6 Discussion

The three implementations of the abstract IU model presented above show that concrete requirements and application areas result in different design decisions and focal points.

While BF's approach is loosely coupled and handles exchange of IUs via shared objects and a mediating module, KTH's implementation is rather closely coupled and publishes IUs through a single buffer that lies in shared memory. UP's approach is somewhat in between: it abstracts away from the transportation layer and enables message passing-based communication as well as shared memory transparently through one interface.

The differences in the underlying module communication infrastructure affect the way incremental IU updates are handled in the systems. In BF's framework modules holding an IU in one of their buffers just get notified when one of the IU's fields changed. Conversely, KTH's IUs are immutable and new information always results in new IUs being published and a change to the graph representation of the buffer—but this allows an efficient coupling of module states and cheap revoke operations. Again, UP's implementation lies in the middle. Here both the whole new state and the delta between the old and new buffer is communicated, which leads to flexibility in how consumers can be implemented, but also potentially to some communication overhead.

In future work, we will explore if further generalisations can be extracted from the different implementations presented here. For now, we hope that the reference architectures presented here can already be an inspiration for further work on incremental conversational systems.

References

Adam Cheyer and David Martin. 2001. The open agent architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1):143–148, March.

H. Pennington, A. Carlsson, and A. Larsson. 2007. *D-Bus Specification Version 0.12*. http://dbus.freedesktop.org/doc/dbus-specification.html.

David Schlangen and Gabriel Skantze. 2009. A General, Abstract Model of Incremental Dialogue Processing. In *Proceedings of EACL 2009*, Athens, Greece.

Gabriel Skantze. 2007. *Error Handling in Spoken Dialogue Systems*. Ph.D. thesis, KTH, Stockholm, Sweden, November.

Towards Semi-Supervised Classification of Discourse Relations using Feature Correlations

Hugo Hernault and **Danushka Bollegala** and **Mitsuru Ishizuka**
Graduate School of Information Science & Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
hugo@mi.ci.i.u-tokyo.ac.jp
danushka@iba.t.u-tokyo.ac.jp
ishizuka@i.u-tokyo.ac.jp

Abstract

Two of the main corpora available for training discourse relation classifiers are the RST Discourse Treebank (RST-DT) and the Penn Discourse Treebank (PDTB), which are both based on the Wall Street Journal corpus. Most recent work using discourse relation classifiers have employed fully-supervised methods on these corpora. However, certain discourse relations have little labeled data, causing low classification performance for their associated classes. In this paper, we attempt to tackle this problem by employing a semi-supervised method for discourse relation classification. The proposed method is based on the analysis of feature co-occurrences in unlabeled data. This information is then used as a basis to extend the feature vectors during training. The proposed method is evaluated on both RST-DT and PDTB, where it significantly outperformed baseline classifiers. We believe that the proposed method is a first step towards improving classification performance, particularly for discourse relations lacking annotated data.

1 Introduction

The RST Discourse Treebank (RST-DT) (Carlson et al., 2001), based on the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) framework, and the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), are two of the most widely-used corpora for training discourse relation classifiers. They are both based on the Wall Street Journal (WSJ) corpus, although there are substantial differences in the relation taxonomy used to annotate the corpus. These corpora have been used in most of the recent work employing discourse relation classifiers, which are based on fully-supervised machine learning approaches (duVerle and Prendinger, 2009; Pitler et al., 2009; Lin et al., 2009).

Still, when building a discourse relation classifier on either corpus, one is faced with the same practical issue: Certain relations are very prevalent, such as ELABORATION[N][S] (RST-DT), with more than 4000 instances, whereas other occur rarely, such as EVALUATION[N][N][1] (RST-DT), with three instances, or COMPARISON.PRAGMATIC CONCESSION (PDTB), with 12 instances. This lack of training data causes poor classification performance on the classes associated to these relations.

In this paper, we try to tackle this problem by using feature co-occurrence information, extracted from unlabeled data, as a way to inform the classifier when unseen features are found in test vectors. The advantage of the method is that it relies solely on unlabeled data, which is abundant, and cheap to collect.

The contributions of this paper are the following: First, we propose a semi-supervised method that exploits the abundant, freely-available unlabeled data, which is harvested for feature co-occurrence information, and used as a basis to extend feature vectors to help classification for cases where unknown features are found in test vectors. Second, the proposed method is evaluated on the RST-DT and PDTB corpus, where it significantly improves F-score when trained on moderately small datasets. For instance, when trained on a dataset with around 1000 instances, the proposed method increases the macro-average F-score up to 30%, compared to a baseline classifier.

2 Related Work

Since the release in 2002 of the RST-DT corpus, several fully-supervised discourse parsers have

[1] We use the notation [N] and [S] respectively to denote the nucleus and satellite in a RST discourse relation.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 55–58,
The University of Tokyo, September 24-25, 2010.

been built in the RST framework. In duVerle and Prendinger (2009), a discourse parser based on Support Vector Machines (SVM) (Vapnik, 1995) is proposed. Shallow lexical, syntactic and structural features, including 'dominance sets' (Soricut and Marcu, 2003) are used.

The unsupervised method of Marcu and Echihabi (2002) was the first to try to detect 'implicit' relations (i.e. relations not accompanied by a cue phrase, such as *'however'*, *'but'*), using word pairs extracted from two spans of text. Their method attempts to capture the difference of polarity in words.

Discourse relation classifiers have also been trained using PDTB. Pitler et al. (2008) performed a corpus study of the PDTB, and found that 'explicit' relations can be most of the times distinguished by their discourse connectives.

Lin et al. (2009) studied the problem of detecting implicit relations in PDTB. Their relational classifier is trained using features extracted from dependency paths, contextual information, word pairs and production rules in parse trees. For the same task, Pitler et al. (2009) also use word pairs, as well as several other types of features such as verb classes, modality, context, and lexical features.

In this paper, we are not aiming at defining novel features for improving performance in RST or PDTB relation classification. Instead we incorporate features that have already shown to be useful for discourse relation learning and explore the possibilities of using unlabeled data for this task.

3 Method

In this section, we describe a semi-supervised method for relation classification, based on feature vector extension. The extension process employs feature co-occurrence information. Co-occurrence information is useful in this context as, for instance, we might know that the word pair *(for, when)* is a good indicator of a TEMPORAL relation. Or, after analyzing a large body of unlabeled data, we might also notice that this word pair co-occurs often with the word 'run-up' placed at the end of a span of text. Suppose now that we have to classify a test instance containing the feature 'run-up', but not the word pair *(for, when)*. In this case, by using the co-occurrence information, we know that the instance has a chance of being a TEMPORAL relation. We first explain how to compute a feature correlation matrix, using unlabeled data. In a second section, we show how to extend feature vectors in order to include co-occurrence information. Finally, we describe the features used in the discourse relation classifiers.

3.1 Feature Correlation Matrix

A training/test instance is represented using a d-dimensional feature vector $\mathbf{f} = [f_1, \ldots, f_d]^{\mathrm{T}}$, where $f_i \in \{0, 1\}$. We define a *feature correlation matrix*, C such that the (i, j)-th element of C, $C_{(i,j)} \in \{0, 1\}$ denotes the correlation between the two features f_i and f_j. If both f_i and f_j appear in a feature vector then we define them to be co-occurring. The number of different feature vectors in which f_i and f_j co-occur is used as a basis to compute $C_{(i,j)}$. Importantly, feature correlations can be calculated using only unlabeled data.

It is noteworthy that feature correlation matrices can be computed using any correlation measure. For the current task we use the χ^2-measure (Plackett, 1983) as the preferred correlation measure because of its simplicity. We create the feature correlation matrix C, such that, for all pairs of features (f_i, f_j),

$$C_{(i,j)} = \begin{cases} 1 \text{ if } \chi^2_{i,j} > c \\ 0 \text{ otherwise} \end{cases} . \quad (1)$$

Here c is the critical value, which, for a confidence level of 0.05 and one degree of freedom, can be set to 3.84.

3.2 Feature Vector Extension

Once the feature correlation matrix is computed using unlabeled data as described in Section 3.1, we can use it to extend a feature vector during testing. One of the reasons explaining why a classifier might perform poorly on a test instance, is that there are features in the test instance that were not observed during training. Let us represent the feature vector corresponding to a test instance x by $\mathbf{f_x}$. Then, we use the feature correlation matrix to find the set of correlated features $F_c(f_i)$ of a particular feature f_i that occur in $\mathbf{f_x}$.

Specifically, for a feature $f_i \in \mathbf{f_x}$, $F'(f_i)$ consists of features f_j, where $C_{(i,j)} = 1$. We define the extended feature vector $\mathbf{f'_x}$ of $\mathbf{f_x}$ as the union of all the features that appear in $\mathbf{f_x}$ and $F_c(f_x)$. Since a discourse relation is defined between two spans of short texts (elementary discourse units), which are typically two clauses or sentences, a particular feature does not usually occur more than once

in a feature vector. Therefore, we introduced the proposed method in the context of binary valued features. However, the above mentioned discussion can be naturally extended to cover real-valued features.

3.3 Features

Figure 1 shows the parse tree for a sentence composed of two discourse units, which serve as arguments of a discourse relation we want to generate a feature vector from. Lexical heads have been calculated using the projection rules of Magerman (1995), and indicated between brackets. For each argument, surrounded by dots, is the minimal set of sub-parse trees containing strictly all the words of the argument.

We extract all possible lemmatized word pairs from the two arguments. Next, we extract from left and right argument separately, all production rules from the sub-parse trees. Finally, we encode in our features three nodes of the parse tree, which capture the local context at the connection point between the two arguments (Soricut and Marcu, 2003): The first node, which we call N_w, is the highest ancestor of the first argument's last word w, and is such that N_w's right-sibling is the ancestor of the second argument's first word. N_w's right-sibling node is called N_r. Finally, we call N_p the parent of N_w and N_r. For each node, we encode in the feature vector its part-of-speech (POS) and lexical head. For instance, in Figure 1, we have N_w = S(comment), N_r = SBAR(when), and N_p = VP(declined).

4 Experiments

It is worth noting that the proposed method is independent of any particular classification algorithm. As our goal is strictly to evaluate the relative benefit of employing the proposed method, we select a logistic regression classifier, for its simplicity. We used the multi-class logistic regression (maximum entropy model) implemented in Classias (Okazaki, 2009). Regularization parameters are set to their default value of one.

Unlabeled instances are created by selecting texts of the WSJ, and segmenting them into elementary discourse units (EDUs) using our sequential discourse segmenter (Hernault et al., 2010). As there is no segmentation tool for the PDTB framework, we assumed that feature correlation information taken from EDUs created using a RST segmenter is also useful for extending feature vectors of PDTB relations.

Since we are interested in measuring the overall performance of a discourse relation classifier across all relation types, we use macro-averaged F-score as the preferred evaluation metric for this task. We train a multi-class logistic regression model without extending the feature vectors as a baseline method. This baseline is expected to show the effect of using the proposed feature extension approach for the task of discourse relation learning.

Experimental results on RST-DT and PDTB datasets are depicted in Figures 2 and 3. We observe that the proposed feature extension method outperforms the baseline for both RST-DT and PDTB datasets for the full range of training dataset sizes. However, the difference between the two methods decreases as we increase the amount of training data. Specifically, with 200 training instances, for RST-DT, the baseline method has a macro-averaged F-score of 0.079, whereas the the proposed method has a macro-averaged F-score of 0.159 (around 101% increase in F-score). For 1000 training instances, the F-score for RST-DT increases by 29.2%, from 0.143 to 0.185, while the F-score for PDTB increases by 27.9%, from 0.109 to 0.139. However, the difference between the two methods diminishes beyond 10000 training instances.

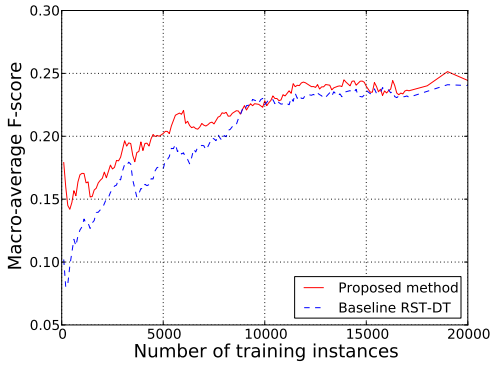


Figure 2: Macro-average F-score (RST-DT) as a function of the number of training instances used.

5 Conclusion

We presented a semi-supervised method for improving the performance of discourse relation classifiers. The proposed method is based on the analysis of co-occurrence information harvested from unlabeled data only. We evaluated

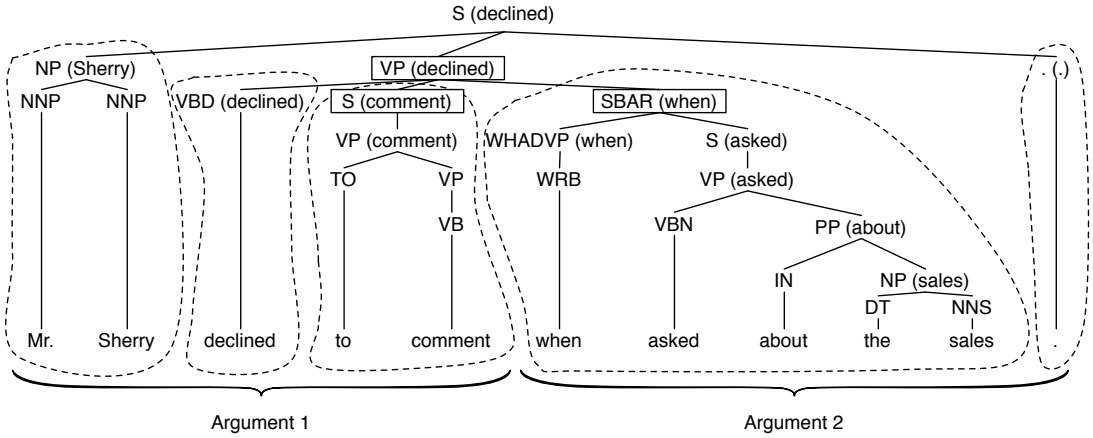


Figure 1: Two arguments of a discourse relation, and the minimum set of subtrees that contain them—lexical heads are indicated between brackets.

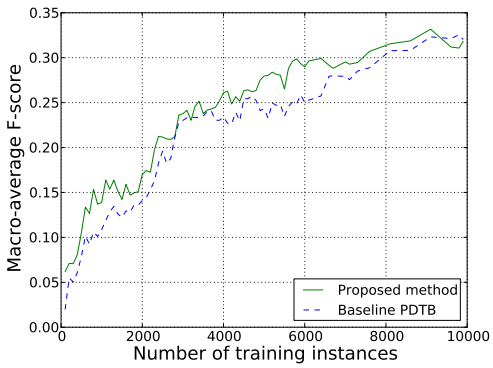


Figure 3: Macro-average F-score (PDTB) as a function of the number of training instances used.

the method on two of the most widely-used discourse corpora, RST-DT and PDTB. The method performs significantly better than a baseline classifier trained on the same features, especially when the number of labeled instances used for training is small. For instance, using 1000 training instances, we observed an increase of nearly 30% in macro-average F-score. This is an interesting perspective for improving classification performance of relations with little training data. In the future, we plan to improve the method by employing ranked co-occurrences. This way, only the most relevant correlated features can be selected during feature vector extension. Finally, we plan to investigate using larger amounts of unlabeled training data.

References

L. Carlson, D. Marcu, and M. E. Okurowski. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. *Proc. of Second SIGdial Workshop on Discourse and Dialogue-Volume 16*, pages 1–10.

D. A. duVerle and H. Prendinger. 2009. A novel discourse parser based on Support Vector Machine classification. In *Proc. of ACL'09*, pages 665–673.

H. Hernault, D. Bollegala, and M. Ishizuka. 2010. A sequential model for discourse segmentation. In *Proc. of CICLing'10*, pages 315–326.

Z. Lin, M-Y. Kan, and H. T. Ng. 2009. Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proc. of EMNLP'09*, pages 343–351.

D. M. Magerman. 1995. Statistical decision-tree models for parsing. *Proc. of ACL'95*, pages 276–283.

W. C. Mann and S. A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

D. Marcu and A. Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proc. of ACL'02*, pages 368–375.

N. Okazaki. 2009. Classias: A collection of machine-learning algorithms for classification.

E. Pitler, M. Raghupathy, H. Mehta, A. Nenkova, A. Lee, and A. Joshi. 2008. Easily identifiable discourse relations. In *Proc. of COLING'08 (Posters)*, pages 87–90.

E. Pitler, A. Louis, and A. Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proc. of ACL'09*, pages 683–691.

R. L. Plackett. 1983. Karl Pearson and the chi-squared test. *International Statistical Review*, 51(1):59–72.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The Penn Discourse Treebank 2.0. In *Proc. of LREC'08*.

R. Soricut and D. Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. *Proc. of NA-ACL'03*, 1:149–156.

V. N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc.

Using entity features to classify implicit discourse relations

Annie Louis, Aravind Joshi, Rashmi Prasad, Ani Nenkova
University of Pennsylvania
Philadelphia, PA 19104, USA
{lannie,joshi,rjprasad,nenkova}@seas.upenn.edu

Abstract

We report results on predicting the sense of implicit discourse relations between adjacent sentences in text. Our investigation concentrates on the association between discourse relations and properties of the referring expressions that appear in the related sentences. The properties of interest include coreference information, grammatical role, information status and syntactic form of referring expressions. Predicting the sense of implicit discourse relations based on these features is considerably better than a random baseline and several of the most discriminative features conform with linguistic intuitions. However, these features do not perform as well as lexical features traditionally used for sense prediction.

1 Introduction

Coherent text is described in terms of discourse relations such as "cause" and "contrast" between its constituent clauses. It is also characterized by entity coherence, where the connectedness of the text is created by virtue of the mentioned entities and the properties of referring expressions. We aim to investigate the association between discourse relations and the way in which references to entities are realized. In our work, we employ features related to entity realization to automatically identify discourse relations in text.

We focus on implicit relations that hold between adjacent sentences in the absence of discourse connectives such as "because" or "but". Previous studies on this task have zeroed in on lexical indicators of relation sense: dependencies between words (Marcu and Echihabi, 2001; Blair-Goldensohn et al., 2007) and the semantic orientation of words (Pitler et al., 2009), or on general syntactic regularities (Lin et al., 2009).

The role of entities has also been hypothesized as important for this task and entity-related features have been used alongside others (Corston-Oliver, 1998; Sporleder and Lascarides, 2008). Corpus studies and reading time experiments performed by Wolf and Gibson (2006) have in fact demonstrated that the type of discourse relation linking two clauses influences the resolution of pronouns in them. However, the predictive power of entity-related features has not been studied independently of other factors. Further motivation for studying this type of features comes from new corpus evidence (Prasad et al., 2008), that *about a quarter of all adjacent sentences are linked purely by entity coherence*, solely because they talk about the same entity. Entity-related features would be expected to better separate out such relations.

We present the first comprehensive study of the connection between entity features and discourse relations. We show that there are notable differences in properties of referring expressions across the different relations. Sense prediction can be done with results better than random baseline using only entity realization information. Their performance, however, is lower than a knowledge-poor approach using only the words in the sentences as features. The addition of entity features to these basic word features is also not beneficial.

2 Data

We use 590 Wall Street Journal (WSJ) articles with overlapping annotations for discourse, coreference and syntax from three corpora.

The Penn Discourse Treebank (PDTB) (Prasad et al., 2008) is the largest available resource of discourse relation annotations. In the PDTB, implicit relations are annotated between adjacent sentences in the same paragraph. They are assigned senses from a hierarchy containing four top level categories–*Comparison*, *Contingency*, *Temporal* and *Expansion*.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 59–62,
The University of Tokyo, September 24-25, 2010.

An example "Contingency" relation is shown below. Here, the second sentence provides the cause for the belief expressed in the first.

Ex 1. These rate indications aren't directly comparable. Lending practices vary widely by location.

Adjacent sentences can also become related solely by talking about a common entity without any of the above discourse relation links between their propositions. Such pairs are annotated as *Entity Relations (EntRels)* in the PDTB, for example:

Ex 2. Rolls-Royce Motor Cars Inc. said it expects its U.S sales to remain steady at about 1,200 cars in 1990. The luxury auto maker last year sold 1,214 cars in the U.S.

We use the coreference annotations from the Ontonotes corpus (version 2.9) (Hovy et al., 2006) to compute our gold-standard entity features. The WSJ portion of this corpus contains 590 articles. Here, nominalizations and temporal expressions are also annotated for coreference but we use the links between noun phrases only. We expect these features computed on the gold-standard annotations to represent an upper bound on the performance of entity features.

Finally, the Penn Treebank corpus (Marcus et al., 1994) is used to obtain gold-standard parse and grammatical role information.

Only adjacent sentences within the same paragraph are used in our experiments.

3 Entity-related features

We associate *each* referring expression in a sentence with a set of attributes as described below. In Section 3.2, we detail how we combine these attributes to compute features for a sentence pair.

3.1 Referring expression attributes

Grammatical role. In exploratory analysis of Comparison relations, we often observed *parallel* syntactic realizations for entities in the subject position of the two sentences:

Ex 3. {Longer maturities}$_{E1}$ are thought to indicate declining interest rates. {Shorter maturities}$_{E2}$ are considered a sign of rising rates because portfolio managers can capture higher rates sooner.

So, for each noun phrase, we record whether it is the subject of a main clause *(msubj)*, subject of other clauses in the sentence *(esubj)* or a noun phrase not in subject position *(other)*.

Given vs. New. When an entity is first introduced in the text, it is considered a *new* entity. Subsequent mentions of the same entity are *given* (Prince, 1992). New-given distinction could help to identify some of the Expansion and Entity relations. When a sentence elaborates on another, it might contain a greater number of new entities.

We use the Ontonotes coreference annotations to mark the information status for entities. For an entity, if an antecedent is found in the previous sentences, it is marked as *given*, otherwise it is a *new* entity.

Syntactic realization. In Entity relations, the second sentence provides more information about a specific entity in the first and a definite description for this second mention seems likely. Also, given the importance of named entities in news, entities with proper names might be the ones frequently described using Entity relations.

We use the part of speech (POS) tag associated with the head of the noun phrase to assign one of the following categories: *pronoun*, *nominal*, *name* or *expletive*. When the head does not belong to the above classes, we simply record its POS tag. We also mark whether the noun phrase is a *definite description* using the presence of the article 'the'.

Modification. We expected modification properties to be most useful for predicting Comparison relations. Also, named or new entities in Entity relations are very likely to have post modification.

We record whether there are premodifiers or postmodifiers in a given referring expression. In the absence of pre- and postmodifiers, we indicate *bare head* realization.

Topicalization. Preposed prepositional or adverbial phrases before the subject of a sentence indicate the topic under which the sentence is framed. We observed that this property is frequent in Comparison and Temporal relations. An example Comparison is shown below.

Ex 4. {*Under British rules*}$_{T1}$, Blue Arrow was able to write off at once $1.15 billion in goodwill arising from the purchase. {*As a US-based company*}$_{T2}$, Blue Arrow would have to amortize the good will over as many as 40 years, creating a continuing drag on reported earnings.

When the left sibling of a referring expression is a topicalized phrase, we mark the *topic* attribute.

Number. Using the POS tag of the head word, we note whether the entity is singular or plural.

3.2 Features for classification

Next, for each *sentence pair*, we associate two sets of features using the attributes described above.

Let $S1$ and $S2$ denote the two adjacent sentences in a relation, where $S1$ occurs first in the text.

Sentence level. These features characterize $S1$ and $S2$ individually. For each sentence, we add a feature for each of the attributes described above. The value of the feature is the number of times that attribute is observed in the sentence; i.e., the feature *S1given* would have a value of 3 if there are 3 *given* entities in the first sentence.

Sentence pair. These features capture the interactions between the entities present in $S1$ and $S2$.

Firstly, for each pair of entities (a, b), such that a appears in $S1$ and b appears in $S2$, we assign one of the following classes: *(i)* SAME: a and b are coreferent, *(ii)* RELATED: their head words are identical, *(iii)* DIFFERENT: neither coreferent nor related. The RELATED category was introduced to capture the parallelism often present in Comparison relations. Even though the entities themselves are not coreferent, they share the same head word (i.e. *longer maturities* and *shorter maturities*).

For features, we use the combination of the class ((i), (ii) or (iii)) with the cross product of the attributes for a and b. For example if a has attributes {*msubj, noun, ...*} and b has attributes {*esubj, defdesc, ...*} and a and b are coreferent, we would increment the count for features– {*sameS1msubjS2esubj, sameS1msubjS2defdesc, sameS1nounS2esubj, sameS1nounS2defdesc ...*}.

Our total set of features observed for instances in the training data is about 2000.

We experimented with two variants of features: one using coreference annotations from the Ontonotes corpus (*gold-standard*) and another based on *approximate* coreference information where entities with identical head words are marked as coreferent.

4 Experimental setup

We define five classification tasks which disambiguate if a specific PDTB relation holds between adjacent sentences. In each task, we classify the relation of interest (*positive*) versus a category with a naturally occurring distribution of *all* of the other relations (*negative*).

Sentence pairs from sections 0 to 22 of WSJ are used as training data and we test on sections 23 and 24. Given the skewed distribution of positive and negative examples for each task, we randomly downsample the negative instances in the training set to be equal to the positive examples. The sizes of training sets for the tasks are

Expansion vs other (4716)
Contingency vs other (2466)
Comparison vs other (1138)
Temporal vs other (474)
EntRel vs other (2378)

Half of these examples are positive and the other negative in each case.

The test set contains 1002 sentence pairs: Comp. (133), Cont. (230), Temp. (34), Expn. (369), EntRel (229), NoRel[1] (7). We do not downsample our test set. Instead, we evaluate our predictions on the natural distribution present in the data to get a realistic estimate of performance.

We train a linear SVM classifier (LIBLINEAR[2]) for each task.[3] The optimum regularization parameter was chosen using cross validation on the training data.

5 Results

5.1 Feature analysis

We ranked the features (based on gold-standard coreference information) in the training sets by their *information gain*. We then checked which attributes are common among the top five features for different classification tasks.

As we had expected, the topicalization attribute and RELATED entities frequently appear among the top features for Comparison.

Features with the *name* attribute were highly predictive of Entity relations as hypothesized. However, while we had expected Entity relations to have a high rate of coreference, we found coreferent mentions to be very indicative of Temporal relations: all the top features involve the SAME attribute. A post-analysis showed that close to 70% of Temporal relations involve coreferent entities compared to around 50% for the other classes.

The number of pronouns in the second sentence was most characteristic of the Contingency relation. In the training set for Contingency task, about 45% of sentences pairs belonging to Contingency relation have a pronoun in the second sentence. This is considerably larger than 32%, which is the percentage of sentence pairs in the negative examples with a pronoun in second sentence.

[1] PDTB relation for sentence pair when both entity and discourse relations are absent, very rare about 1% of our data.

[2] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

[3] SVMs with linear kernel gave the best performance. We also experimented with SVMs with radial basis kernel, Naive Bayes and MaxEnt classifiers.

5.2 Performance on sense prediction

The classification results (fscores) are shown in Table 1. The random baseline (Base.) represents the results if we predicted positive and negative relations according to their proportion in the test set.

Entity features based on both gold-standard (EntGS) and approximate coreference (EntApp) outperform the random baseline for all the tasks. The drop in performance without gold-standard coreference information is strongly noticable only for Expansion relations.

The best improvement from the baseline is seen for predicting Contingency and Entity relations, with around 15% absolute improvement in fscore with both EntGS and EntApp features. The improvements for Comparisons and Expansions are around 11% in the approximate case. Temporal relations benefit least from these features. These relations are rare, comprising 3% of the test set and harder to isolate from other relations. Overall, our results indicate that discourse relations and entity realization have a strong association.

5.3 Comparison with lexical features

In the context of using entity features for sense prediction, one would also like to test how these linguistically rich features compare with simpler knowledge-lean approaches used in prior work.

Specifically, we compare with *word pairs*, a simple yet powerful set of features introduced by Marcu and Echihabi (2001). These features are the cross product of words in the first sentence with those in the second.

We trained classifiers on the word pairs from the sentences in the PDTB training sets. In Table 1, we report the performance of word pairs (WP) as well as their combination with gold-standard entity features (WP+EntGS). Word pairs turn out as stronger predictors for all discourse relations compared to our entity features (except for Expansion prediction with EntGS features). Further, no benefits over word pair results are obtained by combining entity realization information.

Task	Base.	EntGS	EntApp	WP	WP+EntGS
Comp vs Oth.	13.27	24.18	24.14	27.30	26.19
Cont vs Oth.	22.95	37.57	38.16	38.17	38.99
Temp vs Oth.	3.39	7.58	5.61	11.09	10.04
Expn vs Oth.	36.82	52.42	47.82	48.54	49.06
Ent vs Oth.	22.85	38.03	36.73	38.48	38.14

Table 1: Fscore results

6 Conclusion

In this work, we used a task-based approach to show that the two components of coherence—discourse relations and entities—are related and interact with each other. Coreference, givenness, syntactic form and grammatical role of entities can predict the implicit discourse relation between adjacent sentences with results better than random baseline. However, with respect to developing automatic discourse parsers, these entity features are less likely to be useful. They do not outperform or complement simpler lexical features. It would be interesting to explore whether other aspects of entity reference might be useful for this task, such as bridging anaphora. But currently, annotations and tools for these phenomena are not available.

References

S. Blair-Goldensohn, K. McKeown, and O. Rambow. 2007. Building and refining rhetorical-semantic relation models. In *HLT-NAACL*.

S.H. Corston-Oliver. 1998. Beyond string matching and cue phrases: Improving efficiency and coverage in discourse analysis. In *The AAAI Spring Symposium on Intelligent Text Summarization*.

E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. 2006. Ontonotes: the 90% solution. In *NAACL-HLT*.

Z. Lin, M. Kan, and H.T. Ng. 2009. Recognizing implicit discourse relations in the Penn Discourse Treebank. In *EMNLP*.

D. Marcu and A. Echihabi. 2001. An unsupervised approach to recognizing discourse relations. In *ACL*.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*.

E. Pitler, A. Louis, and A. Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *ACL-IJCNLP*.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The penn discourse treebank 2.0. In *LREC*.

E. Prince. 1992. The zpg letter: subject, definiteness, and information status. In *Discourse description: diverse analyses of a fund raising text*, pages 295–325. John Benjamins.

C. Sporleder and A. Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14:369–416.

F. Wolf and E. Gibson. 2006. *Coherence in natural language: data structures and applications*. MIT Press.

Same and *Elaboration* Relations in the Discourse Graphbank

Irina Borisova
University of Groningen,
Groningen, The Netherlands
Saarland University,
Saarbrücken, Germany
borisova.ira@gmail.com

Gisela Redeker
University of Groningen,
Groningen, The Netherlands
g.redeker@rug.nl

Abstract

This study investigates the use of *Same* – a relation that connects the parts of a discontinuous discourse segment – in the Discourse Graphbank (Wolf et al., 2004). Our analysis reveals systematic deviations from the definition of the *Same* relation and a substantial number of confusions between *Same* and *Elaboration* relations. We discuss some methodological and theoretical implications of these findings.

1 Introduction

Coherence relations and their composition (usually assumed to be strictly hierarchical, i.e., treelike) form the core of most corpus-linguistic and computational work on discourse structure (see Taboada & Mann 2006 for an overview). The assumption that discourse structure can be modeled as a tree has recently come under attack e.g. in Wolf & Gibson (2003, 2006; henceforth WG). Based on the *Discourse Graphbank* (Wolf et al 2004; henceforth DG), a manually annotated corpus of 135 newspaper and newswire texts, WG claim that less constrained graph structures are needed that allow for crossed dependencies (i.e. structures in which discourse units ABCD (not necessarily adjacent) have relations AC and BD) and multiple-parent structures (where a unit enters more than one coherence relation and is thus dominated by more than one node).[1]

Among the 11 types of relations distinguished in DG, the *Elaboration* relation, where two asymmetrically related discourse units are "centered around a common event of entity"(Wolf et al 2003: 12), stands out by its heavy involvement in these violations of tree structure constraints. *Elaboration* relations are involved in 50.52% of all crossed dependency structures and in 45.83% of multiple-parent structures. These high percentages are in part due to the high overall frequency of *Elaboration* relations (37.97% of all relations), but clearly exceed that base rate. Elsewhere, *Elaboration* relations, esp. those where the elaborandum is an entity and not a whole proposition, have been criticized as belonging more to referential coherence than to relational coherence (Knott el at 2001). In this study, we show that WG's (somewhat idiosyncratic) definition of the *Elaboration* relation seems to lead to confusion with the 'pseudo-relation' *Same.*

The 'pseudo-relation' *Same-Unit* was introduced by Marcu (Carlson & Marcu 2001) to deal with discontinuous discourse units in the RST Discourse Treebank (Carlson, Marcu & Okurowski 2002). *Same-Unit* (re)connects the parts of a discourse unit that is disrupted by embedded material. In the tree representation, the intervening material is attached to one of the constituent units of the *Same-Unit* relation (Carlson & Marcu 2001:23-26). In DG, this relation is called *Same* and accounts for 17.21% of all relations; only *Elaboration* and *Similarity* are more frequent.[2] As DG allows multiple attachments, *Same* should be expected to be regularly associated with multiple-parent structures, and it is: the percentage of *Same* relations is higher in multiple-parent structures than overall, and the reduction of multiple-

[1] The validity of this claim is contested in Egg & Redeker (2010).

[2] Note that a *Same-Unit* relation is not needed in 'classic' RST, where parenthetical segments are extracted and placed after the segment within which they occur (Redeker & Egg 2006).

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 63–66,
The University of Tokyo, September 24-25, 2010.

parent structures when *Same* relations are removed from the DG is second only to *Elaboration* (Wolf & Gibson 2003:280-282).

Our explorations of *Same* relations in DG revealed a substantial number of cases that do not seem to fit WG's definition of this relation, most notably confusions with *Elaboration* relations and a surprising number of cases where there is no intervening segment to be bridged by the *Same* relation. In this paper, we will present these findings and discuss some consequences for discourse segmentation and the annotation of coherence relations.

2 *Same* relations in DG

The DG coding manual (Wolf et al 2003:15) stipulates as the only condition for a *Same* relation that a discourse segment must have "intervening material". The example in the manual tacitly fits the much more restrictive definition given in (Wolf & Gibson 2003:255) and in (Wolf & Gibson 2006:28):

> "A same relation holds if a subject NP is separated from its predicate by an intervening discourse segment".

Among the 534 *Same* relations in DG,[3] we have identified 128 cases (23.98%) where this definition does not seem to apply. Sixty-four of these cases also do not satisfy the broader definition in the coding manual (see 2.3).

2.1 *Same* or *Elaboration*?

In 35 cases, the *Same* relation is applied to constructions that are elsewhere labeled *Elaborations*. Consider the parallel examples (1) and (2):

```
(1) [42]–[44] elab-loc
[42] There, [43] she said,
[44] robots perform specific
tasks in "islands of
automation," (Text 1)


(2) [32]–[34] same
[32] In the factory of the
future, [33] according to the
university's model, [34]
human chatter will be
replaced by the click-clack
of machines. (Text 1)
```

In these examples, [42] and [32] each specify a location for the state of affairs expressed in the second constituent of the relation, [44] and [34] respectively. Note that [32] is not a subject NP and example (2) thus violates the restricted variant of the *Same* relation definition. Interestingly, examples (1) and (2) differ with respect to the involvement in crossed dependencies and multiple-parent structures. As expected from an elaborating segment, [42] does not participate in any other relations; the three other relations [44] participates in do not include [42]. By contrast, [32] is attached to the intervening segment and in eight other relations in which not [34] by itself, but the combined segment [32]–[34] participates.

In other examples, a general difference between these *Same* and *Elaboration* examples lies in the attachment of the intervening segment: in the *Same* cases, the intervening segment might be attached to the preceding discourse segment, and in the *Elaboration* cases to the following segment.

The confusion between the symmetric *Same* relation (both segments have in principle equal status) and the asymmetric *Elaboration* relation (combining an elaborandum with a less central elaborating segment) might have been caused by WG's definition, which stipulates that the segments be "centered around a common event or entity" (Wolf et al 2003: 12) and thus does not reflect the asymmetry of the *Elaboration* relation.

2.2 Violations of definitional constraints

There are other cases, besides those discussed in 2.1, where the formal requirement of the restrictive definition is not met. In 20 cases, the *Same* relations joins coordinated or disjoint NP's as in example (3):

```
(3) [13]–[16] same
[13] Mrs. Price's husband,
[14] Everett Price, [15] 63,
[16] and their daughters,
(Text 2)
```

In 12 cases, *Same* is used to relate a discourse connective to its host clause as in (4):

```
(4) [4]–[6] same
[4] However, [5] after two
meetings with the Soviets,
[6] a State Department
spokesman said that (Text 8)
```

[3] We have arbitrarily chosen to use the data for annotator 1. The two annotators agreed on segmentation and annotation in 98% of the cases.

Presumably the annotators were using the less restrictive definition in the coding manual. This explanation cannot account for the last category of problematic cases we now turn to.

2.3 Spurious *Same* relations

We found 64 cases in DG where *Same* is assigned to two adjacent discourse segments, thus violating the essential criterion of "intervening material". Such 'spurious' *Same* relations occur with various constructions including the following:

- Complement clauses

(5) [61] The administration should now state [62] that (Text 123, wsj_0655)

- Infinitive clauses

(6) [79] Banco Exterior was one of the last banks [80] to create a brokerage house (Text 122, wsj_0616)

- Conditional clauses

(7) [35] And important U.S. lawmakers must decide at the end of November [36] if the Contras are to receive the rest of the $49 million in so-called humanitarian assistance under a bipartisan agreement (Text 123, wsj_0655).

- Gerund postmodifier phrases

(8) [2] Lawmakers haven't publicly raised the possibility [3] of renewing military aid to the Contras, (Text 123, wsj_0655).

- Temporal "as"-clauses

(9) [31] it came [32] as Nicaragua is under special international scrutiny in anticipation of its planned February elections. (Text 123, wsj_0655)

The 64 spurious *Same* relations are concentrated in only 20 of the 135 texts. Fifty-one of those cases occur in ten texts that were also used in the RST Discourse Treebank. This gives us the interesting opportunity to compare the DG and RST Treebank analyses for these 51 cases. As Table 1 shows, only two of them are labeled *Same-Unit* in the RST Treebank, while 26 (51%) are *Elaboration* relations.

Relations	Frequencies	Percent
Elaboration	26	51.0 %
Attribution	13	25.5 %
Same-Unit	2	3.9 %
Other	10	19.6 %
Total	51	100 %

Table 1: Spurious *Same* relations in DG and relations assigned in the RST Treebank

It is instructive to look at the subtype of *Elaboration* assigned to these cases, which most commonly is the relation *Elaboration-object-attribute-e*. It applies to clausal modifiers, usually postmodifiers of a noun phrase, that express an intrinsic quality of an object. Carlson & Marcu (2001:55) illustrate this relation with the following example:

(10) [Allied Capital is a closed-end management investment company][that will operate as a business development concern.] (wsj_0607)

The constructions with spurious *Same* relations in DG thus often involve restrictive modification, implying a very close tie between the segments involved, possibly prompting the annotators to as it were undo the segmentation.

3 Segmentation rules

Any annotation of discourse relations requires rules for segmenting the text into elementary discourse units. DG follows Carlson & Marcu (2003) in assuming clauses, modifiers and attributions as discourse segments (DSs), but adds some "refinements" (Wolf et al., 2003:8) that may be responsible for some of the problematic cases discussed in section 2.[4] In particular, two of the additional stipulations refer to "elaborations":

[4] A different account of the segmentation is given in (Wolf & Gibson 2006), but the annotation in DG is presumably based on the 2003 manual.

> "Elaborations [..] are separate DSs*: [Mr. Jones,][spokesman for IBM,] [said…]"* (Wolf et al., 2003:8)
>
> "Time-, space-, personal- or detail-elaborations are treated as DSs" (Wolf et al., 2003:9).

This might simply be an unfortunate equivocation, but still is likely to confuse annotators by confounding the segmentation and relation annotation tasks.

4 Conclusions

Our analysis of the *Same* relation in DG has shown systematic deviations from the definition of this (pseudo-)relation and a substantial number of confusions between *Same* and *Elaboration*, both in cases where *Same* cannot apply, as there is no intervening segment, and in cases where both might apply, but parsimony would demand to treat parallel cases equally. Some of the problematic cases may have been caused by the use of relational terminology ("elaboration") in two of the segmentation rules. The problems are not just methodological, though, but may raise questions about the conceptual status of *Elaboration* relations.

The confusion of a bone fide coherence relation with a purely technical construction that serves to recombine the parts of an interrupted segment must be worrisome. More specifically, the comparison with the annotation in the RST Discourse Treebank reveals that many of the 'spurious' *Same* relations in DG are analyzed as *Elaboration-object-attribute-e* relations in the RST Treebank. This is exactly the subcategory of *Elaboration* relations that most clearly operate on the level of entities instead of propositions, and thus arguably might not be proper discourse relations (Knott et al. 2001). This holds a fortiori as Carlson & Marcu's (2001) definition of the *Elaboration-object-attribute-e* relation requires a restrictive modifier construction. The increasing availability of corpora annotated for discourse structure will facilitate the further investigation of these questions.

Acknowledgements

This research was partially supported by a travel grant from the Erasmus Mundus Masters Programme in Language and Communication Technologies to Borisova and by grant 360-70-282 of the Netherlands Organization for Scientific Research (NWO) to Redeker. We would like to thank Robin Cooper and three anonymous reviewers for their very useful comments.

References

Lynn Carlson and Daniel Marcu. 2001. *Discourse Tagging Reference Manual.*

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2002. *RST Discourse Treebank.* Linguistic Data Consortium, Philadelphia.

Markus Egg and Gisela Redeker. 2010. How complex is discourse? *Proceedings of LREC 2010.*

Alistair Knott, Jon Oberlander, Michael O'Donnell, and Chris Mellish. 2001. Beyond *Elaboration*: The interaction of relations and focus in coherent text. In J. Schilperoord T. Sanders and W. Spooren, editors, *Text Representation: Linguistic and Psycholinguistic Aspects*, pp 181–196. Benjamins.

Gisela Redeker and Markus Egg. 2006. Says who? On the treatment of speech attributions in discourse structure. In C. Sidner, J. Harpur, A. Benz, and P. Kühnlein (eds), *Proceedings of the Workshop Constraints in Discourse 2006.* Maynooth: National University of Ireland, pp. 140–146.

Maite Taboada and William C. Mann. 2006. Rhetorical Structure Theory: Looking back and moving ahead. *Discourse Studies, 8* (3), 423–459.

Florian Wolf, Edward Gibson, Amy Fisher, Meredith Knight. 2003. *A Procedure for Collecting a Database of Texts Annotated with Coherence Relations.* Database documentation.

Florian Wolf and Edward Gibson. 2003. Representing discourse coherence: A corpus-based study. *Computational Linguistics, 31*(2):249–287.

Florian Wolf, Edward Gibson, Amy Fisher, and Meredith Knight. 2004. *Discourse Graphbank.* Linguistic Data Consortium, Philadelphia.

Florian Wolf and Edward Gibson. 2006. *Coherence in Natural Language.* MIT Press, Cambridge, MA.

Negotiating causal implicatures

Luciana Benotti
Universidad Nacional de Córdoba
Grupo PLN
Ciudad Universitaria
5000 Córdoba, Argentina
luciana.benotti@gmail.com

Patrick Blackburn
INRIA Nancy Grand-Est
Equipe TALARIS
615, rue du Jardin Botanique
54602 Villers lès Nancy, France
patrick.blackburn@loria.fr

Abstract

In this paper we motivate and describe a dialogue manager which is able to infer and negotiate causal implicatures. A causal implicature is a type of Gricean relation implicature, and the ability to infer them is crucial in situated dialogue. Because situated dialogue interleaves conversational acts and physical acts, the dialogue manager needs to have a grasp on causal implicatures in order not only to decide what physical acts to do next but also to generate causally-aware clarifications.

1 Introduction

In conversation, an *important* part of the content conveyed is not explicitly said, rather it is *implicated*. However, Grice (1975)'s classic concept of *conversational implicature* (CI) is far from fully understood. Traditionally CIs have been classified using the Gricean maxims: there are *relation CIs* (also known as relevance CIs), *quantity CIs*, *quality CIs* and *manner CIs*. In formal pragmatics, the most studied CIs are quantity CIs, probably because they are the ones most obviously amenable to theoretical analysis; see (Geurts, in press) for a survey of the state of the art. Far less studied (and traditionally regarded as somewhat obscure) are relation CIs. Obscure perhaps, but crucial: it has been argued that they subsume all other types of CIs (Wilson and Sperber, 2004). This paper is a first step towards their formalization.

We shall analyze a kind of CI that we call *causal CIs*. Causal CIs are relation CIs as defined by Grice (1975) where the crucial relation is *task domain causality*. Consider the following example:

Mary: The chest is locked, the crown is inside
Bill: Give me the crown
Bill causally implicated: Unlock the chest

In order to carry out the task action required by Bill (to give him the crown) it is necessary to unlock the chest. Hence we say that Bill is implicating, by trading on the domain causal relations (after all, the contents of a chest are not accessible unless the chest is unlock) that Mary is to unlock the chest. Now, once Mary has inferred the causal CI, she may accept this inference silently or negotiate it. Mary might decide to silently accept it because she knows how to get the key; in this case we will say that Mary constructed an *internal bridge* from the current task situation (that is, the crown being inside the locked chest) to the proposal made by Bill (giving him the crown). If Mary decides she has insufficient information to construct the internal bridge (maybe she has no key, or sees that the lock is rusty) she may start a sub-dialogue that we will call an *external bridge*; she might say, for example: *But how can I unlock the chest?* The *internal process of bridging* is what in the literature has been called accommodation (Lewis, 1979) or bridging (Clark, 1975). The *external processes of bridging* constitutes a large part of what we call conversation.

This paper presents a dialogue system (called Frolog) which infers and negotiates *causal* CIs in the context of situated task-oriented dialogue; the framework is intended as a proof-of-concept of the ideas just sketched. We proceed as follows. In Section 2, we motivate the study of causal CIs in dialogue. In Section 3 we present Frolog's dialogue manager which infers causal CIs in situated dialogue. And in Section 4 we illustrate how the negotiation (external bridging) of causal CIs incrementally grounds a pragmatic goal proposed by one of the dialogue participants. Section 5 concludes the paper.

2 Causal implicatures and dialogue

The motivation for our work is both theoretical and practical. On the theoretical side, we believe

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 67–70,
The University of Tokyo, September 24-25, 2010.

that it is crucial to explore CIs in the setting of naturally occurring dialogues. Strangely enough (after all, Grice did call them *conversational* implicatures) this view appears to be novel, perhaps even controversial. In the formal pragmatics literature, CIs are often simply viewed as inferences drawn by a hearer on the basis of a speaker's utterance, contextual information, and the Gricean maxims. We find this perspective too static. CIs (especially relations CIs) are better viewed as intrinsically *interactional* inferences that arise from the dynamics of conversation. As conversations progress, speakers and hearers switch roles: meaning are negotiated and inference becomes bidirectional (Thomason et al., 2006). Moreover, even within a single turn, hearers are not restricted to simply drawing (or failing to draw) "the" CI: in fact, choosing between internal and external bridging is better viewed as part of the process of *negotiating what the CI at stake actually is*. We believe that interactive perspectives will be necessary to extend the theory of CIs beyond the relatively narrow domain of quantity CIs. We also believe that the dialog-centered approach we advocate may have practical consequences. In particular, modeling the *external process of bridging* is a step towards having a pragmatically incremental dialogue manager in the spirit of that sketched in (Buß and Schlangen, 2010).

This is a broad goal, in this paper we focus on clausal implicatures. This restriction gives us an *empirical* handle of CIs. It is not controversial that (in non-conversational activities) the causal relations between acts define the expectations of the interaction. But also in conversational activities situated in a physical task causal relations guide the interaction; we did an empirical study on such a kind of corpus (Benotti, 2009) and we found that, in this corpus, most CIs for which there is evidence (because they are made explicit in a clarification request) can be explained in terms of causal relations. For our empirical study, we annotated and classified the clarification requests (CRs) that appear in the SCARE corpus (Stoia et al., 2008).

3 Inferring causal implicatures

In order to model the causal CIs that we observed in the SCARE corpus, and to experiment with different strategies for negotiating these CIs, we designed a system that mimics the instruction giving setup of the SCARE corpus. In our setup, the DF is a dialogue system that we will call Frolog. The human participant that plays the role of the DG we will call "the player".

In a nutshell, Frolog uses an off-the-shelf planner to compute causal implicatures. That is, it uses classical planning (a well explored and computationally efficient AI technique) to fill out the micro-structure of discourse (the bridging information required in the next step).[1] We do so using the planner BLACKBOX (Kautz and Selman, 1999). Like all classical planners, BLACKBOX takes three inputs: the initial state, the goal, and the available actions. The question of what these three elements should be raises a number of issues.

In Frolog, two types of information are registered: complete and accurate information about the game world in the *world KB* and a representation of the common ground in the *interaction KB*. Which of these should be used in the initial state? In fact, we need both: we infer the actions intended by the player using the information in the interaction KB but we have to verify this sequence of actions on the world KB to check if it can actually be executed.

Let us now define what the goal of the planning problem should be. Frolog should act to make the preconditions of the action true with one restriction. The restriction is that it must be possible for Frolog to manipulate these preconditions. However, we don't need to worry about this restriction because the planner should take care of which propositions are manipulable by Frolog and which are not, given the current state. So we can just define the goal as the conjunction of all the preconditions of the command uttered by the player.

To complete the picture, the actions available to the planner are all the actions in the game action database. This means that we are assuming that all the actions that can be executed, are mutually known to Frolog and the player.

In order to be able to perform bridging to the mutual information it must be mutually known what the preconditions and the effects of the actions involved are. The assumption that the player and Frolog know the exact specification of all the actions that can be executed in the game world is

[1] Thus the work reported here is very different from the traditional work of (Perrault and Allen, 1980; Allen and Allen, 1994): classic papers in the plan-based tradition use *plan recognition* (a more computationally expensive task) to interpret utterances by inserting them into the plan the macro-structure (the global shape) of discourse.

a simplifying assumption. We make it because it enables us to avoid deciding (and implementing) how differing knowledge about actions get coordinated; dealing with such issues simply too difficult at present.

4 Negotiating with a dialogue system

In this section we present a worked out example of how the solutions of the planning problems introduced in the previous section can be used in order to perform internal bridging and trigger external bridging when appropriate.

In this example Frolog does not have enough knowledge yet in order to infer the bridge.[2] This is the case in instruction (37) in Figure 1. In this instruction the player wants Frolog to open the chest but Frolog does not know how to do this. In the screen-shot in the right you can see state of the interaction KB from turns (37) to (39). In this model we can see that the player does not have information about what fits into the chest, in order to unlock it. This is why the planner is not able to find a plan that achieves the precondition `(unlocked chest1)` of the command "open the chest" in instruction (37). The instruction in (39) fails as well making the obstacle explicit. Utterance (40) is an example of a CR caused by a required and unknown thematic relation filler (the second most frequent CR in the SCARE corpus).

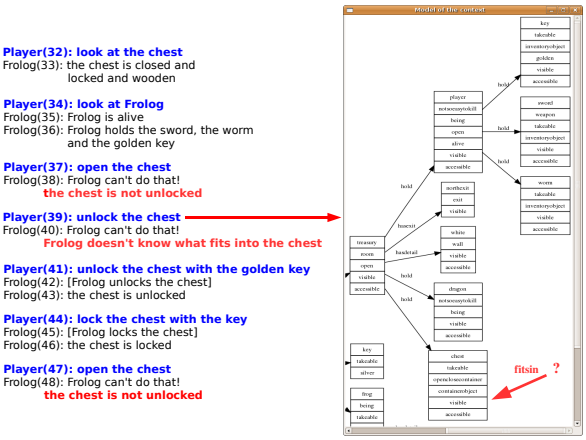


Figure 1: *External* bridging in Frolog

Frolog's behavior continues as expected until turn (48) in which Frolog again is not able to bridge the command open the chest. It is to be expected that the fact that the golden key fits into the chest is mutual knowledge by now but Frolog is not reacting accordingly.

In order to overcome this issue there are two options: (1) either we allow different sources to update the mutual information loosing uniformity in the update process or (2) we add actions that manipulate this kind of information. Here we describe option (1) that turns out to be the procedural version of "strict accommodation" (as defined in (Beaver and Geurts, in press)), and when generalized is of a piece with abduction (Hobbs, 1985). For an discussion of (2) we refer the reader to (Benotti, 2010).

We could say that the intuition behind the strict notion of accommodation is that when the speaker utters a sentence, she is not only trying to achieve the obvious effects of the utterance, but is also communicating the ways in which she assumes the world to be, and on which the success of the utterance depends. Following this intuition it can be argued that the intention behind the utterance "unlock the chest with the golden key" in turn (41) is twofold. It is clear that the player wants to achieve the effects of the action, that is, she wants to have the chest unlocked. But the player also expects Frolog to recognize the assumptions she is making and on which the success of the utterance depends. In particular, she expects Frolog to recognize that she is assuming that the golden key fits into the chest lock (this is why Frolog can coherently ask *why do you think that the golden key will fit?*).

This means that, when an action is executed, the interaction KB will be updated not only with the effects of the action but also with its preconditions. And those preconditions that were not in the interaction KB before will be verbalized as in turn (9) in Figure 2.

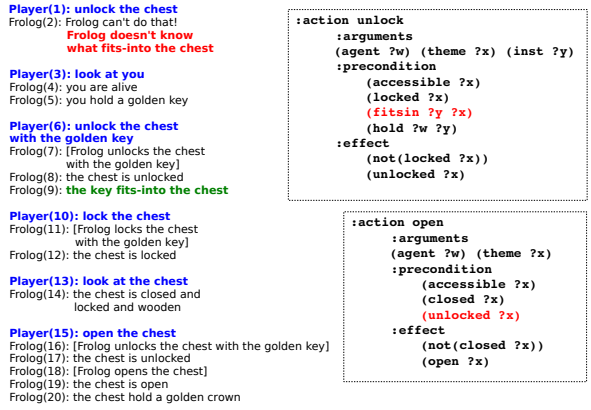


Figure 2: *External* bridging becomes *internal*

[2] For illustrations of other reasons that trigger negotiation of CIs see (Benotti, 2010).

The rest of the interaction (from turns (10) to (15)) show that once the proposition `(fitsin key1 chest1)` is added to the interaction KB the action "open the chest" can be internally bridged even when the chest is locked. Because the player and Frolog mutually know which key fits into the chest.

5 Discussion

Clearly, our inference framework is limited in many ways. But we think we've made a small step in the right direction. Dialogue systems are reaching a development level in which they cannot elude drawing inferences for much longer. This paper is a step in this direction.

Causal implicatures are a kind of relation implicature (historically Grice's most obscure and crucial implicature) whose inference—we've argued—is essential in situated dialogue if our dialogue systems are not to violate the expectations of the user. Causal relations have a direct impact on the coherence structure of situated dialogues such as those in the SCARE corpus; in the SCARE corpus most pragmatic clarification requests make explicit causal implicatures.

We need to have a grasp on causal implicatures in order for our dialogue systems not only to decide what physical acts to do next—internal bridging—but also to generate causally-aware clarification requests—external bridging. Of course the inference framework presented here has many limitations that we discussed throughout the paper and probably classical planning is not the formalism that we will finally want to use in our dialogue systems (at least not in its present form). Our model is intended as a proof of concept, and intentionally stays at a level of formalization that is still simple enough so as not to loose our intuitions. The two intuitions that we don't want to loose sight of are (1) utterances are to be interpreted *in a context* and need to be connected to this context (through some kind of relation, being causality one of the most important ones in situated dialogue) in order to be grounded (2) the process of connecting utterances to the context is a joint process, it is a negotiation that involves decisions of all the dialogue participants.

With the intuitions in place we plan to extend this work mainly by porting the inference framework into new domains.

There is lot to do yet, but we believe that the negotiation of causal implicatures is a step towards an incremental dialogue manager.

References

James Allen and Richard Allen. 1994. *Natural language understanding*. Addison Wesley, 2nd edition.

David Beaver and Bart Geurts. in press. Presupposition. In *Handbook of Semantics*. Mouton de Gruyter.

Luciana Benotti. 2009. Clarification potential of instructions. In *Proc. of SIGDIAL*, pages 196–205, London, United Kingdom.

Luciana Benotti. 2010. *Implicature as an Interactive Process*. Ph.D. thesis, Université Henri Poincaré, INRIA Nancy Grand Est, France. Supervised by P. Blackburn. Reviewed by N. Asher and B. Geurts.

Okko Buß and David Schlangen. 2010. Modelling sub-utterance phenomena in spoken dialogue systems. In *The 2010 Workshop on the Semantics and Pragmatics of Dialogue*, Poznań, Poland.

Herbert Clark. 1975. Bridging. In *Proc. of the Workshop on Theoretical issues in natural language processing*, pages 169–174, Morristown, USA. ACL.

Bart Geurts. in press. *Quantity implicatures*. Cambridge University Press.

Paul Grice. 1975. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics*, volume 3, pages 41–58. Academic Press, New York.

Jerry Hobbs. 1985. Granularity. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 432–435. Morgan Kaufmann.

Henry Kautz and Bart Selman. 1999. Unifying SAT-based and graph-based planning. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 318–325, Stockholm, Sweden.

David Lewis. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8:339–359.

Raymond Perrault and James Allen. 1980. A plan-based analysis of indirect speech acts. *Computational Linguistics*, 6(3-4):167–182.

Laura Stoia, Darla Shockley, Donna Byron, and Eric Fosler-Lussier. 2008. SCARE: A situated corpus with annotated referring expressions. In *Proc. of LREC*.

Richmond Thomason, Matthew Stone, and David DeVault. 2006. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In *Presupposition Accommodation*. Ohio State Pragmatics Initiative.

Deirdre Wilson and Dan Sperber. 2004. Relevance theory. In *Handbook of Pragmatics*, pages 607–632. Blackwell, Oxford.

Presupposition Accommodation as Exception Handling

Philippe de Groote
INRIA Nancy - Grand Est
Philippe.de.Groote@loria.fr

Ekaterina Lebedeva
INRIA Nancy - Grand Est
UHP Nancy 1
ekaterina.lebedeva@loria.fr

Abstract

Van der Sandt's algorithm for handling presupposition is based on a "presupposition as anaphora" paradigm and is expressed in the realm of Kamp's DRT. In recent years, we have proposed a type-theoretic rebuilding of DRT that allows Montague's semantics to be combined with discourse dynamics. Here we explore van der Sandt's theory along the line of this formal framework. It then results that presupposition handling may be expressed in a purely Montagovian setting, and that presupposition accommodation amounts to exception handling.

1 Introduction

Montague (1970) argued that there is no essential difference between natural and mathematical languages. He developed a theory that assigns a lambda-term for each lexical item, and the meaning of a whole sentence could be obtained by composing the lambda-terms via functional application. However, his theory was limited to single sentences. De Groote (2006) extends Montague's framework with a continuation-passing-style technique, developing a framework that is dynamic in a sense reminiscent of Dynamic Predicate Logic (Groenendijk and Stokhof, 1991).

While Montague's semantics is based on Church's (1940) simple type theory and has only two atomic types (ι, the type of individuals; and o, the type of propositions), de Groote (2006) adds an atomic type γ representing the type of the *environment*. For each lambda-term the continuation is what is still to be processed, and its type is $\gamma \to o$.

Since anaphoric expressions are known to be similar to presuppositional expressions (van der Sandt, 1992), it is natural to ask whether our type-theoretic framework can be extended to handle presuppositions. The goal of this paper is to answer this question positively, at least in the case of presuppositions triggered by definite descriptions. To achieve this goal γ will not be defined simply as a list of individuals, but as a list of individuals together with their properties.

2 Background

Van der Sandt (1992) argues that presuppositions and anaphors display similar behavior: they primarily have to be bound to some antecedent previously introduced in the discourse. Therefore, they can be treated by similar mechanisms. He implements his ideas in DRT (Kamp and Reyle, 1993) in such a way that for each new sentence a provisional DRS encoding possible anaphoric elements is constructed. This provisional DRS is then merged with the main DRS, and the presuppositional anaphors are resolved in accordance with certain pragmatic constraints, so that presuppositions can be accommodated when lacking a suitable antecedent.

Geurts (1999) proposes an improvement of van der Sandt's theory, called *the binding theory*, according to which anaphora is a kind of presupposition. Therefore, presuppositions triggered by pronouns and definite descriptions can also be accommodated: a referent is introduced with a poor descriptive content and the descriptive content can be enhanced as the discourse unfolds. Moreover, according to the presuppositional version of the *quotation theory of names* (Kneale, 1962), names (e.g. *John*) are synonymous with definite noun phrases of the form "the individual named John". Hence, presuppositions triggered by names and by definite descriptions can be handled similarly.

De Groote's (2006) dynamic theory provides some improvement over classical DRT. It allows the representations of sentence and discourse to be built from the lexical items in the spirit of Montague. It provides reference marker renaming for

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 71–74,
The University of Tokyo, September 24-25, 2010.

free and may be implemented using well established techniques. We claim that Geurts' binding theory can be incorporated into this framework, providing a fully compositional treatment of definite descriptions.

3 Presupposition in Dynamic Theory

We focus here on presuppositions triggered by definite descriptions, particularly by proper names, pronouns and possessive noun phrases.

3.1 Basic Principles

Imagine that somebody is about to tell a new story and the first sentence of this story is (1).

This story is about John. (1)

If the listener does not know John, he or she will immediately imagine a person named "John" and memorize it. In other words, the listener will *accommodate* the presuppositional content triggered by the proper name *John* in the following way: he or she will *create* a slot in the *environment*, which is some unit representing the knowledge about *John*, and put there what was just learned about *John*. Therefore, the listener will be able to refer to the created slot representing *John* as the discourse evolves. Moreover, the slot for *John* will be different from other slots, i.e. it will have some identity marker, which we call, following Karttunen (1976), *reference marker* or simply *referent*. There is a direct analogy between memory slots introduced above and Heim's (1982; 1983) file cards: they are both aimed to store what has been learned about some individual.

Let j be the referent for *John* and assume that sentence (1) is followed by sentence (2).

John loves Mary. (2)

Mary is a new individual in the discourse and therefore *Mary* will be accommodated introducing a reference marker m exactly as it happened for *John* after the utterance of (1). The story is different for *John* now. The listener already has a representation standing for *John* in the environment, and he or she just has to turn to the corresponding slot (*select* the marker in the environment) and update the slot with the new information that John loves Mary (*bind John* from (2) to the referent j).

3.2 Proper Names

To encode, following Montague's legacy, the observations discussed above as lambda-terms, we first define a selection function *sel* as a function taking two arguments: a property and an environment; and returning a reference marker:

$$sel : (\iota \to o) \to \gamma \to \iota \tag{3}$$

According to Montague, proper names can be interpreted as type-raised individuals, thus the lambda-term standing for *John* in Montague's semantics is (4), where **j** is a constant.

$$[\![John]\!] = \lambda P.P\mathbf{j} \tag{4}$$

In the dynamic interpretation, instead of the constant **j** we would like to have a referent corresponding to *John*. For this, we attempt to select such a referent given a property of being named John, as shown in (5).

$$[\![John]\!] = \lambda P.P(sel(\text{named "John"})) \tag{5}$$

Whether the selection of the marker for *John* succeeds depends on the current environment. Hence, instead of using Montague's individuals (i.e. of type ι) directly, we use individuals parameterized by the environment (i.e. having type $(\gamma \to \iota)$).

Noun phrases are regarded as having type (6), which is analogous to the type for noun phrases (7) given by Montague, i.e. a noun phrase is interpreted by a lambda-term that accepts a property and returns a proposition. The only difference is that now individuals are always parameterized by an environment, and propositions are dynamic[1], i.e. they have type Ω that is defined as $\gamma \to (\gamma \to o) \to o$.

$$[\![NP]\!] = ((\gamma \to \iota) \to \Omega) \to \Omega \tag{6}$$

$$[\![NP]\!] = \quad (\iota \quad \to o) \quad \to o \tag{7}$$

3.3 Pronouns

Pronouns are also presupposition triggers. It can be seen in the case of cataphora, such as, for example, in sentence (8), where in the first part of the sentence the pronoun *he* introduces an individual. Since pronouns have poorer descriptive content than proper names and they have the type of noun phrases (6), they are represented by lambda-terms that are at most as complex as the terms for proper names. The term for the pronoun *he* is shown in (9), which expresses an attempt to select a human individual having masculine gender.

When he woke up, Tom felt better. (8)

[1] Analogously, dynamic predicates take two additional arguments (environment, of type γ, and continuation, of type $(\gamma \to o)$) compared to Montague's interpretation.

$$[\![he]\!] = \lambda P.P(sel\ (\lambda x.\textbf{human}(x) \wedge \textbf{masculine}(x))) \quad (9)$$

If the sentence (8) is uttered in a discourse that does not provide a suitable referent, the presupposition trigerred by *he* will be accommodated (as it happened for *John* in (1) and for *Mary* in (2)). The presuppositional anaphora trigerred by *Tom* in the second part of the sentence could be successfully bound to the introduced referent.

3.4 Possessives

Consider the sentence (10), where we have a possessive noun phrase *John's car* triggering a presupposition that there is a car owned by John.

$$John's\ car\ is\ red. \quad (10)$$

The desired interpretation of *John's car* is shown in (11), which requires a search in the environment for a referent having the property of being a car possessed by John. The embedded presupposition is encoded via a selection function (for the inner presupposition triggered by *John*) embedded into another selection function (for the outer presupposition related to *car*).

$$[\![John's\ car]\!] = \lambda P.P(\lambda e.sel(\lambda x.\textbf{car}x \wedge \textbf{poss}\ x\ sel(\text{named "John"})e)e) \quad (11)$$

However, we would like to express *John's car* compositionally in terms of its constituents. To do so, we define a term (12) taking two arguments - a noun phrase standing for a possessor and a noun standing for an object being possessed, and returning a noun phrase in form of (11). $\wedge$ is a dynamic conjunction having type (13) and defined in (14).

$$[\!['s]\!] = \lambda YX.\lambda P.P(SEL(\lambda x.((Xx) \wedge Y([\![poss]\!]x)))) \quad (12)$$

$$\wedge : \Omega \to (\Omega \to \Omega) \quad (13)$$

$$A \wedge B = \lambda e\phi.Ae(\lambda e.Be\phi) \quad (14)$$

The term [[poss]] in (12) is a usual dynamic two-arguments predicate, its lambda-term is shown in (15). *SEL* is a higher-order selection function. It has the same designation as (3), with the only difference that it functions on the level of dynamic propositions. Thus, the type of *SEL* is (16) and it is analogous to the type of *sel* spelled in (3). Moreover, *SEL* is defined via *sel*, and the corresponding lambda-term is presented in (17).

$$[\![poss]\!] = \lambda xy.\lambda e\phi.\textbf{poss}(xe)(ye) \wedge \phi e \quad (15)$$

$$SEL : ((\gamma \to \iota) \to \Omega) \to \gamma \to (\gamma \to \iota) \quad (16)$$

$$SEL = \lambda Pe.sel(\lambda x.P(\lambda e.x)e(\lambda e.\top))e \quad (17)$$

$$[\![car]\!] = \lambda x.\lambda e\phi.\textbf{car}(xe) \wedge \phi e \quad (18)$$

If we apply the term [[$'s$]] to the term (5) for *John* and the term (18) for *car*, which is just a dynamic unary predicate, we will get the desired result (11).

3.5 Implicit Referents

Sometimes an anaphora wants to be bound, even though no referent was introduced explicitly, as in (19). Already after the first sentence, a listener will learn that John has a wife, i.e. introduce a new referent. The presuppositional anaphora triggerred by the possessive noun phrase *his wife* in the second sentence will be bound to this referent.

$$John\ is\ married.\ His\ wife\ is\ beautiful. \quad (19)$$

This case can be accounted with the lexical interpretation in (20) for *being married*, which is defined by a two-arguments relation **is_married**. The first argument of the relation is the argument x being passed to the lexical interpretation. The second argument is an individual selected from the environment given the property of being either the wife or the husband of x.

$$[\![is\_married]\!] = \lambda x.\lambda e\phi.\textbf{is\_married}(xe)(sel(\lambda y.(\textbf{wife}(y,x) \vee \textbf{husband}(y,x)))e) \wedge \phi e \quad (20)$$

3.6 Discourse Update

A discourse is updated by appending the next sentence, as shown in equation (21). A sentence is defined as a term having the type of a dynamic proposition, i.e. its type is (22), while a discourse is defined as a term having the type of a dynamic proposition evaluated over the environment, i.e its type is (23). A discourse $\mathbb{D}$ updated with a sentence $\mathbb{S}$ results in a term having type (23), thus it has one parameter ϕ of type $(\gamma \to o)$. The body must be a term, of type o, contributed by $\mathbb{D}$. $\mathbb{D}$ itself is a term of type (23). Therefore, it must be given a continuation as an argument constructed with $\mathbb{S}$ and its continuation.

$$\mathbb{D} \cup \mathbb{S} = \lambda\phi.\mathbb{D}(\lambda e.\mathbb{S}e\phi) \quad (21)$$

$$[\![S]\!] = \Omega = \gamma \to (\gamma \to o) \to o \quad (22)$$

$$[\![D]\!] = (\gamma \to o) \to o \quad (23)$$

However, during the computation of $\lambda\phi.\mathbb{D}(\lambda e.\mathbb{S}e\phi)$ one of the selection functions can raise an exception containing a message that a referent having some property Q was not found in the environment. The exception will be catched and the property will be returned to the exception

handler. The handler will have to introduce a referent having the property Q into the representation of the discourse, add this referent to the environment, and call the update function passing to it the amended interpretation of the discourse and the sentence $\mathbb{S}$ as parameters. This can be encoded using an exception handling mechanism as shown in (24) for global accommodation. Note that the definition of discourse update is recursive.

$$\begin{aligned}\mathbb{D} \cup \mathbb{S} \quad &= \lambda\phi.\, \mathbb{D}(\lambda e.\mathbb{S}e\phi) \\ &\texttt{handle } (\mathit{fail}\ Q) \texttt{ with} \\ &\lambda\phi.\mathbb{D}(\lambda e.\exists x.(Qx) \wedge \phi((x, Qx) :: e)) \cup \mathbb{S}\end{aligned} \tag{24}$$

The environment is defined as a list of pairs "referent × proposition" (25). The two-place list constructor :: appends a referent together with the corresponding propositions into the environment, therefore it has the type shown in (26).

$$\gamma = \texttt{ list of } (\iota \times o) \tag{25}$$

$$:: : (\iota \times o) \to \gamma \to \gamma \tag{26}$$

The selection function *sel* can implement any anaphora resolution algorithm, and hence our framework is not confined to any of them.

Considering that the lambda-term for *Mary* is similar to (5) and the lambda-term for the transitive verb *love* is (27), the interpretation for the sentence (2) after beta-reductions will be (28).

$$[\![\mathit{love}]\!] = \lambda YX.X(\lambda x.Y(\lambda y.(\lambda e\phi.\mathbf{love}(xe)(ye) \wedge \phi e))) \tag{27}$$

$$\begin{aligned}&\mathbb{S}_2 = [\![\mathit{love}]\!][\![\mathit{John}]\!][\![\mathit{Mary}]\!] \to^*_\beta \\ &\lambda e\phi.(\mathbf{love}(\mathit{sel}(\text{named "John"})e) \\ &\qquad (\mathit{sel}(\text{named "Mary"})e)) \wedge \phi e\end{aligned} \tag{28}$$

After the sentence (1), the lambda-term representing discourse will be (29).

$$\begin{aligned}\mathbb{D}_1 = \quad &\lambda\phi.\exists y.(\mathbf{story}\ y) \wedge \\ &\exists j.(\text{named "John"}\, j) \wedge \\ &\mathbf{about}\ (y, j) \wedge \\ &\phi((y, \mathbf{story}\ y) :: (j, \text{named "John"}\, j))\end{aligned} \tag{29}$$

After the sentence (2), the lambda-term $\mathbb{D}_1$ in (29) will have to be updated with the term $\mathbb{S}_2$ in (28) as it is defined by the function (24). Since we have a referent for *John* in the environment of $\mathbb{D}_1$, it will be successfuly selected and *John* from $\mathbb{S}_2$ will get bound to it. However, there will be a failure for *Mary*, particularly on the property (named "Mary") since there is no corresponding referent in $\mathbb{D}_1$ yet. The failure will be handled by accommodating *Mary* and introducing the sentence $\mathbb{S}_2$ into the amended interpretation of the discourse, which results in the term shown in (30).

$$\begin{aligned}\mathbb{D}_2 = \mathbb{D}_1 \cup \mathbb{S}_2 = \lambda\phi.&\exists y.(\mathbf{story}\ y) \wedge \\ &\exists j.(\text{named "John"}\, j) \wedge \\ &\mathbf{about}\ (y, j) \wedge \\ &\exists m.(\text{named "Mary"}\, j) \wedge \\ &\mathbf{love}\ (j, m) \wedge \\ &\phi((m, \text{named "Mary"}m) :: \\ &\quad (y, \mathbf{story}\ y) :: \\ &\quad (j, \text{named "John"}\, j))\end{aligned} \tag{30}$$

4 Conclusions

We showed that de Groote's (2006) dynamic framework can be applied to presuppositions triggered by definite descriptions, such as proper names, possessive noun phrases and pronouns; and that the exception handling mechanisms offer a proper way of modeling the dynamics of presupposition. Other presuppositional expressions, such as, for example, factives and aspectual verbs, will require more technicalities. Nevertheless, we believe that the approach can be extended to encompass a general theory of presupposition and we intend to address this in future work.

Acknowledgements: We thank the anonymous reviewers for their useful comments.

References

Church, A. (1940). A formulation of the simple theory of types. *Journal of Symbolic Logic*, (5):56–68.

de Groote, P. (2006). Towards a montagovian account of dynamics. In *Semantics and Linguistic Theory XVI*.

Geurts, B. (1999). *Presuppositions and Pronouns*, volume 3 of *CRiSPI*. Elsevier, Amsterdam.

Groenendijk, J. and Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, 14(1):39–100.

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, University of Massachusetts at Amherst.

Heim, I. (1983). On the projection problem for presuppositions. In Barlow, M., Flickinger, D., and Westcoat, M., editors, *Second Annual West Coast Conference on Formal Linguistics*, pages 114–126. Stanford University.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Studies in Linguistics and Philosophy. Springer.

Karttunen, L. (1976). Discourse referents. In McCawley, J., editor, *Syntax and Semantics 2: Notes From the Linguistic Underground*, pages 363–385. Academic Press, New York.

Kneale, W. (1962). Modality de dicto and de re. In Nagel, E., Suppes, P., and Tarski, A., editors, *Logic, methodology and philosophy of science. Proceedings of the 1960 International Congress*, pages 622–633. Stanford University Press.

Montague, R. (1970). Universal grammar. In *Theoria*, pages 373–398.

van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics*, 9:333–377.

Exploring the Effectiveness of Lexical Ontologies for Modeling Temporal Relations with Markov Logic

Eun Y. Ha, Alok Baikadi, Carlyle J. Licata, Bradford W. Mott, James C. Lester
Department of Computer Science
North Carolina State University
Raleigh, NC, USA
{eha,abaikad,cjlicata,bwmott,lester}@ncsu.edu

Abstract

Temporal analysis of events is a central problem in computational models of discourse. However, correctly recognizing temporal aspects of events poses serious challenges. This paper introduces a joint modeling framework and feature set for temporal analysis of events that utilizes Markov Logic. The feature set includes novel features derived from lexical ontologies. An evaluation suggests that introducing lexical relation features improves the overall accuracy of temporal relation models.

1 Introduction

Reasoning about the temporal aspects of events is a critical task in discourse understanding. Temporal analysis techniques contribute to a broad range of applications including question answering and document summarization, but temporal reasoning is complex. A recent series of shared task evaluation challenges proposed a framework with standardized sets of temporal analysis tasks, including identifying the temporal entities mentioned in text, such as events and time expressions, as well as identifying the temporal relations that hold between those temporal entities (Pustejovsky and Verhagen, 2009).

Our previous work (Ha et al., 2010) addressed modeling temporal relations between temporal entities and proposed a supervised machine-learning approach with *Markov Logic* (ML) (Richardson and Domingos, 2006). As novel features, we introduced two types of lexical relations derived from VerbOcean (Chklovski and Pantel, 2004) and WordNet (Fellbaum, 1998). A preliminary evaluation showed the effectiveness of our approach. In this paper, we extend our previous work and conduct a more rigorous evaluation, focusing on the impact of joint optimization of the features and the effectiveness of the lexical relation features for modeling temporal relations.

2 Related Work

Recently, data-driven approaches to modeling temporal relations for written text have been gaining momentum. Boguraev and Ando (2005) apply a semi-supervised learning technique to recognize events and to infer temporal relations between time expressions and their anchored events. Mani et al. (2006) model temporal relations between events as well as between events and time expressions using maximum entropy classifiers. The participants of TempEval-1 investigate a variety of techniques for temporal analysis of text (Verhagen et al., 2007).

While most data-driven techniques model temporal relations as local pairwise classifiers, this approach has the limitation that there is no systematic mechanism to ensure global consistencies among predicted temporal relations (e.g., if event *A* happens before event *B* and event *B* happens before event *C*, then *A* should happen before *C*). To avoid this drawback, a line of research has explored techniques for the global optimization of local classifier decisions. Chambers and Jurafsky (2008) add global constraints over local classifiers using Integer Linear Programming. Yoshikawa et al. (2009) jointly model related temporal classification tasks using ML. These approaches are shown to improve the accuracy of temporal relation models.

Our work is most closely related to Yoshikawa et al. (2009) in that ML is used for joint model-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 75–78,
The University of Tokyo, September 24-25, 2010.

ing of temporal relations. We extend their work in three primary respects. First, we introduce new lexical relation features. Second, our model addresses a new task introduced in TempEval-2. Third, we employ phrase-based syntactic features (Bethard and Martin 2007) rather than dependency-based syntactic features.

3 Data and Tasks

We use the TempEval-2 data for English for both training and testing of our temporal relation models. The data includes 162 news articles (totaling about 53,000 tokens) as the training set and another 11 news articles as the test set. The corpus is labeled with events, time expressions, and temporal relations. Each labeled event and time expression is further annotated with semantic and syntactic attributes. Six types of temporal relations are considered: *before*, *after*, *overlap*, *before-or-overlap*, *overlap-or-after*, and *vague*.

Consider the following example from the TempEval-2 data, marked up with a time expression t_1 and three events e_1, e_2, and e_3, where e_1 and e_2 are the main events of the first and the second sentences, respectively, and e_3 is syntactically dominated by e_2.

```
But a [minute and a half]t1
later, a pilot from a nearby
flight [calls]e1 in. Ah, we
just [saw]e2 an [explosion]e3
up ahead of us here about
sixteen thousand feet or
something like that.
```

In the first sentence, t_1 and e_1 are linked by a temporal relation *overlap*. Temporal relation *after* holds between the two consecutive main events: e_1 occurs *after* e_2. The main event e_2 of the second sentence *overlaps* with e_3, which is syntactically dominated by e_2.

In this paper, we focus on three subproblems of the temporal relation identification task as defined by TempEval-2: identifying temporal relations between (1) events and time expressions in the same sentence (*ET*); (2) two main events in consecutive sentences (*MM*); and (3) two events in the same sentence when one syntactically dominates another (*MS*), which is a new task introduced in TempEval-2.

4 Features

Surface features include the word tokens and stems of the words. In the TempEval-2 data, an event always consists of a single word token, but time expressions often consist of multiple tokens. We treat the entire string of words in a given time expression as a single feature.

Semantic features are the semantic attributes of individual events and time expressions described in Section 3. In this work, we use the gold-standard values for these features that were manually assigned by human annotators in the training and the test data.

Syntactic features include three features adopted from Bethard and Martin (2007): *gov-prep*, any prepositions governing the event or time expression (e.g., *'for'* in '*for ten years'*); *gov-verb*, the verb governing the event or time expression; *gov-verb-pos*, the part-of-speech (pos) tag of the governing verb. We also consider the pos tag of the word in the event and the time expression.

Lexical relations are the semantic relations between two events derived from VerbOcean (Chklovski and Pantel, 2004) and WordNet (Fellbaum, 1998). VerbOcean contains five types of relations (*similarity*, *strength*, *antonymy*, *enablement*, and *happens-before*) that commonly occur between pairs of verbs. To overcome data sparseness, we expanded the original VerbOcean database by calculating symmetric and transitive closures of key relations. With WordNet, a semantic distance between the associated tokens of each target event pair was computed.

5 Modeling Temporal Relations with Markov Logic

ML is a statistical relational learning framework that provides a template language for defining *Markov Logic Networks* (MLNs). A MLN is a set of weighted first-order clauses constituting a Markov network in which each ground formula represents a feature (Richardson and Domingos, 2006).

Our MLN consists of a set of formulae combining two types of predicates: *hidden* and *observed*. Hidden predicates are those that are not directly observable during test time. A hidden predicate is defined for each task: *relEventTimex* (temporal relation between an event and a time expression), *relMainEvents* (temporal relation between two main events), and *relMainSub* (temporal relation between a main and a dominated event). Observed predicates are those that can be fully observed during test time and represent each of the features described in Section 4.

The following is an example formula used in our MLN:

$$eventTimex(d, e, t) \wedge eventWord(d, e, w) \rightarrow relEventTimex(d, e, t, r) \quad (1)$$

The predicate *eventTimex(d, e, t)* represents the existence of a candidate pair of event *e* and time expression *t* in a document *d*. Given this candidate pair, formula (1) assigns weights to a temporal relation *r* whenever it observes a word token *w* in the given event from the training data. This formula is local because it considers only one hidden predicate (*relEventTimex*).

In addition to local formulae, we also define a set of global formulae to ensure consistency between local decisions:

$$relEventTimex(d, e_1, t, r_1) \wedge relEventTimex(d, e_2, t, r_2) \rightarrow relMainSub(d, e_1, e_2, r_3) \quad (2)$$

Formula (2) is global because it jointly concerns more than one hidden predicate (*relEventTimex* and *relMainSub*) at the same time. This formula ensures consistency between the predicted temporal relations r_1, r_2, and r_3 given a main event e_1, a syntactically dominated event e_2, and a time expression *t* shared by both of these events. Two additional global formulae (3) and (4) are similarly defined to ensure consistency as below.

$$relMainSub(d, e_1, e_2, r_3) \wedge relEventTimex(d, e_2, t, r_2) \rightarrow relEventTimex(d, e_1, t, r_1) \quad (3)$$

$$relMainSub(d, e_1, e_2, r_3) \wedge relEventTimex(d, e_1, t, r_1) \rightarrow relEventTimex(d, e_2, t, r_2) \quad (4)$$

6 Evaluation

To evaluate the proposed approach, we built and compared two models: one model (*NoLex*) used all of the features described in Section 4 except for the lexical relation features, and the other model (*Full*) included the full set of features. The features were generated using the Porter Stemmer and WordNet Lemmatizer in NLTK (Loper and Bird, 2002) and the Charniak Parser (Charniak, 2000). The semantic distance between two word tokens was computed using the path-similarity metric provided by NLTK. All of the models were constructed using Markov TheBeast (Riedel, 2008)

The feature set was optimized for each task on a held-out development data set consisting of approximately 10% of the entire training set (Table 1). Our previous work (Ha et al., 2010) observed that a local optimization approach that selects for each individual task (i.e., each hidden predicate in the given MLN) in isolation from the other tasks could harm the overall accuracy of a joint model because of resulting inconsistencies among individual tasks. In the new experiment described in this section, features were selected for each task to improve overall accuracy of the joint model combining all three tasks, similar to Yoshikawa et al. (2009).

Feature		Task ET	MM	MS
Surface Features	*event-word*	√	√	√
	event-stem	√	√	√
	timex-word	√		
	timex-stem	√		
Semantic Attributes	*event-polarity*	√	√	√
	event-modal	√	√	√
	event-pos	√	√	√*
	event-tense	√	√	√
	event-aspect	√	√	√
	event-class	√	√	√
	timex-type	√		
	timex-value	√		
Syntactic Features	*pos*	√	√	√
	gov-prep	√	√	√
	gov-verb	√	√	√
	gov-verb-pos	√	√	√
Lexical Relations	*verb-rel*		√	√
	word-dist		√	

Table 1: Features used to model each task. *The feature is extracted only from the second event in the pair being compared.

Table 2 reports the resulting performance (*F1* scores) of the models. To isolate the potential effects of global constraints, we first compare the accuracies of the *Full* and the *NoLex* model, averaged from a ten-fold cross validation on the training data before global constraints are added. *Full* achieves relative 12% and 3% improvements over *NoLex* for temporal relation between events and time expressions (*ET*) and between two main events (*MM*), respectively. The improvement for *MM* was statistically significant ($p<0.05$) from a two-tailed paired *t*-test. Note that the *ET* task itself does not use lexical relation features but still achieves an improved result in *Full* over *NoLex*. This is an effect of joint modeling. There is a slight degradation (relative 2%) in the accuracy for temporal relations between main and syntactically dominated events (*MS*). Overall, *Full* achieves relative 5% improvement over *NoLex*. A similar trend of performance improvement in *Full* over *NoLex* was observed when the global formulae were added to each model. The second column (*Global Constraints*) of Table 2 compares the two models trained on the entire training set and tested on the test set after the global formulae were added. However, no statistical significance was found on these improvements. Compared to the state-

of-the-art results achieved by the TempEval-2 participants, *Full* achieves the same or better results on all three addressed tasks.

7 Conclusions

Temporal relations can be modeled with Markov Logic using a variety of features including lexical ontologies. Three tasks relating to the TempEval-2 data were addressed: predicting temporal relations between (1) events and time expressions in the same sentence, (2) two main events in consecutive sentences, and (3) two events in the same sentence when one syntactically dominates the other. An evaluation suggests that utilizing lexical relation features within a joint modeling framework using Markov Logic achieves state-of-the-art performance.

The results suggest a promising direction for future work. The proposed approach assumes events and time expressions are already marked in the data. To construct a fully automatic temporal relation identification system, the approach needs to be extended to include models that recognize events and time expressions in text as well as their semantic attributes. A data-driven approach similar to the one described in this paper may be feasible for this new modeling task. It will entail exploring a variety of features to further understand the complexity underlying the problem of temporal analysis of events.

Task	No Global Constraints		Global Constraints		State-of-the-art
	NoLex	Full	NoLex	Full	
Overall	0.60	0.63 (+5%)	0.59	0.61 (+3%)	NA
ET	0.52	0.58 (+12%)	0.62	0.65 (+5%)	0.63
MM	0.65	0.67 (+3%)*	0.52	0.56 (+8%)	0.55
MS	0.66	0.65 (- 2%)	0.66	0.66 (+0%)	0.66

Table 2. Performance comparison between models in *F1* score. *Statistical significance (p<0.05)

Acknowledgments

This research was supported by the National Science Foundation under Grant IIS-0757535.

References

S. Bethard and J. H. Martin. 2007. CU-TMP: Temporal relation classification using syntactic and semantic features. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 129-132, Prague, Czech Republic.

B. Boguraev and R. K. Ando. 2005. TimeML-compliant text analysis for temporal reasoning. In *Proceedings of the 19th International Joint Conference on Artificial intelligence*, pages 997-1003, Edinburgh, Scotland.

N. Chambers and D. Jurafsky. 2008. Jointly combining implicit constraints improves temporal ordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 698-706, Honolulu, HI.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, pages 132-139, Seattle, WA.

T. Chklovski and P. Pantel. 2004. VerbOcean: Mining the web for fine-grained semantic verb relations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 33-40, Barcelona, Spain.

E. Ha, A. Baikadi, C. Licata, and J. Lester. 2010. NCSU: Modeling temporal relations with Markov Logic and lexical ontology. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 341-344, Uppsala, Sweden.

E. Loper and S. Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 62-69, Philadelphia, PA.

J. Pustejovsky and M. Verhagen. 2009. SemEval-2010 task 13: Evaluating events, time expressions, and temporal relations. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 112-116, Boulder, CO.

S. Riedel. 2008. Improving the accuracy and efficiency of MAP inference for Markov Logic. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*, pages 468-475, Helsinki, Finland.

M. Richardson and P. Domingos. 2006. Markov Logic networks. *Machine Learning*, 62(1):107-136.

M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75-80, Prague, Czech Republic.

K. Yoshikawa, S. Riedel, M. Asahara, and Y. Matsumoto. 2009. Jointly identifying temporal relations with Markov Logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 405-413, Suntec, Singapore.

Reference reversibility with Reference Domain Theory

Alexandre Denis
TALARIS team / UMR 7503 LORIA/INRIA
Lorraine. Campus scientifique, BP 239
F-54506 Vandoeuvre-lès-Nancy cedex
alexandre.denis@loria.fr

Abstract

In this paper we present a reference model based on Reference Domain Theory that can work both in interpretation and generation. We introduce a formalization of key concepts of RDT, the interpretation and generation algorithms and show an example of behavior in the dynamic, asymmetric and multimodal GIVE environment.

1 Introduction

The reference task in a dialogue system is two-fold. On the one hand the system has to *interpret* the referring expressions (RE) produced by the user in his utterances. On the other hand the system has to *generate* the REs for the objects it aims to refer to. We present in this paper a framework that considers that reference interpretation and generation are two sides of the same coin, hence avoiding any potential misunderstanding arising from the two modules discrepancies. Reference Domain Theory (RDT) (Salmon-Alt and Romary, 2000; Salmon-Alt and Romary, 2001) proposes to represent the diversity of referring acts by the diversity of constraints they impose on their context of use. The reversibility then lies in the possibility to express these constraints *independently of the considered task*.

In (Denis, 2010) we described the generation side of RDT in the context of the GIVE-2 challenge (Koller et al., 2010) which is an evaluation of instruction generation systems in a 3D maze. In this paper we propose the interpretation counterpart and show the required modeling to consider the dynamic, asymmetric and multimodal context of GIVE. We first present the reference model in section 2 and 3, discuss the interpretation problems in GIVE in section 4, detail an example in section 5 and present evaluation results in section 6.

2 Reference Domains

A rich contextual structure is required to give an account for the different kinds of discrimination we observe in REs such as semantic discrimination (e.g. "the blue button"), focus discrimination (e.g. "this button") and salience discrimination (e.g. "this one"). We introduce here the structure of *reference domain* which is a local context supporting these different discriminations.

We assume that *Props* is the set of unary predicate names e.g. $\{blue, left, ...\}$, *Types* is the set of types of predicates e.g. {*color, position, ...*}, and *val* is the function $val : Types \rightarrow 2^{Props}$ which maps a type on the predicates names. Finally, E is the set of all objects and V the set of ground predicates e.g. {*blue(b1), ...*}.

A *reference domain* D is then a tuple

$$\langle G_D, S_D, \sigma_D, (c, P, F)\rangle$$

where $G_D \subseteq E$ is the set of objects of the domain, called the *ground of the domain*; $S_D \subseteq$ Props is the *semantic description* of the domain, satisfied by all elements of the ground; $\sigma_D \in \mathbb{N}$ is the *salience* of the domain. And (c, P, F) is a *partition structure* where $c \in$ Types is a *differentiation criterion*; P is the *partition* generated by c; and $F \subseteq P$ is the *focus* of P.

For instance, a domain composed of a blue button b_1 and a red button b_2, with a salience equal to 3, where b_1 and b_2 are differentiated using the color, and where b_1 is in focus, would be noted as:

$$\begin{aligned} D = &\langle \{b_1, b_2\}, \{button\}, 3, \\ &(color, \{\{b_1\}, \{b_2\}\}, \{\{b_1\}\})\rangle \end{aligned}$$

Finally we define a *referential space* (RS) as a set of reference domains (RD) ordered by salience.

3 Referring

A RE impose some constraints on the context in which it can be uttered, that is in which RD the interpretation has to be made. The constraints are represented as *underspecified domains* (UD), specifying the structure of the suitable RD in terms of ground, salience or partition. The explicit definitions of the UDs makes possible to share these definitions between the interpretation and the generation modules, hence allowing the implementation of a *type B reversible reference module* (Klarner, 2005), that is a module in which both directions share the same resources.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 79–82,
The University of Tokyo, September 24-25, 2010.

Expression	$U(N,t)$ **matches** D iff $\exists(c,P,F) \in D$;
this one	$F = \{\{t\}\} \wedge \text{msd}(D)$
this N	$F = \{\{t\}\} \wedge t \in N^{\mathcal{I}}$
the N	$t \in N^{\mathcal{I}} \wedge \{t\} \in P \wedge \forall X \in P, X \neq \{t\} \Rightarrow X \cap N^{\mathcal{I}} = \emptyset$
the other one	$F \neq \emptyset \wedge P \setminus F = \{\{t\}\} \wedge \text{msd}(D)$
the other N	$F \neq \emptyset \wedge P \setminus F = \{\{t\}\} \wedge G_D \subseteq N^{\mathcal{I}}$
another one	$F \neq \emptyset \wedge \{t\} \in P \setminus F \wedge \text{msd}(D)$
another N	$F \neq \emptyset \wedge \{t\} \in P \setminus F \wedge G_D \subseteq N^{\mathcal{I}}$
a N	$t \in N^{\mathcal{I}} \wedge t \in G_D$

Table 1: Underspecified domains for each type of referring expression

3.1 Underspecified domains

The different types of UDs are presented in table 1. Each UD is a parametric conjunction of constraints on a RD, noted $U(N,t)$, where t is the intended referent and $N \subseteq \mathit{Props}$ is a semantic description. $N^{\mathcal{I}}$ stands for the *extension* of N, and $msd(D)$ stands for *most salient description*, that is, there is no more or equally salient domain than D in the current RS with a different description. Each UD is associated to a *wording* combining a determiner and a wording of the semantic description, for instance "the N" is a shortcut for a definite expression whose head noun and modifiers are provided by the wording of N. Finally we say that an UD *matches* a RD if all the constraints of the UD are satisfied by the RD.

3.2 Referring processes

Interpretation and generation can now be defined in terms of UD. The two processes are illustrated in figure 1 and the algorithms are presented in figure 2.

The *interpretation* algorithm consists in finding or creating a RD from the input UD, $U(N,.)$ created from the input RE type and description N. The algorithm then iterates through the RS in salience order, and through all the individuals t of the tested domain to retrieve the first one matching $U(N,t)$. If a matching domain D is found, a restructuring operation is applied and the referent t is focused in the partition of D. On the other hand, if no domain is found, the UD is *accommodated*, that is a new domain and a new referent satisfying the constraints of $U(N,t)$ are created. According to the task, this accommodation may not be possible for all REs, but for sake of simplicity we assume here this operation is always possible.

The *generation* side is the opposite, that is it finds an UD from an input RD. It first selects a RD containing the target referent to generate t, assuming here that the most salient domain has to be preferred. The description N used to instantiate the UDs is composed of the description of the domain and the description of the referent in the partition (line 2). It then iterates through the different UDs by Givenness order (Gundel et al., 1993) and selects the first one that matches. A restructuring operation is applied and the found UD is returned, eventually providing the RE.

The restructuring operation, detailed in (Denis, 2010), aims to restrict the current context by creating a new domain around the referent in the referential space or by increasing the salience of the domain containg the referent. This operation helps to perform focalization in restricted domains.

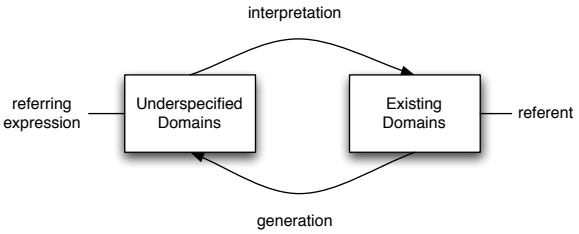


Figure 1: Reference processes

4 The complex context of GIVE

The dynamic, asymmetric and multimodal context of GIVE requires additional mechanisms for interpretation. Asymmetry causes the *late visual context integration*, when the direction giver produces a RE to objects not yet known by the direction follower, that are only visually discovered later on. Space prevents us to describe in details the late integration algorithm, but the idea is, given a new physical object t, to scan existing domains of the actual RS to check if t can be merged semantically with any previous object t'. If this could be the case, the integration leads to create two parallel RS, one in which $t = t'$ (the *fusion* hypothesis) and one in which $t \neq t'$ (the *separation* hypothesis). If this cannot be the case, t is added as a new object. Following (DeVault and Stone, 2007), these alternative contexts can persist across time and further referring expressions may reject one or the other hypothesis as illustrated in section 5.

The second required mechanism is the proper handling of the *multimodal dynamic focus*, that is the combination of the linguistic focus resulting from RE, and the visual focus. It is possible to have two referential spaces for the linguistic or visual context as in (Kelleher et al., 2005; Byron et al., 2005), or to have two foci in a partition. We can also model *interleaved focus*, that is, only one focus per domain but that dynamically corresponds to the linguistic focus or the visual focus. The idea is that after each RE, the referent receives the focus as described in algorithm 1, but whenever the visual context changes, the focus is updated to the visible objects. Although interleaved focus prevents anaphora while the visual context changes, its complexity is enough for our setup.

Algorithm 1 interpret($U(N,.)$, RS)

1: **for all** domain D in RS by salience order **do**
2: **for all** $t \in G_D$ **do**
3: **if** $U(N,t)$ matches D **then**
4: restructure(D, N, RS)
5: focus t in D
6: **return** t
7: **end if**
8: **end for**
9: **end for**
10: **return** accommodate($U(N,.)$, RS)

Algorithm 2 generate(t, RS)

1: $D \leftarrow$ most salient domain containing t
2: $N \leftarrow S_D \cup \{p | p \in val(c), p(t) \in V\}$
3: **for all** $U(N,t)$ sorted by Givenness **do**
4: **if** $U(N,t)$ matches D **then**
5: restructure(D, N, RS)
6: **return** $U(N,t)$
7: **end if**
8: **end for**
9: **return** failure

Figure 2: Reference algorithms, relying on the same underspecified domains

5 Example

In this section we present the interpretation side of some expressions we generated in the GIVE setting (table 2). The detailed generation side of this example can be found in (Denis, 2010). S is the system that interprets the RE of U the user. The situation is: S enters a room with two blue buttons b_1 and b_2, none of them being visible when he enters and U wants to refer to b_1.

state of S	utterance of U
	Push a blue button (b_1)
see(b_2)	Not this one! Look for the other one!
see(b_1)	Yeah! This one!

Table 2: Utterances produced by U

When S enters the room, U generates an indefinite RE "Push a blue button". S first constructs an indefinite UD "a N" with $N = \{blue, button\}$. However, because there exists no RD at first, he has to accommodate the UD, hence creating a new domain D_1 containing a new linguistically focused individual t:

$$D_1 = \langle \{t\}, \{button, blue\}, 1, (\text{id}, \{\{t\}\}, \{\{t\}\}) \rangle$$

We assume that S moves and now sees the blue button b_2 without knowing yet if this is the intended one. The integration of this new physical object then leads to two hypothesis. In the *fusion* hypothesis, $b_2 = t$, and in the *separation* hypothesis, $b_2 \neq t$. In both cases, the visible button is focused in the two versions of D_1, D_{1FUS} and D_{1SEP}:

$$D_{1FUS} = \langle \{t\}, \{button, blue\}, 2, (\text{id}, \{\{t\}\}, \{\{t\}\}) \rangle$$
$$D_{1SEP} = \langle \{t, b_2\}, \{button, blue\}, 2, (\text{id}, \{\{t\}, \{b_2\}\}, \{\{b_2\}\}) \rangle$$

However, U utters "Not this one!" rejecting then the fusion hypothesis. To be able to consider the effects of this utterance, we have to take into account the ellipsis. This can be done by assuming that U is asserting properties of the target of his first RE, that is, he is actually stating that "[t is] not this one!". The RE "this one" leads to the construction of a demonstrative one-anaphora UD that matches t in D_{1FUS} but b_2 in D_{1SEP}. The following schema shows the contradiction in the fusion hypothesis:

	t	is not	this one
fusion	t	$\neq$	t
separation	t	$\neq$	b_2

Being contradictory, the fusion hypothesis is rejected and only D_{1SEP} is maintained. For the readability of the presentation, D_{1SEP} is rewritten as D_1.

The interpretation of "Look for the other one!" is straightforward. A definite alternative one-anaphora UD is built, and both t and b_2 are tested in D_1 but only t is matched because it is unfocused (see the definition of the alternative one-anaphora in table 1).

Now S moves again and sees b_1. As for b_2, the integration of b_1 in the referential space leads to two alternative RS. The buttons b_2 and b_1 cannot be merged (we assume here that S can clearly see they are two different buttons), thus the two alternative RS are whether $b_1 = t$ or $b_1 \neq t$:

$$D_{1FUS} = \langle \{t, b_2\}, \{button, blue\}, 3, (\text{id}, \{\{t\}, \{b_2\}\}, \{\{t\}\}) \rangle$$
$$D_{1SEP} = \langle \{t, b_1, b_2\}, \{button, blue\}, 3, (\text{id}, \{\{t\}, \{b_1\}, \{b_2\}\}, \{\{b_1\}\}) \rangle$$

Eventually S has to interpret "this one". Like previously, in order to take into account the effects of this utterance, S has to resolve the ellipsis and must consider "[t is] this one". The RE "this one" is resolved on t in D_{1FUS} but on b_1 in D_{1SEP}.

	t	is	this one
fusion	t	=	t
separation	t	=	b_1

This is now the separation hypothesis which is inconsistent because we assumed that $b_1 \neq t$. This RS is then ruled out, and only the fusion RS remains.

6 Evaluation

Only the generation direction has been evaluated in the GIVE challenge. The results (Koller et al., 2010) show that the system embedding Reference Domain Theory proves to rely on less instructions than other systems (224) and proves to be the most successful (47% of task success) while being the fastest (344 seconds). We conjecture that the good results of RDT can be explained by the low cognitive load resulting from the use of demonstrative NPs and one-anaphoras, but the role of the overall generation strategy has also to be taken into account in these good results (Denis et al., 2010).

Although it would be very interesting, the interpretation side has not yet been evaluated in the GIVE setting, but only in the MEDIA campaign (Bonneau Maynard et al., 2009) which is an unimodal setting. The results show that the interpretation side of RDT achieves a fair precision in identification (75.2%) but a low recall (44.7%). We assume that the low recall of the module is caused by the cascade of errors, one error at the start of a reference chain leading to several other errors. Nonetheless, we estimate that error cascading would be less problematic in the GIVE setting because of its dynamicity.

7 Conclusions

We presented a reference framework extending (Salmon-Alt and Romary, 2001) in which interpretation and generation can be defined in terms of the constraints imposed by the referring expressions on their context of use. The two modules sharing the same library of constraints, the model is then said *reversible*. However, because of the asymmetry and dynamicity of our setup, the GIVE challenge, additional mechanisms such as uncertainty have to be modeled. In particular, we have to maintain different interpretation contexts like (DeVault and Stone, 2007) to take into account the ambiguity arising from the late integration of the visual context. It would be interesting now to explore deeper our reversibility claim by evaluating the interaction between the two reference algorithms in the GIVE setting.

References

Hélène Bonneau Maynard, Matthieu Quignard, and Alexandre Denis. 2009. MEDIA: a semantically annotated corpus of task oriented dialogs in French. *Language Resources and Evaluation*, 43(4):329–354.

Donna K. Byron, Thomas Mampilly, Vinay Sharma, and Tianfang Xu. 2005. Utilizing visual attention for cross-modal coreference interpretation. In *Proceedings of Context-05*, pages 83–96.

Alexandre Denis, Marilisa Amoia, Luciana Benotti, Laura Perez-Beltrachini, Claire Gardent, and Tarik Osswald. 2010. The GIVE-2 Nancy Generation Systems NA and NM. Technical report.

Alexandre Denis. 2010. Generating referring expressions with Reference Domain Theory. In *Proceedings of the 6th International Natural Language Generation Conference - INLG 2010*, Dublin, Ireland.

David DeVault and Matthew Stone. 2007. Managing ambiguities across utterances in dialogue. In *Proceedings of the 2007 Workshop on the Semantics and Pragmatics of Dialogue (DECALOG 2007)*, Trento, Italy.

Jeanette K. Gundel, Nancy Hedberg, and Ron Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language*, 69(2):274–307.

John Kelleher, Fintan Costello, and Josef van Genabith. 2005. Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context. *Artificial Intelligence*, 167(1-2):62–102.

Martin Klarner. 2005. Reversibility and reusability of resources in NLG and natural language dialog systems. In *Proceedings of the 10th European Workshop on Natural Language Generation (ENLG-05)*, Aberdeen, Scotland.

Alexander Koller, Kristina Striegnitz, Andrew Gargett, Donna Byron, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2010. Report on the second NLG challenge on generating instructions in virtual environments (GIVE-2). In *Proceedings of the 6th International Natural Language Generation Conference - INLG 2010*, Dublin, Ireland.

Susanne Salmon-Alt and Laurent Romary. 2000. Generating referring expressions in multimodal contexts. In *Workshop on Coherence in Generated Multimedia - INLG 2000*, Israel.

Susanne Salmon-Alt and Laurent Romary. 2001. Reference resolution within the framework of cognitive grammar. In *Proceeding of the International Colloquium on Cognitive Science*, San Sebastian, Spain.

Utilizing Review Summarization in a Spoken Recommendation System

Jingjing Liu, Stephanie Seneff and Victor Zue
MIT Computer Science & Artificial Intelligence Laboratory
Cambridge, U.S.A.
{jingl,seneff,zue}@csail.mit.edu

Abstract

In this paper we present a framework for spoken recommendation systems. To provide reliable recommendations to users, we incorporate a review summarization technique which extracts informative opinion summaries from grass-roots users' reviews. The dialogue system then utilizes these review summaries to support both *quality-based* opinion inquiry and *feature-specific* entity search. We propose a probabilistic language generation approach to automatically creating recommendations in spoken natural language from the text-based opinion summaries. A user study in the restaurant domain shows that the proposed approaches can effectively generate reliable and helpful recommendations in human-computer conversations.

1 Introduction

Spoken dialogue systems are presently available for many purposes, such as flight reservations (Seneff and Polifroni, 2000), telephone calls routing (Gorin et al., 1997), and subway lookup (Johnston et al., 2002). Recently, we have been exploring a next generation of intelligent dialogue systems, which can behave like a human agent and provide proactive assistance and selective recommendations (e.g., highly-rated restaurants or hotels) to users.

To enhance dialogue systems with intelligent services, we have to let the system "grow" reliable knowledge and intelligence. Luckily, there has recently been an explosive growth in the availability of public review sites (e.g., yelp.com, tripadvisor.com, etc.) which make a perfect resource for gathering collective opinions. In this paper, we will explore how to utilize summaries of public reviews to automatically generate recommendation-type conversations in spoken dialogue systems. An example of a conversation with our recommendation system is shown in Figure 1.

U: Are there any American restaurants in Brighton?
S: There are 61 American restaurants in Brighton. All of the rated restaurants have *good or excellent ambiance*. Many are on Washington Street, Commonwealth Avenue and Market Street.
U: Show me the ones that have *nice ambiance*.
S: I found 4 entries. Devlin's Restaurant and Bar *is perfect for romantic date*, Stockyard Restaurant has *friendly atmosphere*, Tasca Tapas Restaurant *has excellent ambiance*, and Wings Express *has friendly vibes*.
U: Which one has *good martinis*?
S: Devlin's Restaurant and Bar serves American food. It has *wonderful martinis*, good wine, great fish. It is *perfect for romantic date*. *Great place*.

Figure 1. A real conversation with our recommendation dialogue system in the restaurant domain ('U' is the user and 'S' is the system).

2 Dialogue Management

In our previous work (Liu and Seneff, 2009; Liu et al., 2010) we proposed an approach to extracting representative phrases and creating aspect ratings from public reviews. An example of an enhanced database entry in the restaurant domain is shown in Figure 2. Here, we use these "summary lists" (e.g., ":food", ":atmosphere") as well as aspect ratings (e.g., ":food_rating") to address two types of recommendation inquires: "feature-specific" (e.g., asking for a restaurant that serves good martinis or authentic seafood spaghetti), and "quality-based" (e.g., looking for restaurants with good food quality or nice ambiance).

```
{q restaurant
   :name "devlin's restaurant and bar"
   :atmosphere ("romantic date" "elegant decor")
   :place ("great place")
   :food ("wonderful martinis" "good wine" "great fish")
   :atmosphere_rating "4.2"
   :place_rating "4.2"
   :food_rating "4.3"
   :specialty ("martinis" "wine" "fish")    }
```

Figure 2. A database entry in our system.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 83–86,
The University of Tokyo, September 24-25, 2010.

2.1 Feature-specific Entity Search

To allow the system to identify feature-related topics in users' queries, we modify the context-free grammar in our linguistic parser by including feature-specific topics (e.g., nouns in the summary lists) as a word class. When a feature-specific query utterance is submitted by a user (as exemplified in Figure 3), our linguistic parser will generate a hierarchical structure for the utterance, which encodes the syntactic and semantic structure of the utterance and, especially, identifies the feature-related topics. A feature-specific key-value pair (e.g., "specialty: martinis") is then created from the hierarchical parsing structure, with which the system can filter the database and retrieve the entities that satisfy the constraints.

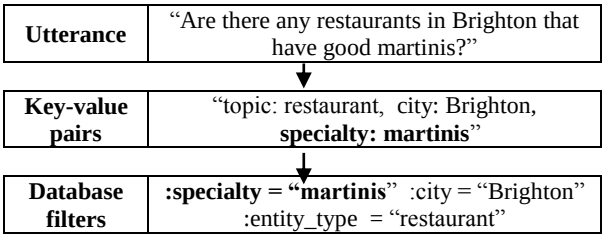


Figure 3. Procedure of feature-specific search.

2.2 Quality-based Entity Search

For quality-based questions, however, similar keyword search is problematic, as the quality of entities has variants of expressions. The assessment of different degrees of sentiment in various expressional words is very subjective, which makes the quality-based search a hard problem.

To identify the strength of sentiment in quality-based queries, a promising solution is to map textual expressions to scalable numerical scores. In previous work (Liu and Seneff, 2009), we proposed a method for calculating a sentiment score for each opinion-expressing adjective or adverb (e.g., 'bad': 1.5, 'good': 3.5, 'great': 4.0, on a scale of 1 to 5). Here, we make use of these sentiment scores and convert the original key-value pair to numerical values (e.g., "great food" → "food_rating: 4.0" as exemplified in Figure 4). In this way, the sentiment expressions can be easily converted to scalable numerical key-value pairs, which will be used for filtering the database by "aspect ratings" of entities. As exemplified in Figure 4, all the entities in the required range of aspect rating (i.e., ":food_rating ≥ 4.0") can be retrieved (e.g., the entity in Figure 2 with "food_rating = 4.3").

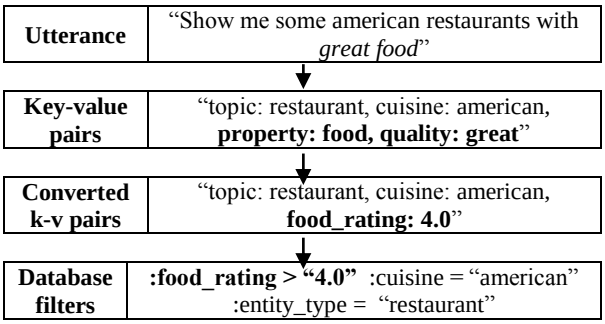


Figure 4. Procedure of qualitative entity search.

3 Probabilistic Language Generation

After corresponding entities are retrieved from the database based on the user's query, the language generation component will create recommendations by expanding the summary lists of the retrieved database entries into natural language utterances.

Most spoken dialogue systems use predefined templates to generate responses. However, manually defining templates for each specific linguistic pattern is tedious and non-scalable. For example, given a restaurant with "nice jazz music, best breakfast spot, great vibes", three templates have to be edited for three different topics (e.g., "<restaurant> *plays* <adjective> music"; "<restaurant> *is* <adjective> breakfast spot"; "<restaurant> *has* <adjective> vibes"). To avoid the human effort involved in the task, corpus-based approaches (Oh and Rudnicky, 2000; Rambow et al., 2001) have been developed for more efficient language generation. In this paper, we propose a corpus-based probabilistic approach which can automatically learn the linguistic patterns (e.g., predicate-topic relationships) from a corpus and generate natural sentences by probabilistically selecting the best-matching pattern for each topic.

The proposed approach consists of three stages: 1) plant seed topics in the context-free grammar; 2) identify semantic structures associated with the seeds; 3) extract association pairs of linguistic patterns and the seeds, and calculate the probability of each association pair.

First, we extract all the nouns and noun phrases that occur in the review summaries as the seeds. As aforementioned, our context-free grammar can parse each sentence into a hierarchical structure. We modify the grammar such that, when parsing a sentence which contains one of these seed topics, the parser can identify the seed as an "active" topic (e.g., "vibes", "jazz music", and "breakfast spot").

The second stage is to automatically identify all the linguistic patterns associated with each seed. To do so, we use a large corpus as the resource pool and parse each sentence in the corpus for linguistic analysis. We modify our parser such that, in a preprocessing step, the predicate and clause structures that are semantically related to the seeds will be assigned with identifiable tags. For example, if the subject or the complement of the clause (or the object of the predicate) is an "active" topic (i.e., a seed), an "active" tag will be automatically assigned to the clause (or the predicate). In this way, when examining syntactic hierarchy of each sentence in the corpus, the system can encode all the linguistic patterns of clauses or predicate-topic relationships associated with the seeds with "active" tags.

Based on these tags, association pairs of "active" linguistic patterns and "active" topics can be extracted automatically. For each seed topic, we calculate the probability of its co-occurrence with each of its associated patterns by:

$$prob(pattern_j | seed_k) = \frac{count(pattern_j,\ seed_k)}{\sum_i count(pattern_i,\ seed_k)} \quad (1)$$

where $seed_k$ is a seed topic, and $pattern_i$ is every linguistic pattern associated with $seed_k$. The probability of $pattern_j$ for $seed_k$ is the percentage of the co-occurrences of $pattern_j$ and $seed_k$ among all the occurrences of $seed_k$ in the corpus. This is similar to a bigram language model. A major difference is that the linguistic pattern is not necessarily the word adjacent to the seed. It can be a long distance from the seed with strong semantic dependencies, and it can be a semantic chunk of multiple words. The long distance semantic relationships are captured by our linguistic parser and its hierarchical encoding structure; thus, it is more reliable than pure co-occurrence statistics or bigrams. Figure 5 shows some probabilities learned from a review corpus. For example, "is" has the highest probability (0.57) among all the predicates that co-occur with "breakfast spot"; while "have" is the best-match for "jazz music".

Association pair	Constituent	Prob.
"at" : "breakfast spot"	PP	0.07
"is" : "breakfast spot"	Clause	**0.57**
"for" : "breakfast spot"	PP	0.14
"love" : "jazz music"	VP	0.08
"have" : "jazz music"	VP	**0.23**
"enjoy": "jazz music"	VP	0.08

Figure 5. Partial table of probabilities of association pairs (VP: verb phrase; PP: preposition phrase).

Given these probabilities, we can define pattern selection algorithms (e.g., always select the pattern with the highest probability for each topic; or rotates among different patterns from high to low probabilities), and generate response utterances based on the selected patterns. The only domain-dependent part of this approach is the selection of the seeds. The other steps all depend on generic linguistic structures and are domain-independent. Thus, this probabilistic method can be easily applied to generic domains for customizing language generation.

4 Experiments

A web-based multimodal spoken dialogue system, CityBrowser (Gruenstein and Seneff, 2007), developed in our group, can provide users with information about various landmarks such as the address of a museum, or the opening hours of a restaurant. To evaluate our proposed approaches, we enhanced the system with a review-summary database generated from a review corpus that we harvested from a review publishing web site (www.citysearch.com), which contains 137,569 reviews on 24,043 restaurants.

We utilize the platform of Amazon Mechanical Turk (AMT) to conduct a series of user studies. To understand what types of queries the system might potentially be handling, we first conducted an AMT task by collecting restaurant inquiries from general users. Through this AMT task, 250 sentences were collected and a set of generic templates encoding the language patterns of these sentences was carefully extracted. Then 10,000 sentences were automatically created from these templates for language model training for the speech recognizer.

To evaluate the quality of recommendations, we presented the system to real users via customized AMT API (McGraw et al., 2010) and gave each subject a set of assignments to fulfill. Each assignment is a scenario of finding a particular restaurant, as shown in Figure 6. The user can talk to the system via a microphone and ask for restaurant recommendations.

We also gave each user a questionnaire for a subjective evaluation and asked them to rate the system on different aspects. Through this AMT task we collected 58 sessions containing 270 utterances (4.6 utterances per session on average) and 34 surveys. The length of the utterances varies significantly, from "Thank you" to "Restaurants along Brattle Street in Cambridge with nice

cocktails." The average number of words per utterance is 5.3.

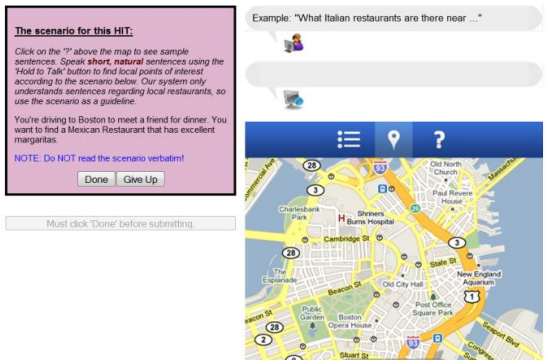


Figure 6. Interface of our system in an AMT assignment.

Among all the 58 sessions, 51 were successfully fulfilled, i.e., in 87.9% of the cases the system provided helpful recommendations upon the user's request and the user was satisfied with the recommendations. Among those seven failed cases, one was due to loud background noise, two were due to users' operation errors (e.g., clicking "DONE" before finishing the scenario), and four were due to recognition performance.

The user ratings in the 34 questionnaires are shown in Figure 7. On a scale of 0 (the center) to 5 (the edge), the average rating is 3.6 on the easiness of the system, 4.4 on the helpfulness of the recommendations, and 4.1 on the naturalness of the system response. These numbers indicate that the system is very helpful at providing recommendation upon users' inquiries, and the response from the system is present in a natural way that people could easily understand.

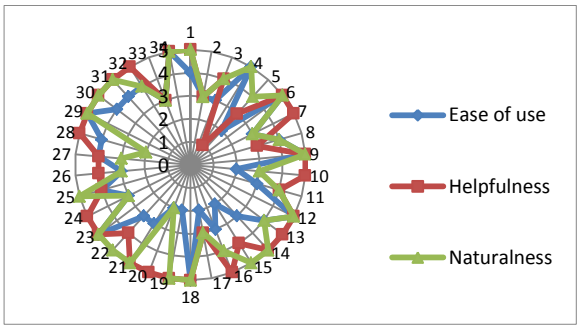


Figure 7. Users' ratings from the questionnaires.

The lower rating of ease of use is partially due to recognition errors. For example, a user asked for "pancakes", and the system recommended "pizza places" to him. In some audio clips recorded, the background noise is relatively high. This may be due to the fact that some AMT workers work from home, where it can be noisy.

5 Conclusions

In this paper we present a framework for incorporating review summarization into spoken recommendation systems. We proposed a set of entity search methods as well as a probabilistic language generation approach to automatically create natural recommendations in human-computer conversations from review summaries. A user study in the restaurant domain shows that the proposed approaches can make the dialogue system provide reliable recommendations and can help general users effectively.

Future work will focus on: 1) improving the system based on users' feedback; and 2) applying the review-based approaches to dialogue systems in other domains.

Acknowledgments

This research is supported by Quanta Computers, Inc. through the T-Party project.

References

Gorin, A., Riccardi, G., and Wright, J. H. 1997. How May I Help You? *Speech Communications*. Vol. 23, pp. 113 – 127.

Gruenstein, A. and Seneff, S. 2007. Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms. *In Proc. the 8th SIGdial Workshop on Discourse and Dialogue*, pp. 111—119.

Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., Maloor, P. 2002. MATCH: An Architecture for Multimodal Dialogue Systems. *In Proc. ACL*, pp. 376 – 383.

Liu, J. and Seneff, S. 2009. Review sentiment scoring via a parse-and-paraphrase paradigm, *In Proc. EMNLP*, Vol. 1.

Liu, J., Seneff, S. and Zue, V. 2010. Dialogue-Oriented Review Summary Generation for Spoken Dialogue Recommendation Systems. *In Proc. NAACL-HLT*.

McGraw, I., Lee, C., Hetherington, L., Seneff, S., Glass, J. 2010. Collecting Voices from the Cloud. *In Proc. LREC*.

Oh, A.H. and Rudnicky, A.I. 2000. Stochastic Language Generation for Spoken Dialogue Systems. *In Proc. of ANLP-NAACL*, pp. 27-32.

Rambow, O., Bangalore, S., Walker, M. 2001. Natural Language Generation in Dialog Systems. *In Proc. Human language technology research*.

Seneff, S. and Polifroni, J. 2000. Dialogue Management in the Mercury Flight Reservation System. *In Proc. Dialogue Workshop, ANLP-NAACL*.

Dialogue Management Based on Entities and Constraints

Yushi Xu **Stephanie Seneff**
Spoken Language Systems Group
MIT Computer Science and Artificial Intelligence Laboratory
United States
{yushixu, seneff}@csail.mit.edu

Abstract

This paper introduces a new dialogue management framework for goal-directed conversations. A declarative specification defines the domain-specific elements and guides the dialogue manager, which communicates with the knowledge sources to complete the specified goal. The user is viewed as another knowledge source. The dialogue manager finds the next action by a mixture of rule-based reasoning and a simple statistical model. Implementation in the flight-reservation domain demonstrates that the framework enables the developer to easily build a conversational dialogue system.

1 Introduction

Conversational systems can be classified into two distinct classes: goal-directed and casual chatting. For goal-directed systems, the system is usually more "knowledgeable" than the user, and it attempts to satisfy user-specified goals. The system's conversational strategies seek the most efficient path to reach closure and end the conversation (Smith, Hipp, & Biermann, 1995).

An essential commonality among different goal-directed applications is that, at the end of a successful conversation, the system presents the user with a "goal" entity, be it a flight itinerary, a route path, or a shopping order. Different conversations result from different properties of the goal entities and different constraints set by the knowledge sources. The properties define the necessary and/or relevant information, such as flight numbers in the flight itinerary. Constraints specify the means to obtain such information. For examples fields "source", "destination" and "date" are required to search for a flight. Once the properties and constraints are known, dialogue rules can easily map to dialogue actions.

This paper introduces a dialogue management framework for goal-directed conversation based on entity and knowledge source specification. The user is viewed as a collaborator with the dialogue manager, instead of a problem-raiser. The dialogue manager follows a set of definitions and constraints, and eventually realizes the goal entity. It also incorporates a simple statistical engine to handle certain decisions.

2 Related Work

In recent years, statistical methods have gained popularity in dialogue system research. Partially Observable Markov decision processes have been the focus of a number of papers (Levin, Pieraccini, & Eckert, 1997; Scheffler & Young, 2001; Frampton & Lemon, 2006; Williams & Young, 2007). These approaches turn the dialogue interaction strategy into an optimization problem. The dialogue manager selects actions prescribed by the policy that maximizes the reward function (Lemon & Pietquin, 2007). This machine learning formulation of the problem automates system development, thus freeing the developers from hand-coded rules.

Other researchers have continued research on rule-based frameworks, in part because they are easier to control and maintain. One common approach is to allow developers to specify the tasks, either using a conditioned sequential script (Zue, et al., 2000; Seneff, 2002), or using a task hierarchy (Hochberg, Kambhatla, & Roukos, 2002). In (Bohus & Rudnicky, 2003)'s work, a tree of dialogue agents, each of which handles different dialogue actions, is specified to control the dialogue progress. The knowledge has also been specified either by first order logic (Bühler & Minker, 2005) or ontology information (Milward & Beveridge, 2004).

3 Dialogue Manager

Figure 1 illustrates the architecture of the proposed dialogue management framework. Com-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 87–90,
The University of Tokyo, September 24-25, 2010.

munication with the dialogue manager (DM) is via "E-forms" (Electronic forms), which consist of language-independent key-value pairs. The language understanding and language generation components mediate between the DM and various knowledge sources (KS), including the user, to interpret the output from the KS and generate input that the KS can understand. Each KS handles one or more sub-domains. For example, a date/time KS can resolve a date expression such as "next Tuesday" to a unique date; a flight database can provide flight information. The KSes are provided by the developer. They can be local (a library) or external (a separate executable).

Within this architecture, the user is viewed as a special KS, who understands and speaks a natural language, so that the whole architecture is completely DM-centered, as shown in Figure 1. An external language understanding system parses the original input into an E-form, and an external language generation component converts the output E-form into the desired natural language. Each particular communication with the user is analogous to other communications with the various KSes. The user is always ranked the lowest in the priority list of the KSes, *i.e.*, only when other knowledge sources cannot provide the desired information does the DM try to ask the user.

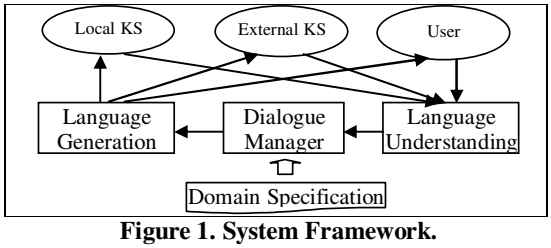


Figure 1. System Framework.

For example, in the flight reservation system, suppose the DM first tries to determine the source airport. If there exists a KS that contains this user's home airport information, the DM will adopt it. If no other KS can provide the information, the DM asks the user for the departure city.

3.1 Entity-Based Specification

Our framework uses an entity-based declarative domain specification. Instead of providing the action sequence in the domain, the developer provides the desired form of the goal entity, and the relationships among all relevant entities.

The specification is decomposed into two parts. The first part is the declaration of the knowledge sources. Each KS may contain one or more sub-domains, and an associated "nation" defines the language processing parameters.

The second part is the entity type definition. For a particular domain, there is one goal entity type, and an arbitrary number of other entity types, *e.g.*, two entity types are defined in the flight reservation system: "itinerary" and "flight." The definition of an entity type consists of a set of members, including their names, types and knowledge domain. A logical expression states the conditions under which the entity can be regarded as completed; *e.g.*, a completed itinerary must contain one or more flights. The entity definition can also include optional elements such as comparative/superlative modifiers or customized command-action and task-action mappings, described in more detail later.

The entity-based specification has an advantage over an action-based specification in two aspects. First, it is easier to define all the entities in a dialogue domain than to list all the possible actions, so the specification is more compact and readable. Secondly, the completion condition and the KS's constraints capture the underlying motivation of the dialogue actions.

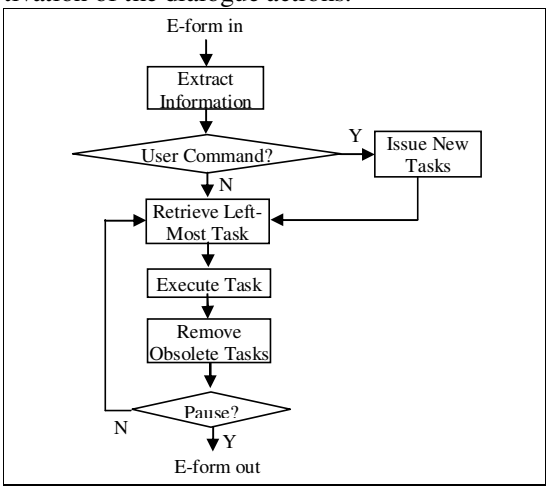


Figure 2. The Main Loop of the DM.

3.2 Dialogue Execution

Similar to the Information-State-Update (Larsson & Traum, 2000) idea, the DM maintains an internal state space with all up-to-date information about the entities. It also keeps a task list tree with a root task "complete goal." In task execution, subtasks (child node) and/or subsequent (right sibling node) tasks are issued. Each time the left-most leaf task is executed, and when a task is completed, the DM checks all tasks and removes those that have been rendered obsolete.

Ten basic tasks are pre-defined in the DM, including *complete_entity*, *inquire_ks*, and some other tasks related to entity manipulation. A *complete_entity* task evaluates the completion

conditions and issues appropriate tasks if they are unmet. An *inquire_ks* task handles communication with the KSes, and issues subtasks if the query does not satisfy the constraints. A default action associated with each task can be replaced by customized task-action mappings if needed.

Figure 2 shows the main loop of the DM. The process loops until a "pause" is signaled, which indicates to await the user's spoken response. An example will be given in Section 4.

3.3 Statistical Inference

To cope with situations that rules cannot handle easily, the framework incorporates a simple statistical engine using a Space Vector Model. It is designed only to support inference on specific small problems, for example, to decide when to ask the user for confirmation of a task. Models are built for each of the inference problems. The output label of a new data point is computed by weighting the labels of all existing data by their inverse distances to the new data point.

Equations (1) to (3) show the detailed math of the computation, where x is the new data point and d^j is the j-th existing data point. α is a fading coefficient which ranges from 0 and 1. β, a correction weight, has a higher value for data points resulting from manual correction. $\delta(\cdot)$ is 1 when the two inputs are equal and 0 otherwise. *sim(x, d)* defines the similarity between the new data point and the existing data point. Function $dis(\cdot)$ indicates the distance for a particular dimension, which is specified by the developer. The weight for each dimension w_i is proportional to the count of distinct values of the particular dimension $c(\boldsymbol{D}_i)$ and the mutual information between the dimension and the output label.

$$f(x) = \underset{y_i}{\operatorname{argmax}} \sum_j \alpha_j \beta_j sim(x, d^j) \cdot \delta(f(d^j), y_i) \quad (1)$$

$$\boldsymbol{sim}(\boldsymbol{x}, \boldsymbol{d}) = \begin{cases} \dfrac{\mathbf{1}}{\sqrt{\sum_i \boldsymbol{w_i} \cdot \boldsymbol{dis^2}(\boldsymbol{x_i}, \boldsymbol{d_i})}} & \boldsymbol{x \neq d} \\ \boldsymbol{S} & \boldsymbol{x = d} \end{cases} \quad (2)$$

$$w_i \propto c(\boldsymbol{D}_i) H(\boldsymbol{D}_i, f(\boldsymbol{D})) \quad (3)$$

4 Implementation in Flight Domain

The framework has been implemented in the flight reservation domain. A grammar was used to parse the user's input, and a set of generation rules was used to convert the DM's output E-form into natural language (Seneff, 2002). Two local KSes are utilized: one resolves complex date and time expressions, and one looks up airport/city codes. A local simulated flight DB will be replaced by a real external one in the future.

Figure 3 illustrates the logic of the flight reservation domain. The database has two alternative sets of conjunctive constraints "destination & source & date" and "flight# & date". Two entity types are defined. The itinerary entity type contains a list of flights, a number of expected flights and a price, with completion condition "#flights > 0". The flight entity type contains members: flight number, date, source, destination, *etc.*, with completion condition "flight# & date".

Table 1 illustrates dialogue planning. In the execution of *flight.complete_entity()*, the DM determines that it needs a flight number according to the entity's completion condition. However, a destination is required to search the flight DB. No other KS offers this information, so the system turns to the user to ask for the destination.

The statistical engine currently supports inference for two problems: whether the execution of a task requires the user's confirmation, and whether the pending list is in focus.

Several customized task actions were defined for the domain. For example, after adding the first flight, a customized task action will automatically create a return flight with appropriate source and destination, unless a one-way trip has been indicated. The implementation of the customized task actions required only about 550 lines of code.

User: I want a flight to Chicago
create itinerary
itinerary.complete_entity()
itinerary.add_entity(:flights)
create flight
flight.complete_entity()
flight.fill_attribute(flight#)
inquire_ks(flight_db, flight#)
flight.fill_attribute(destination)
inquire_ks(user, destination)
System: What city does the flight leave from?

Table 1. An example of the system's reasoning process. Shaded lines denote statistical decisions.

5. Preliminary Evaluation

We conducted a preliminary evaluation with a simulated flight database and a simulated user model. The statistical inference model was trained with 210 turns from 18 conversations. A personality-based user simulator creates random scenarios and simulates user utterances using a probabilistic template-based method. In 50 conversations between the simulated user and the DM, the average number of turns was 14.58, with a high standard deviation of 8.2, due to the variety of the scenario complexity and personalities of the simulator users. Some simulated users

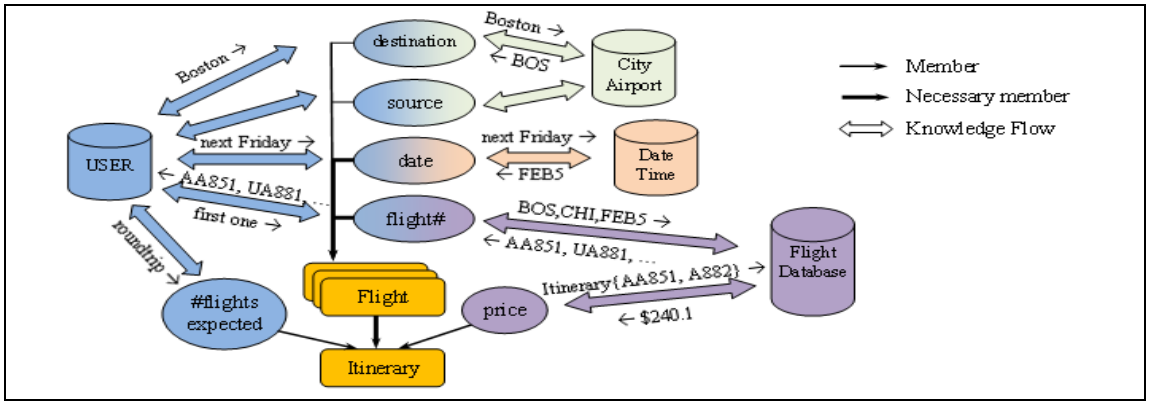


Figure 3. Dialogue Logic for the Flight Booking Domain.

were intentionally designed to be very uncooperative. The DM was able to handle these situations most of the time.

We examined all the simulated dialogues turn by turn. For a total of 729 turns, the DM responded appropriately 92.2% of the time. One third of the failed turns were due to parse failures. Another third resulted from insufficient tutoring. These situations were not well covered in the tutoring phase, but can be easily fixed through a few more manual corrections. The rest of the errors came from various causes. Some were due to defects in the simulator.

6 Conclusions and Future Work

We have introduced a framework for goal-based dialogue planning. It treats the user as a knowledge source, so that the entire framework is DM-centered. A declarative entity-based specification encodes the domain logic simply and clearly. Customized task actions handle any domain-dependent computations, which are kept at a minimum. A simple statistical engine built into the framework offers more flexibility.

In the future, we will integrate the dialogue manager into a speech-enabled framework, and build spoken dialogue systems for flight reservations and other domains of interest.

Acknowledgments

This research is funded by Quanta Computers, Inc., through the T-Party project.

References

Bohus, D., & Rudnicky, A. I. (2003). RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda. *Proc. Eurospeech.* Geneva, Switzerland.

Bühler, D., & Minker, W. (2005). A REASONING COMPONENT FOR INFORMATION-SEEKING AND PLANNING DIALOGUES. *Spoken Multimodal Human-Computer Dialogue in Mobile Environments , 28*, 77-91.

Frampton, M., & Lemon, O. (2006). Learning more effective dialogue strategies using limited dialogue move features. *Proc. ACL*, (pp. 185 - 192). Sidney, Australia.

Hochberg, J., Kambhatla, N., & Roukos, S. (2002). A flexible framework for developing mixed-initiative dialog systems. *Proc. the 3rd SIGdial workshop on Discourse and dialogue* , (pp. 60-63). Philadelphia, Pennsylvania .

Larsson, S., & Traum, D. (2000). Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering , 6* (3-4), 323-340.

Lemon, O., & Pietquin, O. (2007). Machine learning for spoken dialog systems. *Proc. INTERSPEECH 2007*, (pp. 2685–2688). Antwerp, Belgium.

Levin, E., Pieraccini, R., & Eckert, W. (1997). Learning Dialogue Strategies within the Markov Decision Process Framework. *Proc. ASRU 1997.* Santa Barbara, USA.

Milward, D., & Beveridge, M. (2004). Ontologies and the Structure of Dialogue. *Proc. of the Eighth Workshop on the Semantics and Pragmatics of Dialogue*, (pp. 69-76). Barcelona, Spain.

Scheffler, K., & Young, S. (2001). Corpus-based dialogue simulation for automatic strategy learning and evaluation. *Proc. NAACL Workshop on Adaptation in Dialogue.* Pittsburgh, USA.

Seneff, S. (2002). Response Planning and Generation in the Mercury Flight Reservation System. *Computer Speech and Language , 16*, 283-312.

Smith, R. W., Hipp, D. R., & Biermann, A. W. (1995). An architecture for voice dialog systems based on prolog-style theorem proving. *Computational Linguistics , 21* (3), 281-320.

Williams, J. D., & Young, S. (2007). Partially observable Markov decision processes for spoken dialog systems. *Computer Speech & Language , 21* (2), 393-422.

Zue, V., Seneff, S., Glass, J., Polifroni, J., Pao, C., Hazen, T. J., et al. (2000). JUPITER: a telephone-based conversational interface for weather information. *IEEE Transactions on Speech and Audio Processing , 8* (1), 85-96.

Towards Improving the Naturalness of Social Conversations with Dialogue Systems

Matthew Marge, João Miranda, Alan W Black, Alexander I. Rudnicky
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{mrmarge,jmiranda,awb,air}@cs.cmu.edu

Abstract

We describe an approach to improving the naturalness of a social dialogue system, Talkie, by adding disfluencies and other content-independent enhancements to synthesized conversations. We investigated whether listeners perceive conversations with these improvements as natural (i.e., human-like) as human-human conversations. We also assessed their ability to correctly identify these conversations as between humans or computers. We find that these enhancements can improve the perceived naturalness of conversations for observers "overhearing" the dialogues.

1 Introduction

An enduring problem in spoken dialogue systems research is how to make conversations between humans and computers approach the naturalness of human-human conversations. Although this has been addressed in several goal-oriented dialogue systems (e.g., for tutoring, question answering, etc.), *social* dialogue systems (i.e., non-task-oriented) have not significantly advanced beyond so-called "chatbots". Proper social dialogue systems (Bickmore and Cassell, 2004; Higuchi et al., 2002) would be able to conduct open conversations, without being restricted to particular domains. Such systems would find use in many environments (e.g., human-robot interaction, entertainment technology).

This paper presents an approach to improving a social dialogue system capable of chatting about the news by adding content-independent enhancements to speech. We hypothesize that enhancements such as explicit acknowledgments (e.g., *right*, *so*, *well*) and disfluencies can make human-computer conversations sound indistinguishable from those between two humans.

Enhancements to synthesized speech have been found to influence perception of a synthetic voice's hesitation (Carlson et al., 2006) and personality (Nass and Lee, 2001). Andersson et al. (2010) used machine learning techniques to determine where to include conversational phenomena to improve synthesized speech. Adell et al. (2007) developed methods for inserting filled pauses into synthesized speech that listeners found more natural. In these studies, human judges compared utterances in isolation with and without improvements. In our study, we focus on a holistic evaluation of naturalness in dialogues and ask observers to directly assess the naturalness of conversations that they "overhear".

2 The Talkie System

Talkie is a spoken dialogue system capable of having open conversations about recent topics in the news. This system was developed for a dialogue systems course (Lim et al., 2009). Interaction is intended to be unstructured and free-flowing, much like social conversations. Talkie initiates a conversation by mentioning a recent news headline and invites the user to comment on it.

The system uses a database of news topics and human-written comments from the "most blogged about articles" of the New York Times (NYT)[1]. Comments are divided into single sentences to approximate the length of a spoken response. Given a user's utterance (e.g., keywords related to the topic), Talkie responds with the comment that most closely resembles that utterance. Talkie may access any comment related to the topic under discussion (without repetition). The user may choose to switch to a different topic at any time (at which point Talkie will propose a different topic from its set).

[1] http://www.nytimes.com/gst/mostblogged.html
Follow links to each article's comment section.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 91–94,
The University of Tokyo, September 24-25, 2010.

3 Study

We performed a study to determine if the perceived naturalness of conversations could be improved by using heuristic enhancements to speech output. Participants "overheard" conversations (similar to Walker et al. (2004)). Originally typed interactions, the conversations were later synthesized into speech using the Flite speech synthesis engine (Black and Lenzo, 2001). For distinctiveness, conversations were between one male voice (rms) and one female voice (slt). The voices were generated using the CLUSTERGEN statistical parametric synthesizer (Black, 2006). All conversations began with the female voice.

3.1 Dialogue Content

We considered four different conversation types: (1 & 2) between a human and Talkie (human-computer and computer-human depending on the first speaker), (3) between two humans on a topic in Talkie's database (human-human), and (4) between two instances of Talkie (computer-computer). The human-computer and computer-human conditions differed from each other by one utterance; that is, one was a shifted version of the other by one dialogue turn. The human-computer conversations were collected from two people (one native English speaker, one native Portuguese speaker) interacting with Talkie on separate occasions. For human-human conversations, Talkie proposed a topic for discussion. Each conversation contained ten turns of dialogue. To remove any potential effects from the start and end content of the conversations, we selected the middle three turns for synthesis. Each conversation type had five conversations, each about one of five recent headlines (as of May 2010).

3.2 Heuristic Enhancements

We defined a set of rules that added phenomena observed in human-human spoken conversations. These included filled pauses, word repetitions, silences, and explicit acknowledgments. Conversations in this study were enhanced manually by following the set of rules described in Figure 1; an example is shown in Figure 2.

3.3 Participants and Task

Eighty participants were recruited from Amazon's Mechanical Turk[2] (MTurk) for this between-subjects study. Only workers with a 95% HIT (human intelligence task) approval rating could participate. There were three experimental dimensions that we counterbalanced across participants: the four conversation "types" described in Section 3.1, the order of the five topics, and the presence or absence of heuristic enhancements to the synthesized speech that participants heard. Each participant listened to every topic such that exactly five participants heard each condition.

Participants made two judgments per conversation: rating the "naturalness" of a conversation based on a four-point Likert scale and ranking the conversant identities (i.e., as two humans, two computers, or a computer and a human) from most to least confident. We used the non-parametric Kruskal-Wallis one-way analysis of variance to determine statistically significant differences in naturalness responses across groups (naturalness being the dependent measure).

Category I - *Explicit Acknolwedgements*

- inserted sparingly at the beginning of sentences when grammatical (e.g., *well*, *so*, *you know*, *right*).

Category II - *Filled pauses / repetitions*

- no more than three per dialogue
- no more than two in any one sentence
- placed repetitions in places where speaker may have hesitation (e.g., after "I think...")
- utterances may have a filled pause or a repetition followed by a filled pause.

Category III - *Silences*

- added sparingly after filled pauses.

Figure 1: Heuristics used for adding enhancements to synthesized dialogue.

4 Results

We analyzed two measures: (1) naturalness ratings of conversations, and (2) frequencies from highest-rank to lowest-rank of who the participant believed to be the speakers in each conversation. We found no statistically significant ordering effects or biases due to the topic of conversation.

4.1 Speaker Type Condition Results

Results for naturalness scores are summarized in Figure 3. As expected, mean naturalness was highest for the human-human condition and lowest for the computer-computer condition. We found

[2] http://www.mturk.com

Human-computer conversation (original, **enhancements)**

h: Well, that sounds ridiculous, you get taxed proportionally to your paycheck.
c: So, let them retire on social security payments, then you will see changes with <**uh**> **with** an overabundance of funds.
h: I don't see how this would be possible.
c: You know, you don't have the money yet, and you have just increased your debt by 2.5 trillion dollars.
h: Exactly why should we be raising taxes on the rich?
c: With our current politics of division, I <**umm**> worry we have the will or determination to address these issues.

Figure 2: Example conversation with heuristic enhancements marked in bold.

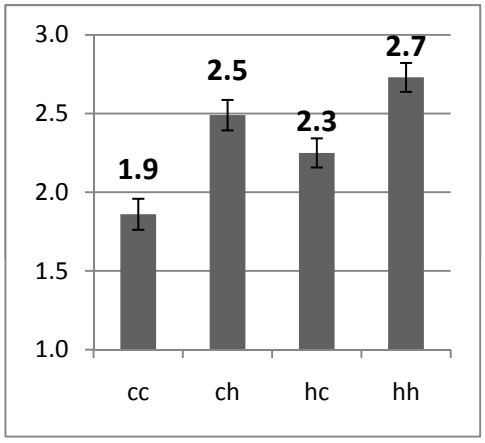


Figure 3: Naturalness across the speaker type condition.

no statistically significant difference in naturalness ratings for the computer-human condition compared to the human-computer condition (H(1) = 2.94; p = 0.09). Also, the computer-computer condition was significantly different from all other conditions, suggesting that conversation flow is an important factor in determining the naturalness of a conversation (H(3) = 42.49, $p < 0.05$).

People rated conversations involving a computer and a human similarly to human-human conversations (without enhancements). There were no statistically significant differences between the three conditions *cc*, *ch*, and *hc* (H(2) = 5.36, p = 0.06). However, a trend indicated that *hc* naturalness ratings differed from those of the *ch* and *hh* conditions. Conversations from the *hc* condition had much lower (18%) mean naturalness ratings compared to their *ch* counterparts, even though they were nearly equivalent in content.

4.2 Heuristic Enhancements Results

There were significant differences in naturalness ratings when heuristic enhancements were present (H(1) = 17.49, $p < 0.05$). Figure 4 shows that the perceived naturalness was on average higher with heuristic enhancements. Overall, mean naturalness improved by 20%. This result agrees with findings from Andersson et al. (2010).

Computer-computer conversations had the highest relative improvement (42%) in mean naturalness. Naturalness ratings were significantly different when comparing these conversations with and without enhancements (H(1) = 11.77, $p < 0.05$). Content-free conversational phenomena appear to compensate for the lack of logical flow in these conversations. According to Figure 5, after enhancements people are no better than chance at correctly determining the speakers in a computer-computer conversation. Thus the heuristic enhancements clearly affect naturalness judgments.

Even the naturalness of conversations with good logical flow can improve with heuristic adjustments; there was a 26% relative improvement in the mean naturalness of human-human conversations. Participant ratings of naturalness were again significantly different (H(1) = 12.45, $p < 0.05$). Note that these conversations were originally typed dialogue. As such, they did not capture turn-taking properties present in conversational speech. When enhanced with conversational phenomena, they more closely resembled natural spoken conversations. As shown in Figure 5, people are more likely than chance to correctly identify two humans as being the participants in the dialogue after these enhancements were applied to speech.

Conversations with one computer and one human also benefited from heuristic enhancements. Improvements in naturalness were marginal, however. Naturalness scores in the *hc* condition improved by 16%, but this improvement was only a trend (H(1) = 3.66, p = 0.06). Improvement was negligible in the *ch* condition. Participants selected the correct speakers in human-computer dialogues no better than random. We note that participants tended to avoid ranking conversations as "human & computer" with confidence (i.e., the highest rank). A significant majority (267 out of 400) of second-rank selections were "human & computer." Participants tended to order conditions

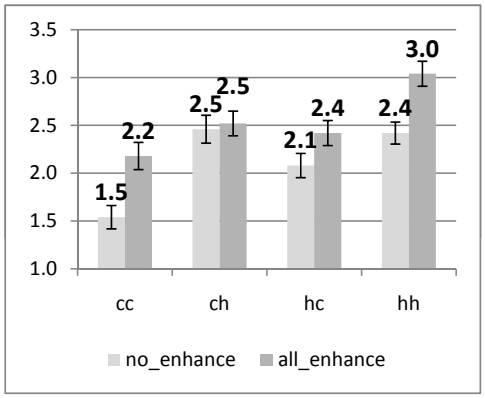


Figure 4: Mean naturalness across enhancement conditions.

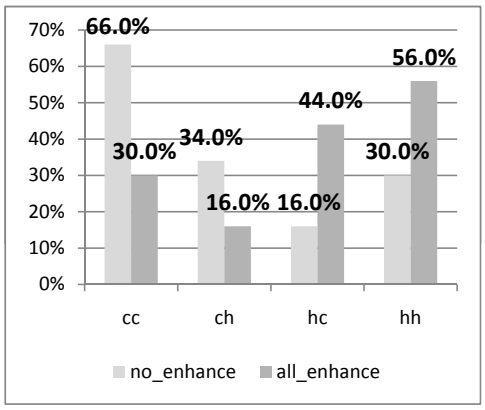


Figure 5: Percentage of participants' selections of members of the conversation that were correct.

from all human to all computer or vice-versa.

5 Conclusions

We have shown that content-independent heuristics can be used to improve the perceived naturalness of conversations. Our conversations sampled a variety of interactions using Talkie, a social dialogue system that converses about recent news headlines. An experiment examined the factors that could influence how external judges rate the naturalness of these conversations.

We found that without enhancements, people rated conversations involving a human and a computer similarly to conversations involving two humans. Adding heuristic enhancements produced different results, depending on the conversation type: computer-computer and human-human conversations had the best gain in naturalness scores. Though it remains to be seen if people are always influenced by such enhancements, they are clearly useful for improving the naturalness of human-computer dialogues.

Future work will involve developing methods to automatically inject enhancements into the synthesized speech output produced by Talkie, as well as determining whether other types of systems can benefit from these techniques.

Acknowledgments

We would like to thank Aasish Pappu, Jose-Pablo Gonzales Brenes, Long Qin, and Daniel Lim for developing the Talkie dialogue system.

References

J. Adell, A. Bonafonte, and D. Escudero. *Filled pauses in speech synthesis: Towards conversational speech.* In TSD'07, Pilsen, Czech Republic, 2007.

S. Andersson, K. Georgila, D. Traum, M. Aylett, and R.A.J. Clark. *Prediction and realisation of conversational characteristics by utilising spontaneous speech for unit selection.* In the 5th International Conference on Speech Prosody, Chicago, Illinois, USA, 2010.

T. Bickmore and J. Cassell. *Social Dialogue with Embodied Conversational Agents.* J. van Kuppevelt, L. Dybkjaer, and N. Bernsen (eds.), Natural, Intelligent and Effective Interaction with Multimodal Dialogue Systems. New York: Kluwer Academic.

A. Black. CLUSTERGEN*: A Statistical Parametric Synthesizer using Trajectory Modeling.* In Interspeech'06 - ICSLP, Pittsburgh, PA, 2006.

A. Black and K. Lenzo. *Flite: a small fast run-time synthesis engine.* In ISCA 4th Speech Synthesis Workshop, Scotland, 2001.

R. Carlson and K. Gustafson and E. Strangert. *Cues for Hesitation in Speech Synthesis.* In Interspeech'06 - ICSLP, Pittsburgh, PA, 2006.

S. Higuchi, R. Rzepka, and K. Araki. *A casual conversation system using modality and word associations retrieved from the web.* In EMNLP'08. Honolulu, Hawaii, 2008.

D. Lim, A. Pappu, J. Gonzales-Brenes, and L. Qin. *The Talkie Spoken Dialogue System.* Unpublished manuscript, Carnegie Mellon Univeristy, 2009.

C. Nass and K. M. Lee. *Does computer-synthesized speech manifest personality? Experimental tests of recognition, similarity-attraction, and consistency-attraction.* Journal of Experimental Psychology: Applied 7 (2001) 171-181.

M. A. Walker, S. J. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, G. Vasireddy. *Generation and evaluation of user tailored responses in multimodal dialogue.* Cognitive Sci. 28 (2004) 811-840.

Route Communication in Dialogue: a Matter of Principles

Theodora Koulouri
Department of Information Systems and Computing
Brunel University
Middlesex UB8 3PH
theodora.koulouri@brunel.ac.uk

Stanislao Lauria
Department of Information Systems and Computing
Brunel University
Middlesex UB8 3PH
stasha.lauria@brunel.ac.uk

Abstract

The present study uses the dialogue paradigm to explore route communication. It revolves around the analysis of a corpus of route instructions produced in real-time interaction with the follower. It explores the variation in forming route instructions and the factors that contribute in it. The results show that visual co-presence influences the performance, conversation patterns and configuration of instructions. Most importantly, the results suggest an analogy between the choices of instruction-givers and the communicative actions of their partners.

1.1 Spatial language in dialogue

The main question this paper attempts to address is how people produce route instructions in dialogue. The current zeitgeist in language research and dialogue system development seems to be the unified investigation of spatial language and dialogue (Coventry et al., 2009). Indicative of the growing prioritisation of dialogue in the study of spatial language are the on-going research efforts within the MapTask[1] project and the GIVE challenge[2] .

1.2 A framework for the analysis of route instructions

The study uses CORK (Communication of Route Knowledge, (Allen 2000)), a framework which provides a component-based analysis of route instructions. The CORK taxonomy differentiates between instructions that are directives (action statements with verbs of movement) and descriptive statements (with state-of-being verbs, like "be" and "see"). Descriptives present a static picture of spatial relations and provide the follower the opportunity to verify his position or reorient himself. The taxonomy also considers elements that provide specificity and distinguishing information about environmental features, called delimiters. Within this framework, Allen (2000) describes a set of principles pertaining to the configuration of route descriptions. Namely, people concentrate descriptives and delimiters on points along the route that offer for uncertainty (like crossroads). Moreover, the selection and placement of these components depends on the characteristics of the environment and the perceived needs of the follower. Evidence from empirical work supports the framework, reporting that errors in navigation increased when the route directions violated these principles. Nevertheless, the applicability of the suggested principles has only been tested in scenarios in which the directions were produced beforehand by either the experimenters or a separate group of subjects.

1.3 The effect of visual information

Studies exploring the effect of visual information on task-oriented interaction converge on that visual feedback leads to more efficient interactions and influences the conversational patterns between participants (Clark and Krych, 2004; Gergle et al., 2004; Koulouri and Lauria, 2009). These phenomena are generally attributed to the ease of establishing "common ground" when visual feedback is available. However, to the authors' knowledge, most related studies have focused on high-level analysis of dialogue acts and many aspects of how interlocutors adapt their linguistic choices remain undefined.

1.4 Aims and hypotheses of study

The present study provides an empirical account of route instructions, as they emerge in real-time interaction with the follower. We offer the fol-

[1] http://www.hcrc.ed.ac.uk/maptask/
[2] http://www.give-challenge.org/research/

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 95–98,
The University of Tokyo, September 24-25, 2010.

lowing tentative hypotheses. Since visual co-presence facilitates grounding of information, it is expected to have a major effect on how route instructions are configured. Next, putting additional emphasis on the inter-individual processes involved in language use, this study aims to test whether the linguistic options mobilised by the instructor ultimately depend upon the contributions of the follower.

2 Methods

A study was designed to elicit natural route instructions in a restricted context. Pairs of participants collaborated in a navigation task, in a "Wizard-of-Oz" set-up. The instructors provided instructions to navigate their partners to designated locations in a simulated town, being under the impression that they were interacting with a software agent (robot). The study manipulated two factors; i) availability of visual information on follower's actions and ii), follower's interactive capacity. With regard to the first factor, there were two conditions in which the ability to monitor the actions of the "robot" was either removed or provided. The second factor also involved two conditions. In the first condition, the followers could interact using unconstrained language (henceforth, "Free" condition). In the second condition (henceforth, "Constrained" condition), a set of predetermined responses available to the followers aimed to coerce them towards more "automated" contributions; for instance, "open-ended" repairs such as "What?", which provide no specific information on the source of the problem. However, the followers were still able to be interactive if they wished so, by clicking the relevant buttons to request clarification or provide location information.

The study followed a between-subjects factorial design. A total of 56 students were allocated in the four conditions: Monitor-Free, Monitor-Constrained, No Monitor-Free, No Monitor-Constrained. The experimental procedure is described in detail in (Koulouri and Lauria, 2009).

2.1 Set-up

The experiment relied on a custom-built system which supported the interactive simulation and enabled real-time direct text communication between the pairs. The interfaces consisted of a graphical display and a dialogue box.

The interface of the instructor displayed the full map of the simulated town (Figure 1). On the upper right corner of the interface, there could be a small "monitor", in which the robot's immediate locality was displayed, but not the robot itself. The presence of the monitor feature depended on the experimental condition.

The followers' interface displayed a fraction of the map, the surroundings of the robot's current position. The robot was operated by the follower using the arrow keys on the keyboard. In the "Free" conditions followers could freely type messages. In the "Constrained" conditions, the followers needed to use the buttons on the interface (Figure 2).

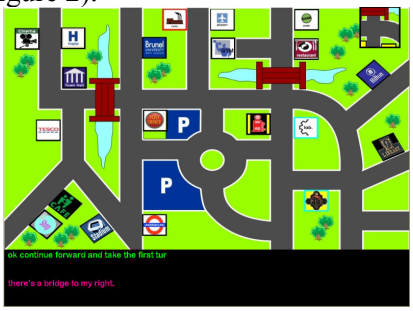


Figure 1. The instructor's interface in the Monitor conditions. The monitor window on the upper right corner was removed in No Monitor conditions.

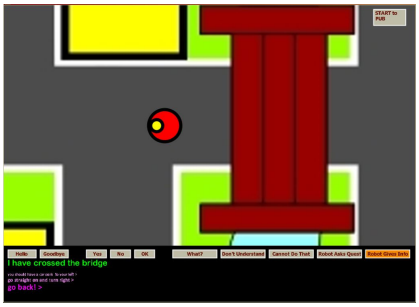


Figure 2. The follower's interface in the Constrained conditions. In the Free conditions, there were no buttons and followers could freely type any message.

2.2 Data analysis

The analysis of the corpus of route instructions followed the CORK framework (Allen, 2000). Communicative statements were classified as **Directives** or **Descriptives**. These communicative statements could contain references to environmental features. The types of environmental features considered were: **Locations** (e.g., buildings or bridges), **Pathways** (e.g., streets), **Choice Points** (e.g., junctions) and **Destination**. Last, instructions can be composed of delimiters, which fall into four categories:

1. **Distance designations:** e.g., "*...until* you see a car park".
2. **Direction designations:** e.g., "go *left"*.
3. **Relational terms:** e.g., *"on your left",*
4. **Modifiers:** e.g., "*big red* bridge", "take the *first/second/last* road".

3 Results

The experiment yielded a large corpus of 160 dialogues, composed of 3,386 turns. 1,485 instructions were collected. First, the analysis considers some common measures of efficiency. Next, the results of the component analysis of instructions are presented.

3.1 Efficiency of interaction

The number and length of turns and instructions and time needed to complete each task are typically used as measures of the efficiency of interaction. Additionally, fewer execution and understanding failures are taken as indicators of superior performance.

Time, number of turns, words and instructions: The ANOVA performed on time per task showed no reliable significant differences among groups. On the other hand, significant effects were observed with regard to all other dependent variables. An interaction effect was revealed after analysis on numbers of turns ($F(1, 24) = 3.993$, $p = .05$). Pairs in the Monitor-Free condition required less turns to complete the task compared to the other groups (column 1 of Table 1). It seems however that instructors in both Monitor conditions were dominating the conversational floor, having produced about 58% of the turns, compared to instructors in the No Monitor conditions ($F(1, 24) = 5.303$, $p = .03$). Nevertheless, it was not the case that instructors in Monitor conditions were "wordier". The number of words was similar among all instructor groups. The results indicated that the total number of words required to complete a task was much lower in Monitor conditions ($F(1, 24) = = 5.215$, $p = 0.03$) (see column 3 in Table 1). Next, instructors in Monitor conditions gave more instructions to guide the followers to the destination ($F(1, 24) = 3.494$, $p = .07$). However, these instructions were considerably shorter compared to the instructions provided by No Monitor instructors ($F(1, 24) = 4.268$, $p = .05$). All differences are amplified in the Monitor-Constrained group, in which more turns and instructions were needed but with fewer words and the "turn possession" of the instructor was the highest among the groups.

Condition	#Turns per task	%Instructor Turns	#Words per task	#Words per Instruction	#Instructions per task	Miscommunication per task
M-F	16.74	57.12%	87.33	4.70	9.08	1.14
M-C	23.95	58.86%	65.02	3.01	11.73	2.05
NM-F	23.63	52.28%	105.38	5.29	8.58	1.20
NM-C	20.15	50.62%	100.35	5.07	7.68	0.69

Table 1. Summary of Results (mean values).

Frequency of miscommunication: Miscommunication was calculated by considering two measures: the number of execution errors and of follower turns that were tagged as expressing non-understanding. The ANOVA revealed an interaction effect ($F(1, 24) = 4.012$, $p = .05$). Striking differences were observed between the Monitor-Constrained group and the rest; in particular, followers in this condition were twice or three times more likely to fail to understand and execute instructions (see last column in Table 1).

3.2 Component analysis of instructions

This section presents the results of the analysis on inclusion of landmark references, types of delimiters and communicative statements.

Landmark references: Instructors in both No Monitor conditions preferred to produce instructions that were anchored on landmarks, especially on 3D locations such as buildings (28% of instructions contained locations vs 14% in the Monitor conditions, ($F(1, 24) = 12.034$, $p = .002$)). On the other hand, Monitor instructors opted for simple action prescriptions. Particularly, 75% of the instructions in the Monitor-Constrained condition omitted any kind of reference (compared to an average of 42% in the other conditions).

Delimiters: Category 2 delimiters that provided simple direction information were prevalent in Monitor conditions ($F(1, 24) = 11.407$, $p = .002$). Further, an interaction effect was found ($F(1, 24) = 3.802$, $p = .01$); the number of category 2 delimiters almost doubles in the Monitor-Constrained condition. On the contrary, the use of category 1 delimiters, which provide information on the boundary of the route is very limited in the Monitor-Constrained condition ($F(1, 24) = 5.350$, $p = .03$). The third category of delimiters includes terms that specify the relation between traveller and an environmental feature ("on your left") or between environmental features. Again, the difference arises in the Monitor-Constrained condition, which included the lowest number of category 3 delimiters (marginal effect, $F(1, 24) = 3.392$, $p = .07$). Finally, the analysis performed on the frequency of category 4 delimiters did not yield any significant effect.

Directive and descriptive communicative statements: An interaction effect was revealed with regard to the proportion of directives and

descriptives in the corpus (F (1, 24) = 3.830, p = .06). The instructors in the Monitor-Constrained condition produced less descriptives, which give information about relations among features in the environment and tap perceptual experience ("you will see a bridge"). In particular, in the Monitor-Constrained condition, 4.7% of instructions were descriptives, whereas the proportion of descriptives averaged 10% in all other conditions.

4 Discussion

The results resonate with previous research. The actions of the followers served as an immediate, accurate and effortless indicator of their current state of understanding, making verbal feedback redundant. Monitor instructors could readily confirm their assumptions about the information requirements of followers and used linguistics shortcuts and simpler instructions exactly at the moment needed. On the other hand, in the No Monitor condition, uncertainty about the position and movement of the robot created the need for elaborate and explicit instructions. The contribution of the present study lies on that it grounds these observations on quantitative analysis, using measures like words, turns and the relative frequencies of certain types of instruction components that vary in information value. Most importantly, it describes the specific ways in which instructors configure their directions in the presence/absence of visual information.

The CORK framework predicts that route protocols which are rich in descriptives and relational terms are associated with more successful navigation, compared to simple directional ones. Our results partially meet this expectation, since large numbers of execution errors and non-understandings were only observed in the Monitor-Constrained condition, whereas miscommunication rates were similar across the other groups. Indeed, this condition was found to generally differ from the rest. In particular, In the Monitor-Constrained condition, the dialogues were the shortest in terms of words. Instructors produced many but short instructions. The component-based analysis revealed that they employed overwhelmingly more action-based instructions without landmark references and descriptives. Boundary information on the route, frame of reference and spatial relations between environmental features were typically omitted. In both Constrained conditions, followers were expected to resort to a "mechanical" interaction, as coerced by the presence of the predefined responses. Inspection of the dialogues revealed that followers in the Monitor-Constrained condition did so, given the precedence of visual feedback. This was not the case with No Monitor-Constrained followers who needed to verbally ground information. Dialogue examples are provided in Table 2 below.

I: turn around I: go straight ahead I: stop I: turn left here I: go ahead F: *What?* I: Go straight ahead	I: Now keep going down the road until you see a car park F: *I am in front of the car park* I: turn right and walk till the end, along the road you will see a gym on your right F: *yes gym to my right side* I: good, keep going straight and you will see a factory on your left

Table 2. Dialogue excerpts from the Monitor-Constrained (column 1) and No Monitor-Constrained (column 2) conditions.

Thus, we propose that the linguistic choices of the followers "prime" the instructor's own strategies. In the Monitor-Constrained Condition, followers were less interactive, and gave fewer responses with lower information value. In harmony, their partners provided less elaborate instructions, which also lacked important information and specificity.

In conclusion, the findings confirm our initial hypotheses. Instructions are sensitive to conditions of (visual) co-presence. Moreover, a direct link was identified between the way in which instructions and follower's contributions are formulated. Following this lead, we are now focusing on a fine-grained analysis of the utterances of the follower.

References

Darren Gergle, Robert E. Kraut and Susan E. Fussell. 2004. Language Efficiency and Visual Technology: Minimizing Collaborative Effort with Visual Information. *Journal of Language and Social Psychology,* 23(4):491-517. Sage Publications, CA.

Gary L. Allen. 2000. Principles and Practices for Communicating Route Knowledge. *Applied Cog. Psychology.*14(4):333–359.

Herbert H. Clark and Meredyth A. Krych. 2004. Speaking While Monitoring Addressees for Understanding. *J. of Memory and Language*, 50:62-81.

Kenny Coventry, Thora Tenbrink and John Bateman, 2009. Spatial Language and Dialogue: Navigating the Domain. In K. Coventry, T. Tenbrink, and J. Bateman (Eds.) Spatial Language and Dialogue. 1-8. Oxford University Press. Oxford, UK.

Theodora Koulouri and Stanislao Lauria. 2009. Exploring Miscommunication and Collaborative Behaviour in Human-Robot Interaction, *SIGdial09*.

The Impact of Dimensionality on Natural Language Route Directions in Unconstrained Dialogue

Vivien Mast
University of Bremen
Bremen, Germany
viv@tzi.de

Jan Smeddinck
University of Bremen
Bremen, Germany
smeddinck@tzi.de

Anna Strotseva
University of Bremen
Bremen, Germany
anna.strotseva@uni-bremen.de

Thora Tenbrink
University of Bremen
Bremen, Germany
tenbrink@uni-bremen.de

Abstract

In this paper we examine the influence of dimensionality on natural language route directions in dialogue. Specifically, we show that giving route instructions in a quasi-3d environment leads to experiential descriptive accounts, as manifested by a higher proportion of location descriptions, lack of chunking, use of 1st person singular personal pronouns, and more frequent use of temporal and spatial deictic terms. 2d scenarios lead to informative instructions, as manifested by a frequent use of motion expressions, chunking of route elements, and use of mainly 2nd person singular personal pronouns.

1 Introduction

In order to build artificial agents that are competent in creating and understanding natural language route directions in situated discourse, it is necessary to explore how situatedness affects the communication of humans about routes. The current study aims at exploring in which ways dimensionality influences the choice of communicative strategies for route directions in discourse.

Previous research about route directions mostly deals with monologues or pretend dialogue (e.g. Rehrl et al., 2009; Klippel et al., 2003), and concerns two-dimensional stimuli, such as map-based tasks (Klippel et al., 2003; Goschler et al., 2008).

The study presented here examines pairs of participants collaborating on a route instruction task in a naturalistic discourse setting under two conditions: In the 2d condition, the instructor was shown a two-dimensional map with the route drawn into it. In the 3d condition however, the instructor navigated along a preset route in Google Maps Street View.

2 Route Instruction Strategies

Route directions consist of *procedures* and *descriptions* that combine to a step-by-step prescription of the actions that are necessary for executing the given course (Michon and Denis, 2001; Longacre, 1983). Since spatial linguistic expressions reflect the mental model already existing on the part of the instructor, the dimensions in which route instructors experience an environment (2d or 3d) may have a systematic impact on the discourse strategies they use. In the following we analyze a range of spatial descriptions, focusing on aspects known to be crucial for spatial interaction, such as descriptions of locations and motion, the use of perspective expressions, chunking of route elements, and personal and spatiotemporal deixis.

2.1 Static and Dynamic Descriptions

Since route directions deal with a static environment in which a movement takes place, they usually include a high proportion of dynamic descriptions of actions (*procedures* in Michon and Denis' (2001) terms), and additional static information about the surroundings (*descriptions*). In our analysis, we distinguished speakers' utterances as *motion descriptions* if they described or requested the literal *motion* of an entity. In contrast, an utterance was marked as *location* if it described a *static spatial relation*, for example the position of the speaker or an object at a certain point in time.

2.2 Perspective Use

When describing routes, speakers either use the *route perspective*, describing route elements or motions from the point of view of a

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 99–102,
The University of Tokyo, September 24-25, 2010.

person traveling along the route, or the *survey perspective*, where the description is based on cardinal directions, or directions as they are defined by the map as a whole (Taylor and Tversky, 1996). Previous research has indicated that perspective choice can be influenced by the specific situation, and by the coordination between speakers in natural discourse (Pickering and Garrod, 2004; Watson et al., 2006). In the present study, we test the hypothesis that navigating a route in a 3d perspective makes it more difficult for the instructor to use the survey perspective, leading to a preference for the route perspective. Further we assume that the follower will adapt to the instructor's perspective choice in terms of language use.

2.3 Chunking of Route Segments

In a study examining online route descriptions to an imaginary follower based on a two-dimensional map, Klippel et al. (2003) found that participants tended to chunk decision points without directional change together. For example, a speaker could say "turn right at the second intersection" instead of "Go straight on, and then turn right". This occurred even when the route was shown as a moving dot on the map. In our study, we address the question whether this also holds for instructors with a three-dimensional view. We expected a frequent usage of chunking in the 2d condition, in which the participants have access to comprehensive structural information, as opposed to a higher degree of separate references in the 3d condition, in which participants experience the environment incrementally.

3 Experiment

22 students (average age 25, 14 male and 8 female) volunteered to participate in the experiment. They formed 11 pairs that each completed one test run and three permuted critical trials. Instructor and follower were placed in different rooms and interacted via telephone software.

The four predetermined routes were identical for all participants, and they differed mildly in complexity, ranging from 9 to 14 decision points. All routes were located in San Francisco and were specifically designed such that, at most decision points, descriptions would be unambiguous with respect to perspective use.

In the 2d condition (5 pairs), instructors were given a map that showed mostly street names and major landmarks such as parks, schools, restaurants, etc., as they appear in the standard Google Maps map view. The route consisted of a marked starting and end point, and was signaled by a thick blue line with arrows indicating the direction. In the 3d condition (6 pairs), instructors interacted directly with Google Street View which had a photographic quasi-3d view and allowed them to observe the surroundings as if navigating on the roads, seeing a vast amount of details of the environment. Street names were clearly readable as an overlay on top of the photographic imagery. The route was indicated by fat blue arrows that the instructors could click on, in order to move in the given direction.

In both conditions, the followers were asked to draw the route on a map that only contained the starting point. The task instruction was the same for both conditions, priming for procedural discourse yet ambiguous with respect to perspective use: "Now you have to tell your partner where you are going. Please do this by giving instructions via the microphone." (translated from German). In the 3d condition instructors were informed that the follower had a different view of the same surroundings.

Taken together this setup differs from previous studies in that it features unconstrained spoken dialogue and is set in a realistic use-case with a three-dimensional setting.

4 Results

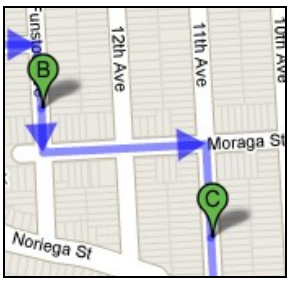
The participants in the 3d condition took significantly longer (M = 125.61 utterances per trial) to complete a task than the participants in the 2d perspective (M = 46.40 utterances per trial, $t(9) = 4.781$, $p = 0.001$).

Figure 1 shows typical examples of the instructors' language in the two conditions. In the 2d condition, instructors as well as followers used survey perspective, as in line 2.2 in Figure 1, significantly more frequently than in the 3d condition (see Table 1). A Chi-square test showed the following results for the instructors: $\chi^2(1) = 200.14$, $p < 0.0001$ and $\chi^2(1) = 91.25$, $p < 0.0001$ for the followers[1]. It is notable that the followers in the 2d condition showed a preference for survey perspective ($\chi^2(1) = 15.38$; $p < 0.0001$), while in the 3d

[1] Mixed, conflated and unclear expressions were excluded from the analysis.

condition they clearly favored route perspective, which was the perspective of the instructor.

3d condition:
1.1 Yes... erm ... now ... there is a crossing again
1.2 Moraga Street
1.3 to the left
1.4 into Moraga Street […]
1.5 then there is a crossing again
1.6 the twelfth
1.7 straight on over there
1.8 So Moraga further



2d condition:
2.1 And then we go down that one up to Moraga Street
2.2 And there we also go right into Moraga Street
2.3 We go through that one up to Eleventh Avenue

Figure 1. Typical examples of instructors' language in the two conditions.

The instructors in the 3d condition used a significantly higher proportion of location descriptions than the instructors in the 2d condition ($t(6.5) = 4.500$, $p = 0.003$). As Table 2 shows, the instructors in the 2d condition relied mainly on motion descriptions (see Figure 1, location descriptions in lines 1.1 and 1.5 as well as motion descriptions in lines 2.1-2.3).

Perspective expressions	3D		2D	
	Instructor	Follower	Instructor	Follower
Route	98.93% (370)	93.33% (112)	50.88% (87)	21.57% (11)
Survey	1.07% (4)	6.67% (8)	49.12% (84)	78.43% (40)
Totals	374	120	171	51

Table 1. Use of perspective expressions in 2d and 3d conditions (absolute values in parentheses).

Chunking of route elements did not occur at all in the 3d condition. In the 2d condition there were 29 intersections that were skipped through chunking, as shown in line 2.3 in Figure 1. This amounts to a mean of 1.9 chunked intersections per route.

Instructors in the 2d condition strongly preferred 2nd person singular pronouns, whereas instructors in the 3d condition showed a preference - though not as strong - for 1st person singular (see Figure 2). Instructors in the 3d condition also used the German formal pronoun *es* 'it' more frequently than those in the 2d condition. This preference is usually displayed in utterances noting the presence of landmarks in the surroundings (e.g. "Da gibt es eine Haltestelle." – "There is a tram stop here.").

Condition	Location	Motion
3D	36.81%	63.19%
2D	14.31%	85.69%

Table 2. Location and motion descriptions by instructor (means per trial).

In the 3d condition, the participants used temporal and spatial deictic terms more frequently than in the 2d condition (*jetzt* 'now' 3d: 7.3 occurrences per 100 utterances, 2d: 2.73. *hier* 'here' 3d: 2.21, 2d: 0.14).

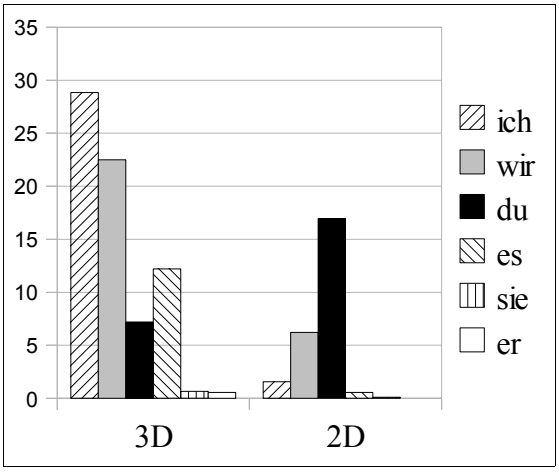


Figure 2. Relative frequency of personal pronouns in the two conditions.

5 Discussion

Our comparison of route directions given while perceiving an environment either as a 2d map or in a 3d view revealed that dimensionality has systematic consequences for discourse strategies on various levels. Location descriptions, route perspective expressions, 1st person singular personal pronouns, impersonal *es* 'it', as well as temporal and spatial deictic terms occurred more frequently in the instructors' discourse in the 3d condition than in the 2d condition. Also, in the 3d condition, instructors did not chunk route elements together. These findings reflect the fact that the instructors

consistently chose a different discourse strategy in this condition. Instead of producing procedural step-by-step instructions, they gave descriptions of the events happening to them and accounts of their surroundings, whereas instructors in the 2d condition gave typical route directions for their partner to follow.

There are three aspects that may be responsible for the different discourse strategies. First, it can be assumed that there is a habitual preference, due to the fact that people providing route directions usually have a 2d representation available to them, or prior knowledge of the relevant route, whereas someone navigating new surroundings would not normally be expected to provide efficient procedural instructions. Second, the lack of structural information in the 3d condition makes it difficult for instructors to describe the route from a survey perspective, or to deliver precise goal-oriented instructions. Third, in the 3d condition, progress for the instructor was slow - comparable to riding a bicycle along the route at moderate speed - due to the technical properties of Google Maps Street View This severed the effect of the inherent lack of structural information, and most probably led the participants to verbalize their progress more frequently than necessary, in order to keep the conversation flowing, instead of to waiting until they reached a point where more efficient instructions would be possible. This factor is also reflected in the number of utterances per trial: The higher number of utterances per trial in the 3d condition (see section 4) is at least partly a result of the technical setup.

In the case of chunking, time does not seem to be the only relevant factor. Klippel et al. (2003) showed that in a 2d scenario in which the route was only gradually revealed in the form of a moving dot on a map, participants still made use of chunking. It remains to be investigated whether the lack of chunking in the present scenario occurred due to the differing dimensionality, or resulted from the unconstrained real dialogue situation, in contrast to the pretend-dialogue used in Klippel et al. (2003).

Further research should differentiate the role of time in the choice of strategy from the impact of perspective. This requires experimental setups that allow for the systematic variation of the speed of the navigation, as well as for better control of such factors as previous knowledge and information density on the route. It would also be necessary to examine two further conditions (instructor: 3d, follower: 2d and instructor: 2d, follower: 3d).

Acknowledgments

Funding by the DFG for the SFB/TR 8, project I5-[DiaSpace], is gratefully acknowledged. We thank the students who participated in our study, as well as Robert Porzel and Elena Andonova for their helpful advice.

References

Juliana Goschler, Elena Andonova and Robert J. Ross. 2008. Perspective Use and Perspective Shift in Spatial Dialogue. In: Christian Freksa et al. (Eds.): *Spatial Cognition VI*, 250-265. Berlin: Springer.

Alexander Klippel, Heike Tappe and Christopher Habel. 2003. Pictorial Representations of Routes: Chunking Route Segments during Comprehension. In: Christian Freksa et al. (Eds.): *Spatial Cognition III*, 11-33. Berlin: Springer.

Robert E. Longacre. 1983. *The Grammar of Discourse*. Plenum Press, New York / London.

Pierre-Emmanuel Michon and Michel Denis. 2001. When and Why Are Visual Landmarks Used in Giving Directions? In: Daniel R. Montello (Ed.): *COSIT 2001*, 292-305.

Martin J. Pickering and Simon Garrod. 2004. Toward a Mechanistic Psychology of Dialogue. In *Behavioral and Brain Sciences*, 27:169-225.

Karl Rehrl, Sven Leitinger, Georg Gartner and Felix Ortag. 2009. An Analysis of Direction and Motion Concepts in Verbal Descriptions of Route Choices. In: Kathleen Stewart Hornsby, Christophe Claramunt, Michel Denis and Gérard Ligozat (Eds.): *COSIT 2009,* 471-488. Springer-Verlag, Berlin / Heidelberg.

Holly A. Taylor and Barbara Tversky. 1996. Perspective in Spatial Descriptions. In *Journal of Memory and Language*, 35:371-391.

Matthew E. Watson, Martin J. Pickering and Holly P. Branigan. 2006. Why Dialogue Methods are Important for Investigating Spatial Language. In: Kenny R. Coventry, John Bateman and Thora Tenbrink (Eds.): *Spatial Language in Dialogue*, 8-22. Oxford University Press, Oxford.

Learning Dialogue Strategies from Older and Younger Simulated Users

Kallirroi Georgila
Institute for Creative Technologies
University of Southern California
Playa Vista, USA
kgeorgila@ict.usc.edu

Maria K. Wolters
School of Informatics
University of Edinburgh
Edinburgh, UK
maria.wolters@ed.ac.uk

Johanna D. Moore
School of Informatics
University of Edinburgh
Edinburgh, UK
J.Moore@ed.ac.uk

Abstract

Older adults are a challenging user group because their behaviour can be highly variable. To the best of our knowledge, this is the first study where dialogue strategies are learned and evaluated with both simulated younger users and simulated older users. The simulated users were derived from a corpus of interactions with a strict system-initiative spoken dialogue system (SDS). Learning from simulated younger users leads to a policy which is close to one of the dialogue strategies of the underlying SDS, while the simulated older users allow us to learn more flexible dialogue strategies that accommodate mixed initiative. We conclude that simulated users are a useful technique for modelling the behaviour of new user groups.

1 Introduction

State-of-the-art statistical approaches to dialogue management (Frampton and Lemon, 2006; Williams and Young, 2007) rely on having adequate training data. Dialogue strategies are typically inferred from data using Reinforcement Learning (RL), which requires on the order of thousands of dialogues to achieve good performance. Therefore, it is no longer feasible to rely on data collected with real users. Instead, training data is generated through interactions of the system with simulated users (SUs) (Georgila et al., 2006). In order to learn good policies, the behaviour of the SUs needs to cover the range of variation seen in real users (Georgila et al., 2006; Schatzmann et al., 2006). Furthermore, SUs are critical for evaluating candidate dialogue policies.

To date, SUs have been used to learn dialogue strategies for specific domains such as flight reservation, restaurant recommendation, etc., and to learn both how to collect information from the user (Frampton and Lemon, 2006) as well as how to present information to the user (Rieser and Lemon, 2009; Janarthanam and Lemon, 2009). In addition to covering different domains, SUs should also be able to model relevant user attributes (Schatzmann et al., 2006), such as cooperativeness vs. non-cooperativeness (López-Cózar et al., 2006; Jung et al., 2009), or age (Georgila et al., 2008). In this paper, we focus on user age.

As the proportion of older people in the population increases, it becomes essential to make spoken dialogue systems (SDS) easy to use for this group of people. Only very few spoken dialogue systems have been developed for older people (e.g. Nursebot (Roy et al., 2000)), and we are aware of no work on learning specific dialogue policies for older people using SUs and RL.

Older people present special challenges for dialogue systems. While cognitive and perceptual abilities generally decline with age, the spread of ability in older people is far larger than in any other segment of the population (Rabbitt and Anderson, 2005). Older users may also use different strategies for interacting with SDS. In our previous work on studying the interactions between older and younger users and a simulated appointment scheduling SDS (Wolters et al., 2009b), we found that some older users were very "social", treating the system like a human, and failing to adapt to the SDS's system-initiative dialogue strategy. A third of the older users, however, tended to be more "factual", using short commands and conforming to the system's dialogue strategy. In that, they were very similar to the younger users (Wolters et al., 2009b).

In previous work (Georgila et al., 2008), we successfully built SUs for both older and younger

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 103–106,
The University of Tokyo, September 24-25, 2010.

adults from the corpus used by (Wolters et al., 2009b) and documented in (Georgila et al., 2010). When we evaluated the SUs using metrics such as precision and recall (Georgila et al., 2006; Schatzmann et al., 2006), we found that SUs trained on older users' data can cover behaviour patterns typical of younger users, but not the opposite. The behaviour of older people is too diverse to be captured by a SU trained on younger users' data. This result agrees with the findings of (Wolters et al., 2009b; Georgila et al., 2010).

In this study, we take our work one step further—we use the SUs developed in (Georgila et al., 2008) to learn dialogue policies and evaluate the resulting policies with data from both older and younger users. Our work is important for two reasons. First, to the best of our knowledge this is the first time that people have used SUs and RL to learn dialogue strategies for the increasingly important population of older users. Second, despite the fact that SUs are used for learning dialogue strategies it is not clear whether they can learn policies that are appropriate for different user populations. We show that SUs can be successfully used to learn policies for older users that are adapted to their specific patterns of behaviour, even though these patterns are far more varied than the behaviour patterns of younger users. This provides evidence for the validity of the user simulation methodology for learning and evaluating dialogue strategies for different user populations.

The structure of the paper is as follows: In section 2 we describe our data set, discuss the differences between older and younger users as seen in our corpus, and describe our user simulations. In section 3, we present the results of our experiments. Finally, in section 4 we present our conclusions and propose future work.

2 The Corpus

In the original dialogue corpus, people were asked to schedule health care appointments with 9 different simulated SDS in a Wizard-of-Oz setting. The systems varied in the number of options presented at each stage of the dialogue (1, 2, 4), and in the confirmation strategies used (explicit confirmation, implicit confirmation, no confirmation). System utterances were generated using a simple template-based algorithm and synthesised using a female Scottish English unit selection voice. The human Wizard took over the function of speech recognition (ASR), language understanding (NLU), and dialogue management components. No ASR or NLU errors were simulated, because having to deal with ASR and/or NLU errors in addition to task completion would have increased cognitive load (Wolters et al., 2009a).

The system (Wizard) followed a strict policy which resulted in dialogues with a fixed schema: First, users arranged to see a specific health care professional, then they arranged a specific half-day, and finally, a specific half-hour time slot on that half-day was agreed. Users were not allowed to skip any stage of the dialogue. This design ensured that all users were presented with the relevant number of options and the relevant confirmation strategy at least three times per dialogue. In a final step, the Wizard confirmed the appointment.

The full corpus consists of 447 dialogues; 3 dialogues were not recorded. A total of 50 participants were recruited, of which 26 were older, aged between 50 and 85 years, and 24 were younger, aged between 18 and 30 years. The older users contributed 232 dialogues, the younger ones 215. Older and younger users were matched for level of education and gender. All dialogues were transcribed orthographically and annotated with dialogue acts and dialogue context information. Using a unique mapping, we associate each dialogue act with a $\langle$speech act, task$\rangle$ pair, where the speech act is task independent and the task corresponds to the slot in focus (health professional, half-day or time slot). For example, $\langle$confirm_pos, hp$\rangle$ corresponds to positive explicit confirmation of the health professional slot. For each dialogue, detailed measures of dialogue quality were recorded: objective task completion, perceived task completion, appointment recall, length (in turns), and extensive user satisfaction ratings. For a detailed discussion of the corpus, see (Georgila et al., 2010).

The choice of dialogue strategy did not affect task completion and appointment recall, but had significant effects on efficiency (Wolters et al., 2009a). Task completion and appointment recall were the same for older and younger users, but older users took more turns to complete the task (Wolters et al., 2009a). Clear differences between the two user groups emerge when we look at interaction patterns in more detail (Wolters et al., 2009b; Georgila et al., 2010). Older people tend to "ground" information (using repetitions) and take the initiative more than younger people. In our corpus it was very common that the older person would provide information about the half-day and the time slot of the appointment before having been asked by the system. However, due to the

	Experiment 1	Experiment 2
slot filled	+50	+50
appointment confirmed	+200	+200
dialogue length	-5 per turn	-5 per turn
slot confirmed	+100	not used
wrong order	-500	not used

Table 1: Reward functions for the experiments.

strict policy of the Wizard, this information would be ignored and the system would later ask for the information that had already been provided.

In our SUs, each user utterance corresponds to a user action described by a list of ⟨speech act, task⟩ pairs. There are 31 distinct system actions and 389 distinct actions for older users. Younger people used a subset of 125 of the older users' actions. Our SUs do not simulate ASR or NLU errors since such errors were not simulated in the collection of the corpus.

We built n-grams of system and user actions with n varying from 2 to 5. Given a history of n-1 actions from system and user, the SU generates an action based on a probability distribution learned from the training data (Georgila et al., 2006). In the present study, n was set to 3, which means that each user action is predicted based on the previous user action and the previous system action.

3 Learning Dialogue Strategies

We performed two experiments. In Experiment 1, our goal was to learn the policy of the Wizard, i.e. the strict system-initiative policy of requesting and confirming information for each slot before moving to the next slot, in the following order: health professional, half-day, time slot. In Experiment 2, our goal was to learn a more flexible policy that could accommodate some degree of user initiative.

The reward functions for both experiments are specified in Table 1; they are similar to the reward functions used in the literature, e.g. (Frampton and Lemon, 2006). Slots that have been filled successfully and confirmed appointments are rewarded, while long dialogues are penalised. For Experiment 1, policies were rewarded that filled slots in the correct order and that confirmed each slot after it had been filled. A large penalty was imposed when the policy deviated from the strict slot order (health professional, half-day, time slot). For Experiment 2, these constraints were removed. Slots could be filled in any order. Confirmations were not required because there was no speech act in the corpus for confirming more than one slot at a time.

In both experiments we used the SARSA-λ algorithm (Sutton and Barto, 1998) for RL. 30,000 iterations were used for learning the final policy for each condition. For each experiment, we learned two policies, Policy-Old, which was based on simulated older users, and Policy-Young, which was based on simulated younger users. The resulting policies were then tested on simulated older users (Test-Old) and simulated younger users (Test-Young). To have comparable results between Experiment 1 and Experiment 2, during testing we score our policies using the reward function of Experiment 2. The best possible score is 190, i.e. the user fills all the slots in one turn and then confirms the appointment. (Note that +50 points are given when a slot is only filled, not confirmed too.) For each test condition, we generated 10,000 simulated dialogues. Overall scores for each combination of policy and SU were established using 5-fold cross-validation.

Our results are summarised in Figure 1. While average rewards were not affected by policy type (ANOVA, $F(1, 68)$=1, p=0.3) or training data set ($F(1, 185)$=3, p=0.09), we found a very strong interaction between policy type and data set ($F(1, 3098)$=51, p=0.000). Learning with simulated younger users yields better strict policies than learning with older users (Tukey's Honest Significant Difference Test, Δ=20, 95% CI = $[11, 30]$, p=0.000), while learning with simulated older users yields better flexible policies than learning with younger users (Δ=15, 95% CI = $[6, 24]$, p=0.001). This is what we would expect from our corpus analysis, since the interaction behaviour of older users is far more variable than that of younger users (Wolters et al., 2009b; Georgila et al., 2010).

The strict policy that was learned from simulated younger users was as follows, with only slight variations: first request the type of health professional, then implicitly confirm the health professional and request the half-day slot, then implicitly confirm the half-day slot and request the time slot, and then confirm the appointment. The strict policy learned from simulated older users was similar, but less successful, because most older users do not readily conform to the fixed structure.

The flexible policy learned from simulated older users takes into account initiative from the user and does not always confirm. The score for the flexible policy learned from simulated younger users was relatively low, even though the resulting

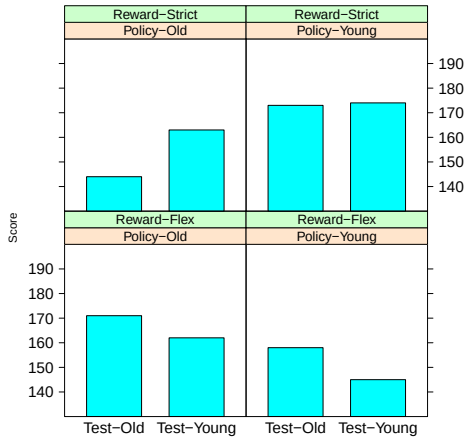


Figure 1: Mean scores for each combination of reward function, training set, and test set (5-fold cross-validation).

policy was very similar to the strict policy learned from younger users (i.e. a sequence of information requests and implicit confirmations), and even though the behaviour of younger users is far more predictable than the behaviour of older users. It appears that the explicit penalty for violating the order of slots is crucial for fully exploiting the patterns in younger users' behaviour.

4 Conclusions

We have shown that SUs can be used to learn appropriate policies for older adults, even though their interaction behaviour is more complex and diverse than that of younger adults. Crucially, simulated older users allowed us to learn a more flexible version of the strict system-initiative dialogue strategies that were used for creating the original corpus of interactions. These results are consistent with previous analyses of the original corpus (Wolters et al., 2009b; Georgila et al., 2010) and support the validity of the user simulation methodology for learning and evaluating dialogue strategies.

In our future work, we will experiment with more complex SUs, e.g. linear feature combination models (Georgila et al., 2006), and see if they can be used to learn similar policies. We also plan to study the effect of training and testing with different user simulation techniques, such as n-grams versus linear feature combination models.

Acknowledgements

This research was partially supported by the MATCH project (SHEFC-HR04016, `http://www.match-project.org.uk`). Georgila is supported by the U.S. Army Research, Development, and Engineering Command (RDECOM). The content does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.

References

M. Frampton and O. Lemon. 2006. Learning more effective dialogue strategies using limited dialogue move features. In *Proc. ACL*.

K. Georgila, J. Henderson, and O. Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. Interspeech*.

K. Georgila, M. Wolters, and J. Moore. 2008. Simulating the behaviour of older versus younger users. In *Proc. ACL*.

K. Georgila, Maria Wolters, J.D. Moore, and R.H. Logie. 2010. The MATCH corpus: A corpus of older and younger users' interactions with spoken dialogue systems. *Language Resources and Evaluation*, 44(3):221–261.

S. Janarthanam and O. Lemon. 2009. A two-tier user simulation model for reinforcement learning of adaptive referring expression generation policies. In *Proc. SIGdial*.

S. Jung, C. Lee, K. Kim, and G.G. Lee. 2009. Hybrid approach to user intention modeling for dialog simulation. In *Proc. ACL*.

R. López-Cózar, Z. Callejas, and M. McTear. 2006. Testing the performance of spoken dialogue systems by means of an artificially simulated user. *Artificial Intelligence Review*, 26(4):291–323.

P. Rabbitt and M.M. Anderson. 2005. The lacunae of loss? Aging and the differentiation of human abilities. In F.I. Craik and E. Bialystok, editors, *Lifespan Cognition: Mechanisms of Change*, chapter 23. Oxford University Press, New York, NY.

V. Rieser and O. Lemon. 2009. Natural language generation as planning under uncertainty for spoken dialogue systems. In *Proc. EACL*.

N. Roy, J. Pineau, and S. Thrun. 2000. Spoken dialog management for robots. In *Proc. ACL*.

J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowlege Engineering Review*, 21(2):97–126.

R.S. Sutton and A.G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

J. Williams and S. Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

M. Wolters, K. Georgila, J.D. Moore, R.H. Logie, S.E. MacPherson, and M. Watson. 2009a. Reducing working memory load in spoken dialogue systems. *Interacting with Computers*, 21(4):276–287.

M. Wolters, K. Georgila, J.D. Moore, and S.E. MacPherson. 2009b. Being old doesn't mean acting old: How older users interact with spoken dialog systems. *ACM Trans. Accessible Computing*, 2(1).

Sparse Approximate Dynamic Programming for Dialog Management

Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin
SUPELEC - IMS Research Group, Metz - France.
{senthilkumar.chandramohan, matthieu.geist, olivier.pietquin}@supelec.fr

Abstract

Spoken dialogue management strategy optimization by means of Reinforcement Learning (RL) is now part of the state of the art. Yet, there is still a clear mismatch between the complexity implied by the required naturalness of dialogue systems and the inability of standard RL algorithms to scale up. Another issue is the sparsity of the data available for training in the dialogue domain which can not ensure convergence of most of RL algorithms. In this paper, we propose to combine a sample-efficient generalization framework for RL with a feature selection algorithm for the learning of an optimal spoken dialogue management strategy.

1 Introduction

Optimization of dialogue management strategies by means of Reinforcement Learning (RL) (Sutton and Barto, 1998) is now part of the state of the art in the research area of Spoken Dialogue Systems (SDS) (Levin and Pieraccini, 1998; Singh et al., 1999; Pietquin and Dutoit, 2006; Williams and Young, 2007). It consists in casting the dialogue management problem into the Markov Decision Processes (MDP) paradigm (Bellman, 1957) and solving the associated optimization problem. Yet, there is still a clear mismatch between the complexity implied by the required naturalness of the dialogue systems and the inability of standard RL algorithms to scale up. Another issue is the sparsity of the data available for training in the dialogue domain because collecting and annotating data is very time consuming. Yet, RL algorithms are very data demanding and low amounts of data can not ensure convergence of most of RL algorithms. This latter problem has been extensively studied in the recent years and is addressed by simulating new dialogues thanks to a statistical model of human-machine interaction (Pietquin, 2005) and user modeling (Eckert et al., 1997; Pietquin and Dutoit, 2006; Schatzmann et al., 2006). However, this results in a variability of the learned strategy depending on the user modeling method (Schatzmann et al., 2005) and no common agreement exists on the best user model.

The former problem, that is dealing with complex dialogue systems within the RL framework, has received much less attention. Although some works can be found in the SDS literature it is far from taking advantage of the large amount of machine learning literature devoted to this problem. In (Williams and Young, 2005), the authors reduce the complexity of the problem (which is actually a Partially Observable MDP) by automatically condensing the continuous state space in a so-called *summary space*. This results in a clustering of the state space in a discrete set of states on which standard RL algorithms are applied. In (Henderson et al., 2008), the authors use a linear approximation scheme and apply the SARSA(λ) algorithm (Sutton and Barto, 1998) in a batch setting (from data and not from interactions or simulations). This algorithm was actually designed for online learning and is known to converge very slowly. It therefore requires a lot of data and especially in large state spaces. Moreover, the choice of the features used for the linear approximation is particularly simple since features are the state variables themselves. The approximated function can therefore not be more complex than an hyper-plane in the state variables space. This drawback is shared by the approach of (Li et al., 2009) where a batch algorithm (Least Square Policy Iteration or LSPI) is combined to a pruning method to only keep the most meaningful features. In addition the complexity of LSPI is $O(p^3)$.

In the machine learning community, this issue is actually addressed by function approximation accompanied with dimensionality reduction. The

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 107–115,
The University of Tokyo, September 24-25, 2010.

data sparsity problem is also widely addressed in this literature, and sample-efficiency is one main trend of research in this field. In this paper, we propose to combine a sample-efficient batch RL algorithm (namely the Fitted Value Iteration (FVI) algorithm) with a feature selection method in a novel manner and to apply this original combination to the learning of an optimal spoken dialogue strategy. Although the algorithm uses a linear combination of features (or basis functions), these features are much richer in their ability of representing complex functions.

The ultimate goal of this research is to provide a way of learning optimal dialogue policies for a large set of situations from a small and fixed set of annotated data in a tractable way.

The rest of this paper is structured as follows. Section 2 gives a formal insight of MDP and briefly reminds the casting of the dialogue problem into the MDP framework. Section 3.2 provides a description of approximate Dynamic Programming along with LSPI and FVI algorithms. Section 4 provides an overview on how LSPI and FVI can be combined with a feature selection scheme (which is employed to learn the representation of the Q-function from the dialogue corpus). Our experimental set-up, results and a comparison with state-of-the-art methods are presented in Section 5. Eventually, Section 6 concludes.

2 Markov Decision Processes

The MDP (Puterman, 1994) framework is used to describe and solve sequential decision making problems or equivalently optimal control problems in the case of stochastic dynamic systems. An MDP is formally a tuple $\{S, A, P, R, \gamma\}$ where S is the (finite) state space, A the (finite) action space, $P \in \mathcal{P}(S)^{S\times A}$ the family of Markovian transition probabilities[1], $R \in \mathbb{R}^{S\times A\times S}$ the reward function and γ the discounting factor ($0 \leq \gamma \leq 1$). According to this formalism, a system to be controlled steps from state to state ($s \in S$) according to transition probabilities P as a consequence of the controller's actions ($a \in A$). After each transition, the system generates an immediate reward (r) according to its reward function R. How the system is controlled is modeled with a so-called *policy* $\pi \in A^S$ mapping states to actions. The quality of a policy is quantified by the so-called value function which maps each state to the expected discounted cumulative reward given that the agent starts in this state and follows the policy π: $V^\pi(s) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, \pi]$. An optimal policy π^* maximizes this function for each state: $\pi^* = \text{argmax}_\pi V^\pi$. Suppose that we are given the optimal value function V^* (that is the value function associated to an optimal policy), deriving the associated policy would require to know the transition probabilities P. Yet, this is usually unknown. This is why the state-action value (or Q-) function is introduced. It adds a degree of freedom on the choice of the first action:

$$Q^\pi(s,a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a, \pi] \quad (1)$$

The optimal policy is noted π^* and the related Q-function $Q^*(s,a)$. An action-selection strategy that is greedy according to this function ($\pi(s) = \text{argmax}_a Q^*(s,a)$) provides an optimal policy.

2.1 Dialogue as an MDP

The casting of the spoken dialogue management problem into the MDP framework (MDP-SDS) comes from the equivalence of this problem to a sequential decision making problem. Indeed, the role of the dialogue manager (or the decision maker) is to select and perform dialogue acts (actions in the MDP paradigm) when it reaches a given dialogue turn (state in the MDP paradigm) while interacting with a human user. There can be several types of system dialogue acts. For example, in the case of a restaurant information system, possible acts are *request(cuisine_type)*, *provide(address)*, *confirm(price_range)*, *close etc.* The dialogue state is usually represented efficiently by the Information State paradigm (Larsson and Traum, 2000). In this paradigm, the dialogue state contains a compact representation of the history of the dialogue in terms of system acts and its subsequent user responses (user acts). It summarizes the information exchanged between the user and the system until the considered state is reached.

A dialogue management strategy is thus a mapping between dialogue states and dialogue acts. Still following the MDP's definitions, the optimal strategy is the one that maximizes some cumulative function of rewards collected all along the interaction. A common choice for the immediate reward is the contribution of each action to user satisfaction (Singh et al., 1999). This subjective

[1]Notation $f \in A^B$ is equivalent to $f : B \to A$

reward is usually approximated by a linear combination of objective measures like dialogue duration, number of ASR errors, task completion *etc*. (Walker et al., 1997).

3 Solving MDPs

3.1 Dynamic Programming

Dynamic programming (DP) (Bellman, 1957) aims at computing the optimal policy π^* if the transition probabilities and the reward function are known.

First, the *policy iteration* algorithm computes the optimal policy in an iterative way. The initial policy is arbitrary set to π_0. At iteration k, the policy π_{k-1} is evaluated, that is the associated Q-function $Q^{\pi_{k-1}}(s,a)$ is computed. To do so, the Markovian property of the transition probabilities is used to rewrite Equation (1) as :

$$\begin{aligned} Q^{\pi}(s,a) &= E_{s'|s,a}[R(s,a,s') + \gamma Q^{\pi}(s',\pi(s'))] \\ &= T^{\pi}Q^{\pi}(s,a) \end{aligned} \quad (2)$$

This is the so-called Bellman evaluation equation and T^{π} is the Bellman evaluation operator. T^{π} is linear and therefore this defines a linear system that can be solved by standard methods or by an iterative method using the fact that Q^{π} is the unique fixed-point of the Bellman evaluation operator (T^{π} being a contraction): $\hat{Q}^{\pi}_i = T^{\pi}\hat{Q}^{\pi}_{i-1}, \quad \forall \hat{Q}^{\pi}_0 \ \lim_{i\to\infty} \hat{Q}^{\pi}_i = Q^{\pi}$. Then the policy is improved, that is π_k is greedy respectively to $Q^{\pi_{k-1}}$: $\pi_k(s) = \operatorname{argmax}_{a\in A} Q^{\pi_{k-1}}(s,a)$. Evaluation and improvement steps are iterated until convergence of π_k to π^* (which can be demonstrated to happen in a finite number of iterations when $\pi_k = \pi_{k-1}$).

The *value iteration* algorithm aims at estimating directly the optimal state-action value function Q^* which is the solution of the Bellman optimality equation (or equivalently the unique fixed-point of the Bellman optimality operator T^*):

$$\begin{aligned} Q^*(s,a) &= E_{s'|s,a}[R(s,a,s') + \gamma \max_{b\in A} Q^*(s',b)] \\ &= T^*Q^*(s,a) \end{aligned} \quad (3)$$

The T^* operator is not linear, therefore computing Q^* via standard system-solving methods is not possible. However, it can be shown that T^* is also a contraction (Puterman, 1994). Therefore, according to Banach fixed-point theorem, Q^* can be estimated using the following iterative way:

$$\hat{Q}^*_i = T^*\hat{Q}^*_{i-1}, \quad \forall \hat{Q}^*_0 \ \lim_{i\to\infty} \hat{Q}^*_i = Q^* \quad (4)$$

However, the convergence takes an infinite number of iterations. Practically speaking, iterations are stopped when some criterion is met, classically a small difference between two iterations: $\|\hat{Q}^*_i - \hat{Q}^*_{i-1}\| < \xi$. The estimated optimal policy (which is what we are ultimately interested in) is greedy respectively to the estimated optimal Q-function: $\hat{\pi}^*(s) = \operatorname{argmax}_{a\in A} \hat{Q}^*(s,a)$.

3.2 Approximate Dynamic Programming

DP-based approaches have two drawbacks. First, they assume the transition probabilities and the reward function to be known. Practically, it is rarely true and especially in the case of spoken dialogue systems. Most often, only examples of dialogues are available which are actually trajectories in the state-action space. Second, it assumes that the Q-function can be exactly represented. However, in real world dialogue management problems, state and action spaces are often too large (even continuous) for such an assumption to hold. Approximate Dynamic Programming (ADP) aims at estimating the optimal policy from trajectories when the state space is too large for a tabular representation. It assumes that the Q-function can be approximated by some parameterized function $\hat{Q}_\theta(s,a)$. In this paper, a linear approximation of the Q-function will be assumed: $\hat{Q}_\theta(s,a) = \theta^T\phi(s,a)$. where $\theta \in \mathbb{R}^p$ is the parameter vector and $\phi(s,a)$ is the set of p basis functions. All functions expressed in this way define a so-called *hypothesis space* $\mathcal{H} = \{\hat{Q}_\theta | \theta \in \mathbb{R}^p\}$. Any function Q can be *projected* onto this hypothesis space by the operator Π defined as

$$\Pi Q = \operatorname*{argmin}_{\hat{Q}_\theta \in \mathcal{H}} \|Q - \hat{Q}_\theta\|^2. \quad (5)$$

The goal of the ADP algorithms explained in the subsequent sections is to compute the best set of parameters θ given the basis functions.

3.2.1 Least-Squares Policy Iteration

The least-squares policy iteration (LSPI) algorithm has been introduced by Lagoudakis and Parr (2003). The underlying idea is exactly the same as for policy iteration: interleaving evaluation and improvement steps. The improvement steps are same as before, but the evaluation step should learn an approximate representation of the Q-function using samples. In LSPI, this is done using the Least-Squares Temporal Differences (LSTD) algorithm of Bradtke and Barto (1996).

LSTD aims at minimizing the distance between the approximated Q-function $\hat{Q}_\theta$ and the projection onto the hypothesis space of its image through the Bellman evaluation operator $\Pi T^\pi \hat{Q}_\theta$: $\theta_\pi = \text{argmin}_{\theta \in \mathbb{R}^p} \|\hat{Q}_\theta - \Pi T^\pi \hat{Q}_\theta\|^2$. This can be interpreted as trying to minimize the difference between the two sides of the Bellman equation (1) (which should ideally be zero) in the hypothesis space. Because of the approximation, this difference is most likely to be non-zero.

Practically, T^π is not known, but a set of N transitions $\{(s_j, a_j, r_j, s'_j)_{1 \leq j \leq N}\}$ is available. LSTD therefore solves the following optimization problem: $\theta_\pi = \text{argmin}_\theta \sum_{j=1}^N C_j^N(\theta)$ where $C_j^N(\theta) = (r_j + \gamma \hat{Q}_{\theta_\pi}(s'_j, \pi(s'_j)) - \gamma \hat{Q}_\theta(s_j, a_j))^2$. Notice that θ_π appears in both sides of the equation, which renders this problem difficult to solve. However, thanks to the linear parametrization, it admits an analytical solution, which defines the LSTD algorithm:

$$\theta_\pi = (\sum_{j=1}^N \phi_j \Delta\phi_j^\pi)^{-1} \sum_{j=1}^N \phi_j r_j \qquad (6)$$

with $\phi_j = \phi(s_j, a_j)$ and $\Delta\phi_j^\pi = \phi(s_j, a_j) - \gamma\phi(s'_j, \pi(s'_j))$.

LSPI is initialized with a policy π_0. Then, at iteration k, the Q-function of policy π_{k-1} is estimated using LSTD, and π_k is greedy respectively to this estimated state-action value function. Iterations are stopped when some stopping criterion is met (*e.g.*, small differences between consecutive policies or associated Q-functions).

3.2.2 Least-Squares Fitted Value Iteration

The Fitted Value Iteration (FVI) class of algorithms (Bellman and Dreyfus, 1959; Gordon, 1995; Ernst et al., 2005) generalizes value iteration to model-free and large state space problems. The T^* operator (eq. (3)) being a contraction, a straightforward idea would be to apply it iteratively to the approximation similarly to eq. (4): $\hat{Q}_{\theta_k} = T^* \hat{Q}_{\theta_{k-1}}$. However, $T^* \hat{Q}_\theta$ does not necessarily lie in $\mathcal{H}$, it should thus be projected again onto the hypothesis space $\mathcal{H}$. By considering the same projection operator Π as before, this leads to finding the parameter vector θ satisfying: $\hat{Q}_\theta^* = \Pi T^* \hat{Q}_\theta^*$. The fitted-$Q$ algorithm (a special case of FVI) assumes that the composed ΠT^* operator is a contraction and therefore admits an unique fixed point, which is searched for through the classic iterative scheme: $\hat{Q}_{\theta_k} = \Pi T^* \hat{Q}_{\theta_{k-1}}$. However, the model (transition probabilities and the reward function) is usually not known, therefore a *sampled* Bellman optimality operator $\hat{T}^*$ is considered instead. For a transition sample (s_j, a_j, r_j, s'_j), it is defined as: $\hat{T}^* Q(s_j, a_j) = r_j + \gamma \max_{a \in A} Q(s'_j, a)$. This defines the general fitted-Q algorithm (θ_0 being chosen by the user): $\hat{Q}_{\theta_k} = \Pi \hat{T}^* \hat{Q}_{\theta_{k-1}}$. Fitted-$Q$ can then be specialized by choosing how $\hat{T}^* \hat{Q}_{\theta_{k-1}}$ is projected onto the hypothesis space, that is the supervised learning algorithm that solves the projection problem of eq. (5). The least squares algorithm is chosen here.

The parametrization being linear, and a training base $\{(s_j, a_j, r_j, s'_j)_{1 \leq j \leq N}\}$ being available, the least-squares fitted-Q (LSFQ for short) is derived as follows (we note $\phi(s_j, a_j) = \phi_j$):

$$\theta_k = \underset{\theta \in \mathbb{R}^p}{\text{argmin}} \sum_{j=1}^N (\hat{T}^* \hat{Q}_{\theta_{k-1}}(s_j, a_j) - \hat{Q}_\theta(s_j, a_j))^2 \qquad (7)$$
$$= (\sum_{j=1}^N \phi_j \phi_j^T)^{-1} \sum_{j=1}^N \phi_j (r_j + \gamma \max_{a \in A} (\theta_{k-1}^T \phi(s'_j, a)))$$

Equation (7) defines an iteration of the proposed linear least-squares-based fitted-Q algorithm. An initial parameter vector θ_0 should be chosen, and iterations are stopped when some criterion is met (maximum number of iterations or small difference between two consecutive parameter vector estimates). Assuming that there are M iterations, the optimal policy is estimated as $\hat{\pi}^*(s) = \text{argmax}_{a \in A} \hat{Q}_{\theta_M}(s, a)$.

4 Learning a sparse parametrization

LSPI and LSFQ (FVI) assume that the basis functions are chosen beforehand. However, this is difficult and problem-dependent. Thus, we propose to combine these algorithms with a scheme which *learns* the representation from dialogue corpora.

Let's place ourselves in a general context. We want to learn a parametric representation for an approximated function $f_\theta(z) = \theta^T \phi(z)$ from samples $\{z_1, \ldots, z_N\}$. A classical choice is to choose a kernel-based representation (Scholkopf and Smola, 2001). Formally, a kernel $K(z, \tilde{z}_i)$ is a continuous, positive and semi-definite function (*e.g.*, Gaussian or polynomial kernels) centered on $\tilde{z}_i$. The feature vector $\phi(z)$ is therefore of the form: $\phi(z) = \begin{pmatrix} K(z, \tilde{z}_1) & \ldots & K(z, \tilde{z}_p) \end{pmatrix}$. The question this section answers is the following: given the training basis $\{z_1, \ldots, z_N\}$ and a kernel

K, how to choose the number p of basis functions and the associated kernel centers $(\tilde{z}_1, \ldots, \tilde{z}_p)$?

An important result about kernels is the Mercer theorem, which states that for each kernel K there exists a mapping $\varphi : z \in Z \rightarrow \varphi(z) \in \mathcal{F}$ such that $\forall z_1, z_2 \in Z$, $K(z_1, z_2) = \langle \varphi(z_1), \varphi(z_2) \rangle$ (in short, K defines a dot product in $\mathcal{F}$). The space $\mathcal{F}$ is called the feature space, and it can be of infinite dimension (*e.g.*, Gaussian kernel), therefore φ cannot always be explicitly built. Given this result and from the bilinearity of the dot product, f_θ can be rewritten as follows: $f_\theta(z) = \sum_{i=1}^{p} \theta_i K(z, \tilde{z}_i) = \langle \varphi(z), \sum_{i=1}^{p} \theta_i \varphi(\tilde{z}_i) \rangle$. Therefore, a kernel-based parametrization corresponds to a linear approximation in the feature space, the weight vector being $\sum_{i=1}^{p} \theta_i \varphi(\tilde{z}_i)$. This is called the *kernel trick*. Consequently, kernel centers $(\tilde{z}_1, \ldots, \tilde{z}_p)$ should be chosen such that $(\varphi(\tilde{z}_1), \ldots, \varphi(\tilde{z}_p))$ are linearly independent in order to avoid using redundant basis functions. Moreover, kernel centers should be chosen among the training samples. To sum up, learning such a parametrization reduces to finding a dictionary $\mathcal{D} = (\tilde{z}_1, \ldots, \tilde{z}_p) \in \{z_1, \ldots, z_N\}$ such that $(\varphi(\tilde{z}_1), \ldots, \varphi(\tilde{z}_p))$ are linearly independent and such that they span the same subspace as $(\varphi(z_1), \ldots, \varphi(z_N))$. Engel et al. (2004) provides a dictionary method to solve this problem, briefly sketched here.

The training base is sequentially processed, and the dictionary is initiated with the first sample: $\mathcal{D}_1 = \{z_1\}$. At iteration k, a dictionary $\mathcal{D}_{k-1}$ computed from $\{z_1, \ldots, z_{k-1}\}$ is available and the k^{th} sample z_k is considered. If $\varphi(z_k)$ is linearly independent of $\varphi(\mathcal{D}_{k-1})$, then it is added to the dictionary: $\mathcal{D}_k = \mathcal{D}_{k-1} \cup \{z_k\}$. Otherwise, the dictionary remains unchanged: $\mathcal{D}_k = \mathcal{D}_{k-1}$. Linear dependency can be checked by solving the following optimization problem (p_{k-1} being the size of $\mathcal{D}_{k-1}$): $\delta = \text{argmin}_{w \in \mathbb{R}^{p_{k-1}}} \|\varphi(z_k) - \sum_{i=1}^{p_{k-1}} w_i \varphi(\tilde{z}_i)\|^2$. Thanks to the kernel trick (that is the fact that $\langle \varphi(z_k), \varphi(\tilde{z}_i) \rangle = K(z_k, \tilde{z}_i)$) and to the bilinearity of the dot product, this optimization problem can be solved analytically and without computing explicitly φ. Formally, linear dependency is satisfied if $\delta = 0$. However, an approximate linear dependency is allowed, and $\varphi(z_k)$ will be considered as linearly dependent of $\varphi(\mathcal{D}_{k-1})$ if $\delta < \nu$, where ν is the so-called *sparsification factor*. This allows controlling the trade-off between quality of the representation and its sparsity. See Engel et al. (2004) for details as well as an efficient implementation of this dictionary approach.

4.1 Resulting algorithms

We propose to combine LSPI and LSFQ with the sparsification approach exposed in the previous section: a kernel is chosen, the dictionary is computed and then LSPI or LSFQ is applied using the learnt basis functions. For LSPI, this scheme has been proposed before by Xu et al. (2007) (with the difference that they generate new trajectories at each iteration whereas we use the same for all iterations). The proposed sparse LSFQ algorithm is a novel contribution of this paper.

We start with the sparse LSFQ algorithm. In order to train the dictionary, the inputs are needed (state-action couples in this case), but not the outputs (reward are not used). For LSFQ, the input space remains the same over iterations, therefore the dictionary can be computed in a preprocessing step from $\{(s_j, a_j)_{1 \leq j \leq N}\}$. Notice that the matrix $(\sum_{j=1}^{N} \phi_j \phi_j^T)^{-1}$ remains also the same over iterations, therefore it can be computed in a preprocessing step too. The proposed sparse LSFQ algorithm is summarized in appendix Algorithm 1.

For the sparse LSPI algorithm, things are different. This time, the inputs depend on the iteration. More precisely, at iteration k, the input is composed of state-action couples (s_j, a_j) but also of transiting state-action couples $(s'_j, \pi_{k-1}(s'_j))$. Therefore the dictionary has to be computed at each iteration from $\{(s_j, a_j)_{1 \leq j \leq N}, (s'_j, \pi_{k-1}(s'_j))_{1 \leq j \leq N}\}$. This defines the parametrization which is considered for the Q-function evaluation. The rest of the algorithm is as for the classic LSPI and it is summarized in appendix Algorithm 2.

Notice that sparse LSFQ has a lower computational complexity than the sparse LSPI. For sparse LSFQ, dictionary and the matrix P^{-1} are computed in a preprocessing step, therefore the complexity per iteration is in $O(p^2)$, with p being the number of basis functions computed using the dictionary method. For LSPI, the inverse matrix depends on the iteration, as well as the dictionary, therefore the computational complexity is in $O(p_k^3)$ per iteration, where p_k is the size of the dictionary computed at the k^{th} iteration.

5 Experimental set-up and results

5.1 Dialogue task and RL parameters

The experimental setup is a form-filling dialogue system in the tourist information domain similar to the one studied in (Lemon et al., 2006). The system aims to give information about restaurants in the city based on specific user preferences. Three slots are considered: (i) location, (ii) cuisine and (iii) price-range of the restaurant. The dialogue state has three continuous components ranging from 0 to 1, each representing the average of filling and confirmation confidence of the corresponding slots. The MDP SDS has 13 actions: Ask-slot (3 actions), Explicit-confirm (3 actions), Implicit-confirm and Ask-slot value (6 actions) and Close-dialogue (1 action). The γ parameter was set to 0.95 in order to encourage delayed rewards and also to induce an implicit penalty for the length of the dialogue episode. The reward function R is presented as follows: every correct slot filling is awarded 25, every incorrect slot filling is awarded -75 and every empty slot filling is awarded -300. The reward is awarded at the end of the dialogue.

5.2 Dialogue corpora for policy optimization

So as to perform sparse LSFQ or sparse LSPI, a dialogue corpus which represents the problem space is needed. As for any batch learning method, the samples used for learning should be chosen (if they can be chosen) to span across the problem space. In this experiment, a user simulation technique was used to generate the data corpora. This way, the sensibility of the method to the size of the training data-set could be analyzed (available human-dialogue corpora are limited in size). The user simulator was plugged to the DIPPER (Lemon et al., 2006) dialogue management system to generate dialogue samples. To generate data, the dialogue manager strategy was jointly based on a simple hand-coded policy (which aims only to fill all the slots before closing the dialogue episode irrespective of slot confidence score *i.e.,*) and random action selection.

Randomly selected system acts are used with probability ϵ and hand-coded policy selected system acts are used with probability (1-ϵ). During our data generation process the ϵ value was set to 0.9. Rather than using a fully random policy we used an ϵ-greedy policy to ensure that the problem space is well sampled and in the same time at least few episodes have successful completion of task compared to a totally random policy. We ran 56,485 episodes between the policy learner and an unigram user simulation, using the ϵ-greedy policy (of which 65% are successful task completion episodes) and collected 393,896 dialogue turns (state transitions). The maximum episode length is set as 100 dialogue turns. The dialogue turns (samples) are then divided into eight different training sets each with 5.10^4 samples.

5.3 Linear representation of Q-function

Two different linear representations of the Q-function were used. First, a set of basis functions computed using the dictionary method outlined in Section 4 is used. A Gaussian kernel is used for the dictionary computation (σ = 0.25). The number of elements present in the dictionary varied based on the number of samples used for computation and the sparsification factor. It was observed during the experiments that including a constant term to the Q-function representation (value set to 1) in addition to features selected by the dictionary method avoided weight divergence. Our second representation of Q-function used a set of hand-picked features presented as a set of Gaussian functions, centered in μ_i and with the same standard deviation $\sigma_i = \sigma$). Our RBF network had 3 Gaussians for each dimension in the state vector and considering that we have 13 actions, in total we used 351 (*i.e,* $3^3 \times 13$) features for approximating the Q-function. This allows considering that each state variable contributes to the value function differently according to its value contrarily to similar work (Li et al., 2009; Henderson et al., 2008) that considers linear contribution of each state variable. Gaussians were centered at μ_i = 0.0, 0.5, 1.0 in every dimension with a standard deviation $\sigma_i = \sigma = 0.25$. Our stopping criteria was based on comparison between L_1 norm of succeeding weights and a threshold ξ which was set to 10^{-2} *i.e,* convergence if $\sum_i \left(|\theta_i^n - \theta_i^{n-1}|^1\right) < \xi$, where n is the iteration number. For sparse LSPI since the dictionary is computed during each iteration, stopping criteria based on ξ is not feasible thus the learning was stopped after 30 iterations.

5.4 Evaluation of learned policy

We ran a set of learning iterations using two different representations of Q-function and with different numbers of training samples (one sample is a dialogue turn, that is a state transition $\{s, a, r, s'\}$). The number of samples used for training ranged

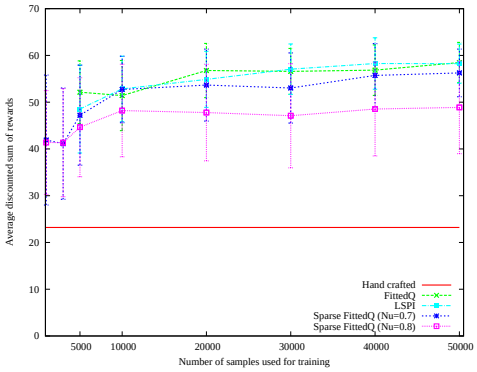


Figure 1: FittedQ policy evaluation statistics

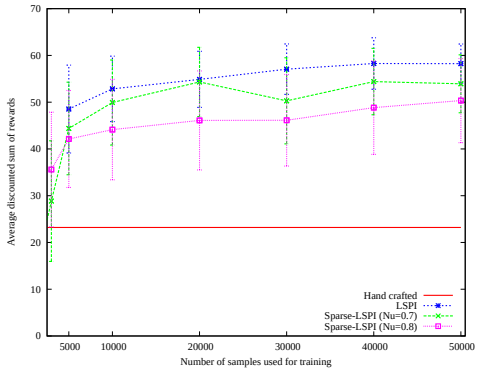


Figure 2: LSPI policy evaluation statistics

from 1.10^3 to 50.10^3 samples (no convergence of weights was observed with fewer samples than 1.10^3). The training is repeated for each of the 8 training data sets. Dictionary computed using different number of training samples and with ν=0.7 and 0.8 had a maximum of 367 and 306 elements respectively (with lower values of ν the number of features is higher than the hand-selected version). The policies learned were then tested using a unigram user simulation and the DIPPER dialogue management framework. Figures 1 and 2 show the average discounted sum of rewards of policies tested over 8×25 dialogue episodes.

5.5 Analysis of evaluation results

Our experimental results show that the dialogue policies learned using sparse SLFQ and LSPI with the two different Q-function representations perform significantly better than the hand-coded policy. Most importantly it can be observed from Figure 1 and 2 that the performance of sparse LSFQ and sparse LSPI (which uses the dictionary method for feature selection) are nearly as good as LSFQ and LSPI (which employs more numerous hand-selected basis functions). This shows the effectiveness of using the dictionary method for learning the representation of the Q-function from the dialogue corpora. For this specific problem the set of hand selected features seem to perform better than sparse LSPI and sparse LSFQ, but this may not be always the case. For complex dialogue management problems feature selection methods such as the one studied here will be handy since the option of manually selecting a good set of features will cease to exist.

Secondly it can be concluded that, similar to LSFQ and LSPI, the sparse LSFQ and sparse LSPI based dialogue management are also sample efficient and needs only few thousand samples (recall that a sample is a dialogue turn and not a dialogue episode) to learn fairly good policies, thus exhibiting a possibility to learn a good policy directly from very limited amount of dialogue examples. We believe this is a significant improvement when compared to the corpora requirement for dialogue management using other RL algorithms such as SARSA. However, sparse LSPI seems to result in poorer performance compared to sparse LSFQ.

One key advantage of using the dictionary method is that only mandatory basis functions are selected to be part of the dictionary. This results in fewer feature weights ensuring faster convergence during training. From Figure 1 it can also be observed that the performance of both LSFQ and LSPI (using hand selected features) are nearly identical. From a computational complexity point of view, LSFQ and LSPI roughly need the same number of iterations before the stopping criterion is met. However, reminding that the proposed LSFQ complexity is $O(p)^2$ per iteration whereas LSPI complexity is $O(p^3)$ per iteration, LSFQ is computationally less intensive.

6 Discussion and Conclusion

In this paper, we proposed two sample-efficient generalization techniques to learn optimal dialogue policies from limited amounts of dialogue examples (namely sparse LSFQ and LSPI). Particularly, a novel sparse LSFQ method has been proposed and was demonstrated to out-perform handcrafted and LSPI-based policies while using a limited number of features. By using a kernel-based approximation scheme, the power of representation of the state-action value function (or Q-function) is increased with comparison to state-of-

the-art algorithms (such as (Li et al., 2009; Henderson et al., 2008)). Yet the number of features is also increased. Using a sparsification algorithm, this number is reduced while policy performances are kept. In the future, more compact representation of the state-action value function will be investigated such as neural networks.

Acknowledgments

The work presented here is part of an ongoing research for CLASSiC project (Grant No. 216594, www.classic-project.org) funded by the European Commission's 7th Framework Programme (FP7).

References

Richard Bellman and Stuart Dreyfus. 1959. Functional approximation and dynamic programming. *Mathematical Tables and Other Aids to Computation*, 13:247–251.

Richard Bellman. 1957. *Dynamic Programming*. Dover Publications, sixth edition.

Steven J. Bradtke and Andrew G. Barto. 1996. Linear Least-Squares algorithms for temporal difference learning. *Machine Learning*, 22(1-3):33–57.

Wieland Eckert, Esther Levin, and Roberto Pieraccini. 1997. User Modeling for Spoken Dialogue System Evaluation. In *ASRU'97*, pages 80–87.

Yaakov Engel, Shie Mannor, and Ron Meir. 2004. The Kernel Recursive Least Squares Algorithm. *IEEE Transactions on Signal Processing*, 52:2275–2285.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. 2005. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*, 6:503–556.

Geoffrey Gordon. 1995. Stable Function Approximation in Dynamic Programming. In *ICML'95*.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets. *Computational Linguistics, vol. 34(4), pp 487-511*.

Michail G. Lagoudakis and Ronald Parr. 2003. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149.

Staffan Larsson and David R. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering, vol. 6, pp 323–340*.

Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew Stuttle. 2006. An ISU dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the TALK in-car system. In *EACL'06*, Morristown, NJ, USA.

Esther Levin and Roberto Pieraccini. 1998. Using markov decision process for learning dialogue strategies. In *ICASSP'98*.

Lihong Li, Suhrid Balakrishnan, and Jason Williams. 2009. Reinforcement Learning for Dialog Management using Least-Squares Policy Iteration and Fast Feature Selection. In *InterSpeech'09*, Brighton (UK).

Olivier Pietquin and Thierry Dutoit. 2006. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech & Language Processing, 14(2): 589-599*.

Olivier Pietquin. 2005. A probabilistic description of man-machine spoken communication. In *ICME'05*, pages 410–413, Amsterdam (The Netherlands), July.

Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley-Interscience, April.

Jost Schatzmann, Matthew N. Stuttle, Karl Weilhammer, and Steve Young. 2005. Effects of the user model on simulation-based learning of dialogue strategies. In *ASRU'05*, December.

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, vol. 21(2), pp. 97–126*.

Bernhard Scholkopf and Alexander J. Smola. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA.

Satinder Singh, Michael Kearns, Diane Litman, and Marilyn Walker. 1999. Reinforcement learning for spoken dialogue systems. In *NIPS'99*. Springer.

Richard S. Sutton and Andrew G. Barto. 1998. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, 3rd edition, March.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In *ACL'97*, pages 271–280, Madrid (Spain).

Jason Williams and Steve Young. 2005. Scaling up pomdps for dialogue management: the summary pomdp method. In *ASRU'05*.

Jason D. Williams and Steve Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language, vol. 21(2), pp. 393–422*.

Xin Xu, Dewen Hu, and Xicheng Lu. 2007. Kernel-based least squares policy iteration for reinforcement learning. *IEEE Transactions on Neural Networks*, 18(4):973–992, July.

Appendix

This appendix provides pseudo code for the algorithms described in the paper.

Algorithm 1: Sparse LSFQ.

Initialization;
Initialize vector θ_0, choose a kernel K and a sparsification factor ν;
Compute the dictionary;
$\mathcal{D} = \{(\tilde{s}_j, \tilde{a}_j)_{1 \leq j \leq p}\}$ from $\{(s_j, a_j)_{1 \leq j \leq N}\}$;
Define the parametrization;
$Q_\theta(s,a) = \theta^T \phi(s,a)$ with $\phi(s,a) = (K((s,a),(\tilde{s}_1,\tilde{a}_1)), \ldots, K((s,a),(\tilde{s}_p,\tilde{a}_p)))^T$;
Compute P^{-1};
$P^{-1} = (\sum_{j=1}^N \phi_j \phi_j^T)^{-1}$;
for $k = 1, 2, \ldots, M$ **do**
  Compute θ_k, see Eq. (7);
end
$\hat{\pi}_M^*(s) = \text{argmax}_{a \in A} \hat{Q}_{\theta_M}(s,a)$;

Algorithm 2: Sparse LSPI.

Initialization;
Initialize policy π_0, choose a kernel K and a sparsification factor ν;
for $k = 1, 2, \ldots$ **do**
  Compute the dictionary;
  $\mathcal{D} = \{(\tilde{s}_j, \tilde{a}_j)_{1 \leq j \leq p_k}\}$ from $\{(s_j, a_j)_{1 \leq j \leq N}, (s'_j, \pi_{k-1}(s'_j))_{1 \leq j \leq N}\}$;
  Define the parametrization;
  $Q_\theta(s,a) = \theta^T \phi(s,a)$ with $\phi(s,a) = (K((s,a),(\tilde{s}_1,\tilde{a}_1)), \ldots, K((s,a),(\tilde{s}_{p_k},\tilde{a}_{p_k})))^T$;
  Compute θ_{k-1}, see Eq. (6);
  Compute π_k;
  $\pi_k(s) = \text{argmax}_{a \in A} \hat{Q}_{\theta_{k-1}}(s,a)$;
end

Parameter estimation for agenda-based user simulation

Simon Keizer, Milica Gašić, Filip Jurčíček, François Mairesse,
Blaise Thomson, Kai Yu, and Steve Young *
University of Cambridge, Department of Engineering, Cambridge (UK)
{sk561,mg436,fj228,farm2,brmt2,ky219,sjy}@cam.ac.uk

Abstract

This paper presents an agenda-based user simulator which has been extended to be trainable on real data with the aim of more closely modelling the complex rational behaviour exhibited by real users. The trainable part is formed by a set of *random decision points* that may be encountered during the process of receiving a system act and responding with a user act. A sample-based method is presented for using real user data to estimate the parameters that control these decisions. Evaluation results are given both in terms of statistics of generated user behaviour and the quality of policies trained with different simulators. Compared to a handcrafted simulator, the trained system provides a much better fit to corpus data and evaluations suggest that this better fit should result in improved dialogue performance.

1 Introduction

In spoken dialogue systems research, modelling dialogue as a (Partially Observable) Markov Decision Process ((PO)MDP) and using reinforcement learning techniques for optimising dialogue policies has proven to be an effective method for developing robust systems (Singh et al., 2000; Levin et al., 2000). However, since this kind of optimisation requires a simulated user to generate a sufficiently large number of interactions to learn from, this effectiveness depends largely on the quality of such a user simulator. An important requirement for a simulator is for it to be realistic, i.e., it should generate behaviour that is similar to that of real users. Trained policies are then more likely to perform better on real users, and evaluation results on simulated data are more likely to predict results on real data more accurately.

*This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and by the EU FP7 Programme under grant agreement 216594 (CLASSiC project: www.classic-project.org).

This is one of the reasons why learning user simulation models from data on real user behaviour has become an important direction of research (Scheffler and Young, 2001; Cuayáhuitl et al., 2005; Georgila et al., 2006). However, the data driven user models developed so far lack the complexity required for training high quality policies in task domains where user behaviour is relatively complex. Handcrafted models are still the most effective in those cases.

This paper presents an agenda-based user simulator which is handcrafted for a large part, but additionally can be trained with data from real users (Section 2). As a result, it generates behaviour that better reflects the statistics of real user behaviour, whilst preserving the complexity and rationality required to effectively train dialogue management policies. The trainable part is formed by a set of *random decision points*, which, depending on the context, may or may not be encountered during the process of receiving a system act and deciding on a response act. If such a point is encountered, the simulator makes a random decision between a number of options which may directly or indirectly influence the resulting output. The options for each random decision point are reasonable in the context in which it is encountered, but a uniform distribution of outcomes might not reflect real user behaviour.

We will describe a sample-based method for estimating the parameters that define the probabilities for each possible decision, using data on real users from a corpus of human-machine dialogues (Section 3). Evaluation results will be presented both in terms of statistics on generated user behaviour and the quality of dialogue policies trained with different user simulations (Section 4).

2 Agenda-based user simulation

In agenda-based user simulation, user acts are generated on the basis of a *user goal* and an *agenda* (Schatzmann et al., 2007a). The simulator presented here is developed and used for a tourist in-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 116–123,
The University of Tokyo, September 24-25, 2010.

formation application, but is sufficiently generic to accommodate slot-filling applications in any domain.[1] The user goal consists of the type of venue, for example `hotel`, `bar` or `restaurant`, a list of constraints in the form of slot value pairs, such as `food=Italian` or `area=east`, and a list of slots the user wants to know the value of, such as the address (`addr`), phone number (`phone`), or price information (`price`) of the venue. The user goals for the simulator are randomly generated from the domain ontology describing which combinations of venue types and constraints are allowed and what are the possible values for each slot. The agenda is a stack-like structure containing planned user acts. When the simulator receives a system act, the status of the user goal is updated as well as the agenda, typically by pushing new acts onto it. In a separate step, the response user act is selected by popping one or more items off the agenda.

Although the agenda-based user simulator introduced by Schatzmann et al. (2007a) was entirely handcrafted, it was realistic enough to successfully test a prototype POMDP dialogue manager and train a dialogue policy that outperformed a handcrafted baseline (Young et al., 2009). A method to train an agenda-based user simulator from data was proposed by Schatzmann et al. (2007b). In this approach, operations on the agenda are controlled by probabilities learned from data using a variation of the EM algorithm. However, this approach does not readily scale to more complex interactions in which users can, for example, change their goal midway through a dialogue.

2.1 Random decision parameters

Each time the user simulator receives a system act, a complex, two-fold process takes place involving several decisions, made on the basis of both the nature of the incoming system act and the information state of the user, i.e., the status of the user goal and agenda. The first phase can be seen as an information state update and involves actions like filling requested slots or checking whether the provided information is consistent with the user goal constraints. In the second phase, the user decides which response act to generate, based on the updated agenda. Many of the decisions involved are deterministic, allowing only one possible option given the context. Other decisions allow for some degree of variation in the user behaviour and are governed by probability distributions over the options allowed in that context. For example, if the system has offered a venue that matches the user's goal, the user can randomly decide to either change his goal or to accept the venue and ask for additional information such as the phone number.

The non-deterministic part of the simulator is formalised in terms of a set of *random decision points* (RDPs) embedded in the decision process. If an RDP is encountered (depending on the context), a random choice between the options defined for that point is made by sampling from a probability distribution. Most of the RDPs are controlled by a multinomial distribution, such as deciding whether or not to change the goal after a system offer. Some RDPs are controlled by a geometric distribution, like in the case where the user is planning to specify one of his constraints (with an inform act popped from the agenda) and then repeatedly adds an additional constraint to the act (by combining it with an additional inform act popped from the agenda) until it randomly decides not to add any more constraints (or runs out of constraints to specify). The parameter for this distribution thus controls how cautious the user is in providing information to the system.

Hence, the user simulator can be viewed as a 'decision network', consisting of deterministic and random decision points. This is illustrated in Figure 1 for the simplified case of a network with only four RDPs; the actual simulator has 23 RDPs, with 27 associated parameters in total. Each time the simulator receives a system act, it follows a path through the network, which is partly determined by that system act and the user goal and agenda, and partly by random decisions made according to the probability distributions for each random decision point i given by its parameters $\theta_{\mathbf{i}}$.

3 Training the simulator from data

The parameterisation of the user simulator as described in Section 2.1 forms the basis for a method for training the simulator with real user data. The parameters describing the probability distributions for each RDP are estimated in order to generate user behaviour that fits the user behaviour in the corpus as closely as possible. In order to do so, a sample based maximum likelihood approach is taken, in which the simulator is run repeatedly against the system acts in the corpus, and the random decisions that lead to simulated acts matching the true act in the corpus are recorded. The parameters are then estimated using the counts for each of the random decision points.

[1] We have to date also implemented systems in appointment scheduling and bus timetable inquiries.

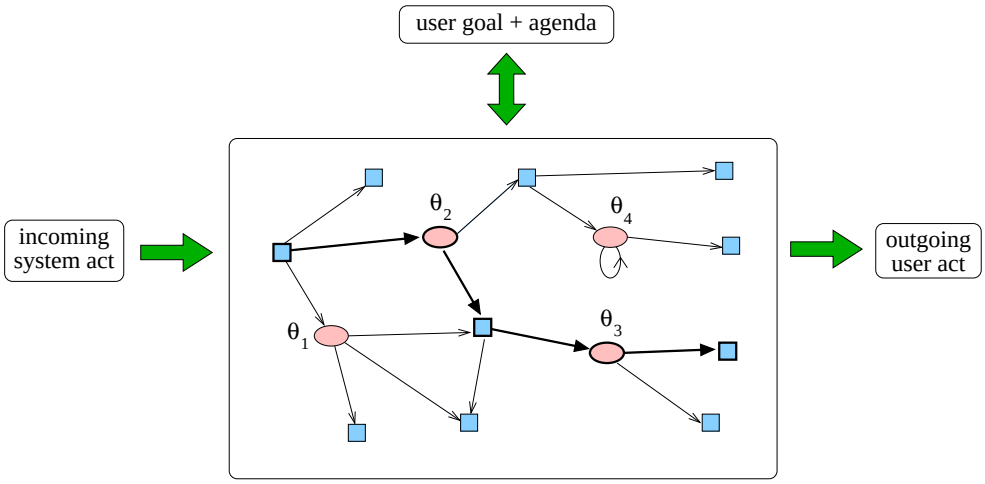


Figure 1: User simulator viewed as a 'decision network': square nodes indicate deterministic decision points; round nodes indicate random decision points, and have associated parameters θ_i; the loop on one of the nodes indicates it has a geometric distribution associated with it.

3.1 Parameter estimation

Before starting the process of matching simulated acts with true acts and collecting counts for the RDPs, the parameters are initialised to values corresponding to uniform distributions. Then, the simulator is run against all dialogues in the corpus in such a way that for each turn in a dialogue (consisting of a system act and a user act), the user simulator is provided with the system act and is run repeatedly to generate several simulated user response acts for that turn. For the first turn of a dialogue, the simulator is initialised with the correct user state (see Section 3.2). For each response, the simulator may make different random decisions, generally leading to different user acts. The decisions that lead to a simulated act that matches the true act are recorded as successful. By generating a sufficiently large number of simulated acts, all possible combinations of decisions are explored to find a matching act. Given the high complexity of the simulator, this sampling approach is preferred over directly enumerating all decision combinations to identify the successful ones. If none of the combinations are successful, then either a) the processing of the dialogue is ended, or b) the correct context is set for the next turn and processing is continued. Whereas the former approach aims at matching sequences of turns, the latter only aims at matching each user turn separately. In either case, after all data is processed, the parameters are estimated using the resulting counts of successful decisions for each of the RDPs.

For each RDP i, let DP_i represent the decision taken, and d_{ij} the j'th possible decision. Then, for each decision point i that is controlled by a multinomial distribution, the corresponding parameter estimates θ_{ij} are obtained as follows from the decision frequencies $c(DP_i = d_{ij})$:

$$\theta_{ij} = \frac{c(DP_i = d_{ij})}{\sum_j c(DP_i = d_{ij})} \quad (1)$$

Random decision points that are controlled by geometric distributions involve potentially multiple random decisions between two options (Bernoulli trials). The parameters for such RDPs are estimated as follows:

$$\theta_i = \left(\frac{1}{n} \sum_{k=1}^{n} b_{ik} \right)^{-1} \quad (2)$$

where b_{ik} is the number of Bernoulli trials required at the k'th time decision point i was encountered. In terms of the decision network, this estimate is correlated with the average number of times the loop of the node was taken.

3.2 User goal inference

In order to be able to set the correct user goal state in any given turn, a set of update rules is used to infer the user's goals from a dialogue beforehand, on the basis of the entire sequence of system acts and 'true' user acts (see Section 4.1) in the corpus. These update rules are based on the notion of *dialogue act preconditions*, which specify conditions of the dialogue context that must hold for a dialogue agent to perform that act. For example, a precondition for the act `inform(area=central)` is that the speaker wants a venue in the centre. The user act model

of the HIS dialogue manager is designed according to this same notion (Keizer et al., 2008). In this model, the probability of a user act in a certain dialogue context (the last system act and a hypothesis regarding the user goal) is determined by checking the consistency of its preconditions with that context. This contributes to updating the system's belief state on the basis of which it determines its response action. For the user goal inference model, the user act is given and therefore its preconditions can be used to directly infer the user goal. So, for example, in the case of observing the user act `inform(area=central)`, the constraint `(area=central)` is added to the user goal.

In addition to using the inferred user goals, the agenda is corrected in cases where there is a mismatch between real and simulated user acts in the previous turn.

In using this offline goal inference model, our approach takes a position between (Schatzmann et al., 2007b), in which the user's goal is treated as hidden, and (Georgila et al., 2006), in which the user's goal is obtained directly from the corpus annotation.

4 Evaluation

The parameter estimation technique for training the user simulator was evaluated in two different ways. The first evaluation involved comparing the statistics of simulated and real user behaviour. The second evaluation involved comparing dialogue manager policies trained with different simulators.

4.1 Data

The task of the dialogue systems we are developing is to provide tourist information to users, involving venues such as bars, restaurants and hotels that the user can search for and ask about. These venues are described in terms of features such as price range, area, type of food, phone number, address, and so on. The kind of dialogues with these systems are commonly called slot-filling dialogues.

Within the range of slot-filling applications the domain is relatively complex due to its hierarchical data structure and relatively large number of slots and their possible values. Scalability is indeed one of the primary challenges to be addressed in statistical approaches to dialogue system development, including user simulation.

The dialogue corpus that was used for training and evaluating the simulator was obtained from the evaluation of a POMDP spoken dialogue system with real users. All user utterances in the resulting corpus were transcribed and semantically annotated in terms of dialogue acts. Dialogue acts consist of a series of semantic items, including the type (describing the intention of the speaker, e.g., `inform` or `request`) and a list of slot value pairs (e.g., `food=Chinese` or `area=south`). An extensive analysis of the annotations from three different people revealed a high level of inter-annotator agreement (ranging from 0.81 to 0.94, depending on which pair of annotations are compared), and a voting scheme for selecting a single annotation for each turn ensured the reliability of the 'true' user acts used for training the simulator.

4.2 Corpus statistics results

A first approach to evaluating user simulations is to look at the statistics of the user behaviour that is generated by a simulator and compare it with that of real users as observed in a dialogue corpus. Several metrics for such evaluations have been considered in the literature, all of which have both strong points and weaknesses. For the present evaluation, a selection of metrics believed to give a reasonable first indication of the quality of the user simulations was considered[2].

4.2.1 Metrics

The first corpus-based evaluation metric is the **Log Likelihood (LL)** of the data, given the user simulation model. This is what is in fact maximised by the parameter estimation algorithm. The log likelihood can be computed by summing the log probabilities of each user turn d_u in the corpus data $\mathcal{D}$:

$$ll(\mathcal{D}|\{\theta_{ij}\},\{\theta_i\}) = \sum_u \log P(d_u|\{\theta_{ij}\},\{\theta_i\}) \quad (3)$$

The user turn probability is given by the probability of the decision paths (directed paths in the decision network of maximal length, such as the one indicated in Figure 1 in bold) leading to a simulated user act in that turn that matches the true user act. The probability of a decision path is obtained by multiplying the probabilities of the decisions made at each decision point i that was encountered, which are given by the parameters θ_{ij}

[2]Note that not all selected metrics are metrics in the strict sense of the word; the term should therefore be interpreted as a more general one.

and θ_i:

$$\log P(d_u|\{\theta_{ij}\},\{\theta_i\}) = \sum_{i\in I^m(u)} \log\big(\sum_j \theta_{ij}\cdot\delta_{ij}(u)\big) + \sum_{i\in I^g(u)} \log\big(\sum_k (1-\theta_i)^{k-1}\cdot\theta_i\cdot\mu_{ik}(u)\big) \quad (4)$$

where $I^m(u) = \{i \in I^m | \sum_j \delta_{ij}(u) > 0\}$ and $I^g(u) = \{i \in I^g | \sum_k \mu_{ik}(u) > 0\}$ are the subsets of the multinomial (I^m) and geometric (I^g) decision points respectively containing those points that were encountered in any combination of decisions resulting in the given user act:

$$\delta_{ij}(u) = \begin{cases} 1 & \text{if decision } DP_i = d_{ij} \text{ was taken in any of the matching combinations} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

$$\mu_{ik}(u) = \begin{cases} 1 & \text{if any of the matching combinations required } k > 0 \text{ trials} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

It should be noted that the log likelihood only represents those turns in the corpus for which the simulated user can produce a matching simulated act with some probability. Hence, it is important to also take into account the **corpus coverage** when considering the log likelihood in corpus based evaluation. Dividing by the number of matched turns provides a useful normalisation in this respect.

The expected **Precision (PRE)**, **Recall (RCL)**, and **F-Score (FS)** are obtained by comparing the simulated user acts with the true user acts in the same context (Georgila et al., 2006). These scores are obtained by pairwise comparison of the simulated and true user act for each turn in the corpus at the level of the semantic items:

$$PRE = \frac{\#(matched\ items)}{\#(items\ in\ simulated\ act)} \quad (7)$$

$$RCL = \frac{\#(matched\ items)}{\#(items\ in\ true\ act)} \quad (8)$$

$$FS = \frac{2\cdot PRE\cdot RCL}{PRE+RCL} \quad (9)$$

By sampling a sufficient number of simulated acts for each turn in the corpus and comparing them with the corresponding true acts, this results in an accurate measure on average.

The problem with precision and recall is that they are known to heavily penalise unseen data. Any attempt to generalise and therefore increase the variability of user behaviour results in lower scores.

Another way of evaluating the user simulator is to look at the global user act distributions it generates and compare them to the distributions found in the real user data. A common metric for comparing such distributions is the **Kullback-Leibler (KL) distance**. In (Cuayáhuitl et al., 2005) this metric was used to evaluate an HMM-based user simulation approach. The KL distance is computed by taking the average of the two KL divergences[3] $D_{KL}(simulated||true)$ and $D_{KL}(true||simulated)$, where:

$$D_{KL}(p||q) = \sum_i p_i \cdot log_2(\frac{p_i}{q_i}) \quad (10)$$

KL distances are computed for both full user act distributions (taking into account both the dialogue act type and slot value pairs) and user act type distributions (only regarding the dialogue act type), denoted by KLF and KLT respectively.

4.2.2 Results

For the experiments, the corpus data was randomly split into a training set, consisting of 4479 user turns in 541 dialogues, used for estimating the user simulator parameters, and a test set, consisting of 1457 user turns in 175 dialogues, used for evaluation only. In the evaluation, the following parameter settings were compared: 1) non-informative, uniform parameters (UNIF); 2) handcrafted parameters (HDC); 3) parameters estimated from data (TRA); and 4) deterministic parameters (DET), in which for each RDP the probability of the most probable decision according to the estimated parameters is set to 1, i.e., at all times, the most likely decision according to the estimated parameters is chosen.

For both trained and deterministic parameters, a distinction is made between the two approaches to matching user acts during parameter estimation. Recall that in the turn-based approach, in each turn, the simulator is run with the corrected context to find a matching simulated act, whereas in the sequence-based approach, the matching process for a dialogue is stopped in case a turn is encountered which cannot be matched by the simulator. This results in estimated parameters TRA-T and deterministic parameters DET-T for

[3] Before computing the distances, add-one smoothing was applied in order to avoid zero-probabilities.

PAR	nLL-T	nLL-S	PRE	RCL	FS	KLF	KLT
UNIF	-3.78	-3.37	16.95 (± 0.75)	9.47 (± 0.59)	12.15	3.057	2.318
HDC	-4.07	-2.22	44.31 (± 0.99)	34.74 (± 0.95)	38.94	1.784	0.623
TRA-T	$\mathbf{-2.97}$	-	37.60 (± 0.97)	28.14 (± 0.90)	32.19	1.362	0.336
DET-T	$-\infty$	-	47.70 (± 1.00)	40.90 (± 0.98)	44.04	2.335	0.838
TRA-S	-	$\mathbf{-2.13}$	43.19 (± 0.99)	35.68 (± 0.96)	39.07	$\mathbf{1.355}$	$\mathbf{0.155}$
DET-S	-	$-\infty$	$\mathbf{49.39}$ (± 1.00)	$\mathbf{43.04}$ (± 0.99)	$\mathbf{46.00}$	2.310	0.825

Table 1: Results of the sample-based user simulator evaluation on the Mar'09 training corpus (the corpus coverage was 59% for the turn-based and 33% for the sequence-based matching approach).

PAR	nLL-T	nLL-S	PRE	RCL	FS	KLF	KLT
UNIF	-3.61	-3.28	16.59 (± 1.29)	9.32 (± 1.01)	11.93	2.951	2.180
HDC	-3.90	-2.19	45.35 (± 1.72)	36.04 (± 1.66)	40.16	1.780	0.561
TRA-T	$\mathbf{-2.84}$	-	38.22 (± 1.68)	28.74 (± 1.57)	32.81	$\mathbf{1.405}$	0.310
DET-T	$-\infty$	-	49.15 (± 1.73)	42.17 (± 1.71)	45.39	2.478	0.867
TRA-S	-	$\mathbf{-2.12}$	43.90 (± 1.72)	36.52 (± 1.67)	39.87	1.424	$\mathbf{0.153}$
DET-S	-	$-\infty$	$\mathbf{50.73}$ (± 1.73)	$\mathbf{44.41}$ (± 1.72)	$\mathbf{47.36}$	2.407	0.841

Table 2: Results of the sample-based user simulator evaluation on the Mar'09 test corpus (corpus coverage 59% for the turn-based, and 36% for sequence-based matching).

the turn-based approach and analogously TRA-S and DET-S for the sequence-based approach. The corresponding normalised (see Section 4.2.1) log-likelihoods are indicated by nLL-T and nLL-S.

Tables 1 and 2 give the results on the training and test data respectively. The results show that in terms of log-likelihood and KL-distances, the estimated parameters outperform the other settings, regardless of the matching method. In terms of precision/recall (given in percentages with 95% confidence intervals), the estimated parameters are worse than the handcrafted parameters for turn-based matching, but have similar scores for sequence-based matching.

The results for the deterministic parameters illustrate that much better precision/recall scores can be obtained, but at the expense of variability as well as the KL-distances. It will be easier to train a dialogue policy on such a deterministic simulator, but that policy is likely to perform significantly worse on the more varied behaviour generated by the trained simulator, as we will see in Section 4.3.

Out of the two matching approaches, the sequence-based approach gives the best results: TRA-S outperforms TRA-T on all scores, except for the coverage which is much lower for the sequence-based approach (33% vs. 59%).

4.3 Policy evaluation results

Although the corpus-based evaluation results give a useful indication of how realistic the behaviour generated by a simulator is, what really should be evaluated is the dialogue management policy that is trained using that simulator. Therefore, different parameter sets for the simulator were used to train and evaluate different policies for the Hidden Information State (HIS) dialogue manager (Young et al., 2009). Four different policies were trained: one policy using handcrafted simulation parameters (POL-HDC); two policies using simulation parameters estimated (using the sequence-based matching approach) from two data sets that were obtained by randomly splitting the data into two parts of 358 dialogues each (POL-TRA1 and POL-TRA2); and finally, a policy using a deterministic simulator (POL-DET) constructed from the trained parameters as discussed in Section 4.2.2. The policies were then each evaluated on the simulator using the four parameter settings at different semantic error rates.

The performance of a policy is measured in terms of a reward that is given for each dialogue, i.e. a reward of 20 for a successful dialogue, minus the number of turns. A dialogue is considered successful if the system has offered a venue matching the predefined user goal constraints and has given the correct values of all requested slots for this venue. During the policy optimisation, in which a reinforcement learning algorithm tries to optimise the expected long term reward, this dialogue scoring regime was also used.

In Figures 2, 3, and 4, evaluation results are given resulting from running 3000 dialogues at each of 11 different semantic error rates. The curves show average rewards with 95% confidence intervals. The error rate is controlled by a hand-

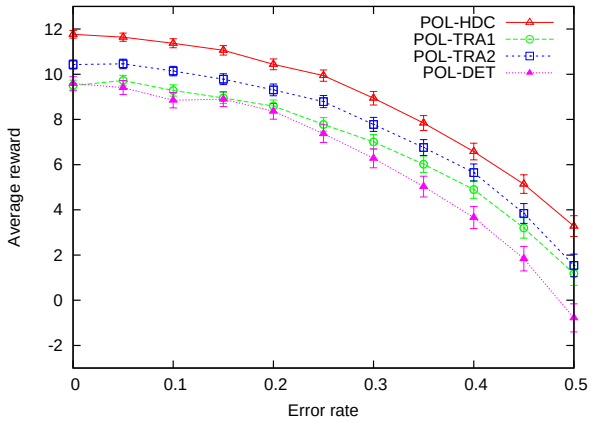


Figure 2: Average rewards for each policy when evaluated on UM-HDC.

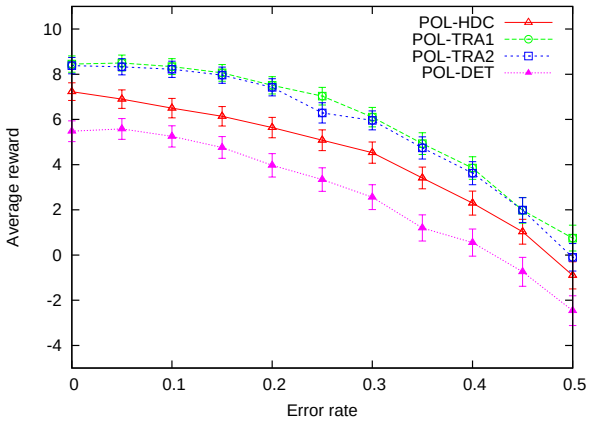


Figure 3: Average rewards for each policy when evaluated on UM-TRA1.

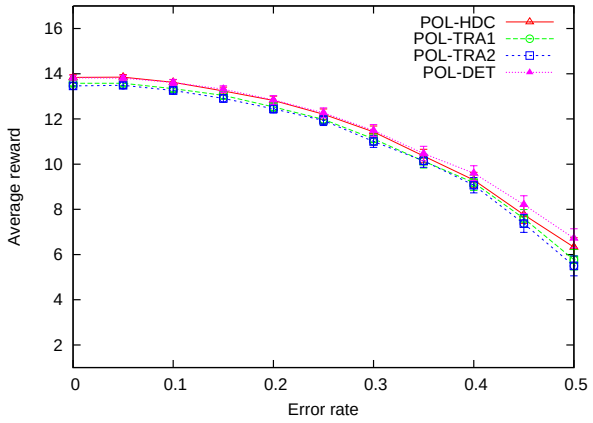


Figure 4: Average rewards for each policy when evaluated on UM-DET.

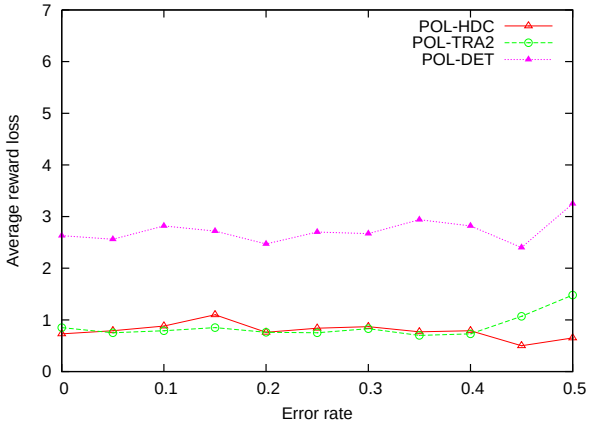


Figure 5: Average loss in reward for each policy, across three different simulators.

crafted error model that converts the user act generated by the simulator into an n-best list of dialogue act hypotheses.

The policy that was trained using the handcrafted simulator (POL-HDC) outperforms the other policies when evaluated on that same simulator (see Figure 2), and both policies trained using the trained simulators (POL-TRA1 and POL-TRA2) outperform the other policies when evaluated on either trained simulator (see Figure 3 for the evaluation on UM-TRA1; the evaluation on UM-TRA2 is very similar and therefore omitted). There is little difference in performance between policies POL-TRA1 and POL-TRA2, which can be explained by the fact that the two trained parameter settings are quite similar, in contrast to the handcrafted parameters. The policy that was trained on the deterministic parameters (POL-DET) is competitive with the other policies when evaluated on UM-DET (see Figure 4), but performs significantly worse on the other parameter settings which generate the variation in behaviour that the dialogue manager did not encounter during training of POL-DET.

In addition to comparing the policies when evaluated on each simulator separately, another comparison was made in terms of the average performance across all simulators. For each policy and each simulator, we first computed the difference between the policy's performance and the 'maximum' performance on that simulator as achieved by the policy that was also trained on that simulator, and then averaged over all simulators. To avoid biased results, only one of the trained simulators was included. The results in Figure 5 show that the POL-TRA2 policy is more robust than POL-DET, and has similar robustness as POL-HDC. Similar results are obtained when including UM-TRA1 only.

Given that the results of Section 4.2 show that the dialogues generated by the trained simulator more closely match real corpus data, and given that the above simulation results show that the POL-TRA policies are at least as robust as the

other policies, it seems likely that policies trained using the trained user simulator will show improved performance when evaluated on real users.

However, this claim can only be properly demonstrated in a real user evaluation of the dialogue system containing different dialogue management policies. Such a user trial would also be able to confirm whether the results from evaluations on the trained simulator can more accurately predict the actual performance expected with real users.

5 Conclusion

In this paper, we presented an agenda-based user simulator extended to be trainable on real user data whilst preserving the necessary rationality and complexity for effective training and evaluation of dialogue manager policies. The extension involved the incorporation of random decision points in the process of receiving and responding to a system act in each turn. The decisions made at these points are controlled by probability distributions defined by a set of parameters.

A sample-based maximum likelihood approach to estimating these parameters from real user data in a corpus of human-machine dialogues was discussed, and two kinds of evaluations were presented. When comparing the statistics of real versus simulated user behaviour in terms of a selection of different metrics, overall, the estimated parameters were shown to give better results than the handcrafted baselines. When evaluating dialogue management policies trained on the simulator with different parameter settings, it was shown that: 1) policies trained on a particular parameter setting outperform other policies when evaluated on the same parameters, and in particular, 2) a policy trained on the trained simulator outperforms other policies on a trained simulator. With the general goal of obtaining a dialogue manager that performs better in practice, these results are encouraging, but need to be confirmed by an evaluation of the policies on real users.

Additionally, there is still room for improving the quality of the simulator itself. For example, the variation in user behaviour can be improved by adding more random decision points, in order to achieve better corpus coverage. In addition, since there is no clear consensus on what is the best metric for evaluating user simulations, additional metrics will be explored in order to get a more balanced indication of the quality of the user simulator and how the various metrics are affected by modifications to the simulator. Perplexity (related to the log likelihood, see (Georgila et al., 2005)), accuracy (related to precision/recall, see (Zukerman and Albrecht, 2001; Georgila et al., 2006)), and Cramér-von Mises divergence (comparing dialogue score distributions, see (Williams, 2008)) are some of the metrics worth considering.

References

H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira. 2005. Human-computer dialogue simulation using hidden markov models. In *Proc. ASRU'05*, pages 290–295.

K. Georgila, J. Henderson, and O. Lemon. 2005. Learning user simulations for information state update dialogue systems. In *Proc. Interspeech '05*.

K. Georgila, J. Henderson, and O. Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. Interspeech/ICSLP*.

S. Keizer, M. Gašić, F. Mairesse, B. Thomson, K. Yu, and S. Young. 2008. Modelling user behaviour in the HIS-POMDP dialogue manager. In *Proc. SLT*, Goa, India.

E. Levin, R. Pieraccini, and W. Eckert. 2000. A stochastic model of human-machine interaction for learning dialogue strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1).

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. 2007a. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Proceedings HLT/NAACL*, Rochester, NY.

J. Schatzmann, B. Thomson, and S. Young. 2007b. Statistical user simulation with a hidden agenda. In *Proc. SIGDIAL'07*, pages 273–282, Antwerp, Belgium.

K. Scheffler and S. Young. 2001. Corpus-based dialogue simulation for automatic strategy learning and evaluation. In *Proceedings NAACL Workshop on Adaptation in Dialogue*.

S. Singh, M. Kearns, D. Litman, and M. Walker. 2000. Reinforcement learning for spoken dialogue systems. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems (NIPS)*. MIT Press.

J. Williams. 2008. Evaluating user simulations with the Cramér-von Mises divergence. *Speech Communication*, 50:829–846.

S. Young, M. Gašić, S. Keizer, F. Mairesse, B. Thomson, and K. Yu. 2009. The Hidden Information State model: a practical framework for POMDP based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.

I. Zukerman and D. Albrecht. 2001. Predictive statistical models for user modeling. *User Modeling and User-Adapted Interaction*, 11:5–18.

Adaptive Referring Expression Generation in Spoken Dialogue Systems: Evaluation with Real Users

Srinivasan Janarthanam
School of Informatics
University of Edinburgh
s.janarthanam@ed.ac.uk

Oliver Lemon
Interaction Lab
Mathematics and Computer Science
Heriot-Watt University
o.lemon@hw.ac.uk

Abstract

We present new results from a real-user evaluation of a data-driven approach to learning user-adaptive referring expression generation (REG) policies for spoken dialogue systems. Referring expressions can be difficult to understand in technical domains where users may not know the technical 'jargon' names of the domain entities. In such cases, dialogue systems must be able to model the user's (lexical) domain knowledge and use appropriate referring expressions. We present a reinforcement learning (RL) framework in which the system learns REG policies which can adapt to unknown users online. For real users of such a system, we show that in comparison to an adaptive hand-coded baseline policy, the learned policy performs significantly better, with a 20.8% average increase in adaptation accuracy, 12.6% decrease in time taken, and a 15.1% increase in task completion rate. The learned policy also has a significantly better subjective rating from users. This is because the learned policies adapt online to changing evidence about the user's domain expertise. We also discuss the issue of evaluation in simulation versus evaluation with real users.

1 Introduction

We present new results from an evaluation with real users, for a reinforcement learning (Sutton and Barto, 1998) framework to learn user-adaptive referring expression generation policies from data-driven user simulations. Such a policy allows the system to choose appropriate expressions to refer to domain entities in a dialogue setting. For instance, in a technical support conversation, the system could choose to use more technical terms with an expert user, or to use more descriptive and general expressions with novice users, and a mix of the two with intermediate users of various sorts (see examples in Table 1).

Jargon: Please plug one end of the `broadband cable` into the `broadband filter`.
Descriptive: Please plug one end of the `thin white cable with grey ends` into the `small white box`.

Table 1: Referring expression examples for 2 entities (from the corpus)

In natural human-human conversations, dialogue partners learn about each other and adapt their language to suit their domain expertise (Issacs and Clark, 1987). This kind of adaptation is called `Alignment through Audience Design` (Clark and Murphy, 1982; Bell, 1984). We assume that users are mostly unknown to the system and therefore that a spoken dialogue system (SDS) must be capable of observing the user's dialogue behaviour, modelling his/her domain knowledge, and adapting accordingly, just like human interlocutors. Therefore unlike systems that use static user models, our system has to dynamically model the user's domain knowledge in order to adapt during the conversation.

We present a corpus-driven framework for learning a user-adaptive REG policy from a small corpus of non-adaptive human-machine interaction. We show that the learned policy performs better than a simple hand-coded adaptive policy in terms of accuracy of adaptation, dialogue time and task completion rate when evaluated with real users in a wizarded study.

In section 2, we present some of the related work. Section 3 and section 4 describe the dialogue system framework and the user simulation

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 124–131,
The University of Tokyo, September 24-25, 2010.

model. In section 5, we present the training and in section 6, we present the evaluation for different REG policies with real users.

2 Related work

Rule-based and supervised learning approaches have been proposed to learn and adapt during conversations dynamically. Such systems learn from a user at the start and later adapt to the domain knowledge of the user. However, they either require expensive expert knowledge resources to hand-code the inference rules (Cawsey, 1993) or a large corpus of expert-layperson interaction from which adaptive strategies can be learned and modelled, using methods such as Bayesian networks (Akiba and Tanaka, 1994). In contrast, we present an approach that learns in the absence of these expensive resources. It is also not clear how supervised approaches choose between when to seek more information and when to adapt. In this study, we show that using reinforcement learning this decision is learned automatically.

Reinforcement Learning (RL) has been successfully used for learning dialogue management policies since (Levin et al., 1997). The learned policies allow the dialogue manager to optimally choose appropriate dialogue acts such as instructions, confirmation requests, and so on, under uncertain noise or other environment conditions. There have been recent efforts to learn information presentation and recommendation strategies using reinforcement learning (Hernandez et al., 2003; Rieser and Lemon, 2009; Rieser and Lemon, 2010), and joint optimisation of Dialogue Management and NLG using hierarchical RL has been proposed by (Lemon, 2010). In addition, we present a framework to learn to choose appropriate referring expressions based on a user's domain knowledge. Following a proof-of-concept study using a hand-coded rule-based user simulation (Janarthanam and Lemon, 2009c), we previously showed that adaptive REG policies can be learned using an RL framework with data-driven user simulations and that such policies perform better than simple hand-coded policies (Janarthanam and Lemon, 2010).

3 The Dialogue System

In this section, we describe the different modules of the dialogue system. The interaction between the different modules is shown in figure 1 (in learning mode). The dialogue system presents the user with instructions to setup a broadband connection at home. In the Wizard of Oz setup, the system and the user interact using speech. However, in our machine learning setup, they interact at the abstract level of dialogue actions and referring expressions. Our objective is to learn to choose the appropriate referring expressions to refer to the domain entities in the instructions.

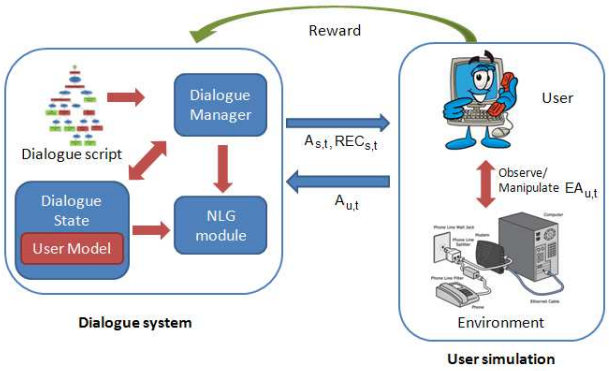


Figure 1: System User Interaction (learning)

3.1 Dialogue Manager

The dialogue manager identifies the next dialogue act ($A_{s,t}$ where t denotes turn, s denotes system) to give to the user based on the dialogue management policy π_{dm}. The dialogue management is coded in the form of a finite state machine. In this dialogue task, the system provides instructions to either observe or manipulate the environment. When users ask for clarifications on referring expressions, the system clarifies ($provide\_clar$) by giving information to enable the user to associate the expression with the intended referent. When the user responds in any other way, the instruction is simply repeated. The dialogue manager is also responsible for updating and managing the system state $S_{s,t}$ (see section 3.2). The system interacts with the user by passing both the system action $A_{s,t}$ and the referring expressions $REC_{s,t}$ (see section 3.3).

3.2 The dialogue state

The dialogue state $S_{s,t}$ is a set of variables that represent the current state of the conversation. In our study, in addition to maintaining an overall dialogue state, the system maintains a user model $UM_{s,t}$ which records the initial domain knowledge of the user. It is a dynamic model that starts

with a state where the system does not have any knowledge about the user. Since the model is updated according to the user's behaviour, it may be inaccurate if the user's behaviour is itself uncertain. Hence, the user model used in this system is not always an accurate model of the user's knowledge and reflects a level of uncertainty about the user.

Each jargon referring expression `x` is represented by a three-valued variable in the dialogue state: `user_knows_x`. The three values that each variable takes are `yes`, `no`, `not_sure`. The variables are updated using a simple user model update algorithm after the user's response each turn. Initially each variable is set to `not_sure`. If the user responds to an instruction containing the referring expression `x` with a clarification request, then `user_knows_x` is set to `no`. Similarly, if the user responds with appropriate information to the system's instruction, the dialogue manager sets `user_knows_x` is set to `yes`. Only the user's initial knowledge is recorded. This is based on the assumption that an estimate of the user's initial knowledge helps to predict the user's knowledge of the rest of the referring expressions.

3.3 REG module

The REG module is a part of the NLG module whose task is to identify the list of domain entities to be referred to and to choose the appropriate referring expression for each of the domain entities for each given dialogue act. In this study, we focus only on the production of appropriate referring expressions to refer to domain entities mentioned in the dialogue act. It chooses between the two types of referring expressions - jargon and descriptive. For example, the domain entity *broadband_filter* can be referred to using the jargon expression "broadband filter" or using the descriptive expression "small white box"[1]. Although adaptation is the primary goal, it should be noted that in order to get an idea of the user the system is dealing with, it needs to seek information using jargon expressions.

The REG module operates in two modes - learning and evaluation. In the learning mode, the REG module is the learning agent. The REG module learns to associate dialogue states with optimal referring expressions. This is represented by a REG policy $\pi_{reg} : UM_{s,t} \rightarrow REC_{s,t}$, which maps the states of the dialogue (user model) to optimal referring expressions. The referring expression choices $REC_{s,t}$ is a set of pairs identifying the referent R and the type of expression T used in the current system utterance. For instance, the pair $(\textit{broadband\_filter}, desc)$ represents the descriptive expression "small white box".

$$REC_{s,t} = \{(R_1, T_1), ..., (R_n, T_n)\}$$

In the evaluation mode, a trained REG policy interacts with unknown users. It consults the learned policy π_{reg} to choose the referring expressions based on the current user model.

4 User Simulations

In this section, we present user simulation models that simulate the dialogue behaviour of a real human user. Several user simulation models have been proposed for use in reinforcement learning of dialogue policies (Georgila et al., 2005; Schatzmann et al., 2006; Schatzmann et al., 2007; Ai and Litman, 2007). However, they are suited only for learning dialogue management policies, and not natural language generation policies. In particular, our model is the first to be sensitive to a system's choices of referring expressions. Earlier, we presented a two-tier simulation trained on data precisely for REG policy learning (Janarthanam and Lemon, 2009a). However, it is not suited for training on small corpus like the one we have at our disposal. In contrast to the earlier model, we now condition the clarification requests on the referent class rather than the referent itself to handle the data sparsity problem.

4.1 Corpus-driven action selection model

The user simulation (US) receives the system action $A_{s,t}$ and its referring expression choices $REC_{s,t}$ at each turn. The US responds with a user action $A_{u,t}$ (u denoting user). This can either be a clarification request (cr) or an instruction response (ir). The US produces a clarification request cr based on the class of the referent $C(R_i)$, type of the referring expression T_i, and the current domain knowledge of the user for the referring expression $DK_{u,t}(R_i, T_i)$. Domain entities whose jargon expressions raised clarification requests in the corpus were listed and those that had more than the mean number of clarification requests were classified as `difficult` and others as `easy` entities (for example, *power_adaptor* is `easy` - all

[1] We will use italicised forms to represent the domain entities (e.g. *broadband_filter*) and double quotes to represent the referring expressions (e.g. "broadband filter").

users understood this expression, *broadband_filter* is `difficult`). Clarification requests are produced using the following model.

$$P(A_{u,t} = cr(R_i, T_i)|C(R_i), T_i, DK_{u,t}(R_i, T_i))$$
$$\text{where } (R_i, T_i) \in REC_{s,t}$$

One should note that the actual literal expression is not used in the transaction. Only the entity that it is referring to (R_i) and its type (T_i) are used. However, the above model simulates the process of interpreting and resolving the expression and identifying the domain entity of interest in the instruction. The user identification of the entity is signified when there is no clarification request produced (i.e. $A_{u,t} = none$). When no clarification request is produced, the environment action $EA_{u,t}$ is generated using the following model.

$$P(EA_{u,t}|A_{s,t}) \text{ if } A_{u,t}! = cr(R_i, T_i)$$

Finally, the user action is an instruction response which is determined by the system action $A_{s,t}$. Instruction responses can be either $provide\_info$, $acknowledgement$ or $other$ based on the system's instruction.

$$P(A_{u,t} = ir|EA_{u,t}, A_{s,t})$$

All the above models were trained on our corpus data using *maximum likelihood estimation* and smoothed using a variant of *Witten-Bell discounting*. The corpus contained dialogues between a non-adaptive dialogue system and real users. According to the data, clarification requests are much more likely when jargon expressions are used to refer to the referents that belong to the `difficult` class and which the user doesn't know about. When the system uses expressions that the user knows, the user generally responds to the instruction given by the system.

4.2 User Domain knowledge

The user domain knowledge is initially set to one of several models at the start of every conversation. The models range from novices to experts which were identified from the corpus using `k-means` clustering. A novice user knows only "power adaptor", an expert knows all the jargon expressions and intermediate users know some. We assume that users can interpret the descriptive expressions and resolve their references. Therefore, they are not explicitly represented. We only code the user's knowledge of jargon expressions using boolean variables representing whether the user knows the expression or not.

4.3 Corpus

We trained the action selection model on a small corpus of 12 non-adaptive dialogues between real users and a dialogue system. There were six dialogues in which users interacted with a system using just jargon expressions and six with a system using descriptive expressions. For more discussions on our user simulation models and the corpus, please refer to (Janarthanam and Lemon, 2009b; Janarthanam and Lemon, 2009a; Janarthanam and Lemon, 2010).

5 Training

The REG module was trained (operated in learning mode) using the above simulations to learn REG policies that select referring expressions based on the user expertise in the domain. In this section, we discuss how to code the learning agent's goals as reward. We then discuss how the reward function is used to train the learning agent.

5.1 Reward function

We designed a reward function for the goal of adapting to each user's domain knowledge. We present the Adaptation Accuracy score (AA) that calculates how accurately the agent chose the appropriate expressions for each referent r, with respect to the user's knowledge. So, when the user knows the jargon expression for r, the appropriate expression to use is jargon, and if s/he doesn't know the jargon, a descriptive expression is appropriate. Although the user's domain knowledge is dynamically changing due to learning, we base appropriateness on the initial state, because our objective is to adapt to the initial state of the user $DK_{u,initial}$. However, in reality, designers might want their system to account for user's changing knowledge as well. We calculate accuracy per referent RA_r and then calculate the overall mean adaptation accuracy (AA) over all referents as shown below.

$$RA_r = \frac{\#(appropriate\_expressions(r))}{\#(instances(r))}$$
$$Adaptation Accuracy AA = \frac{1}{\#(r)} \Sigma_r RA_r$$

5.2 Learning

The REG module was trained in learning mode using the above reward function using the SHARSHA reinforcement learning algorithm (with linear function approximation) (Shapiro and Langley, 2002). This is a hierarchical variant of SARSA,

which is an on-policy learning algorithm that updates the current behaviour policy (see (Sutton and Barto, 1998)). The training produced approx. 5000 dialogues. The user simulation was calibrated to produce three types of users: Novice, Intermediate and Expert, randomly but with equal probability.

Initially, the REG policy chooses randomly between the referring expression types for each domain entity in the system utterance, irrespective of the user model state. Once the referring expressions are chosen, the system presents the user simulation with both the dialogue act and referring expression choices. The choice of referring expression affects the user's dialogue behaviour. For instance, choosing a jargon expression could evoke a clarification request from the user, based on which, the dialogue manager updates the internal user model ($UM_{s,t}$) with the new information that the user is ignorant of the particular expression. It should be noted that using a jargon expression is an *information seeking* move which enables the REG module to estimate the user's knowledge level. The same process is repeated for every dialogue instruction. At the end of the dialogue, the system is rewarded based on its choices of referring expressions. If the system chooses jargon expressions for novice users or descriptive expressions for expert users, penalties are incurred and if the system chooses REs appropriately, the reward is high. On the one hand, those actions that fetch more reward are reinforced, and on the other hand, the agent tries out new state-action combinations to explore the possibility of greater rewards. Over time, it stops exploring new state-action combinations and exploits those actions that contribute to higher reward. The REG module learns to choose the appropriate referring expressions based on the user model in order to maximize the overall adaptation accuracy. Figure 2 shows how the agent learns using the data-driven (**Learned DS**) during training. It can be seen in the figure 2 that towards the end the curve plateaus, signifying that learning has converged.

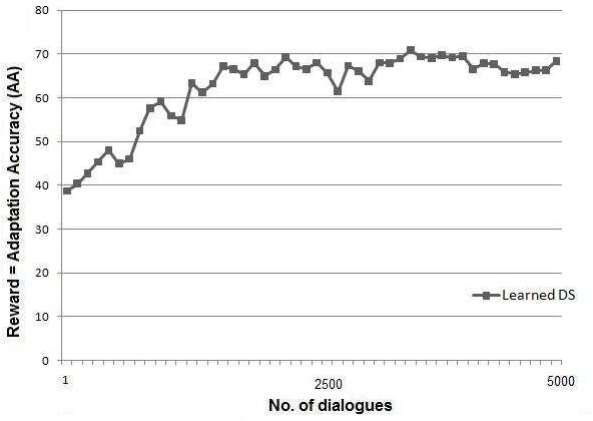


Figure 2: Learning curve - Training

6 Evaluation

In this section, we present the details of the evaluation process, the baseline policy, the metrics used, and the results. In a recent study, we evaluated the learned policy and several hand-coded baselines with simulated users and found that the Learned-DS policy produced higher adaptation accuracy than other policies (Janarthanam and Lemon, 2010). An interesting issue for research in this area is to what extent evaluation results obtained in simulated environments transfer to evaluations with real users (Lemon et al., 2006).

6.1 Baseline system

In order to compare the performance of the learned policy with a baseline, a simple rule-based policy was built. This baseline was chosen because it performed better in simulation, compared to a variety of other baselines (Janarthanam and Lemon, 2010). It uses jargon for all referents by default and provides clarifications when requested. It exploits the user model in subsequent references after the user's knowledge of the expression has been set to either `yes` or `no`. Therefore, although it is a simple policy, it adapts to a certain extent ('locally'). We refer to this policy as the 'Jargon-adapt' policy. It should be noted that this policy was built in the absence of expert domain knowledge and/or an expert-layperson corpus.

6.2 Process

We evaluated the two policies with real users. 36 university students from different backgrounds (e.g. Arts, Humanities, Medicine and Engineering) participated in the evaluation. 17 users were given a system with Jargon-adapt policy and 19 users interacted with a system with Learned-DS policy. Each user was given a pre-task recognition test to record his/her initial domain knowledge. The experimenter read out a list of technical terms and the user was asked to point out to the domain entities laid out in front of them. They were then

given one of the two systems - learned or baseline, to interact with. Following the system instructions, they then attempted to set up the broadband connection. When the dialogue had ended, the user was given a post-task test where the recognition test was repeated and their responses were recorded. The user's broadband connection setup was manually examined for task completion (i.e. the percentage of correct connections that they had made in their final set-up). The user was given the task completion results and was then given a user satisfaction questionnaire to evaluate the features of the system based on the conversation.

All users interacted with a wizarded system employing one of the two REG policies (see figure 3). The user's responses were intercepted by a human interpreter (or "wizard") and were annotated as dialogue acts, to which the automated dialogue manager responded with a system dialogue action (the dialogue policy was fixed). The wizards were not aware of the policy used by the system. The respective policies chose only the referring expressions to generate the system utterance for the given dialogue action. The system utterances were converted to speech by a speech synthesizer (Cereproc) and were played to the user.

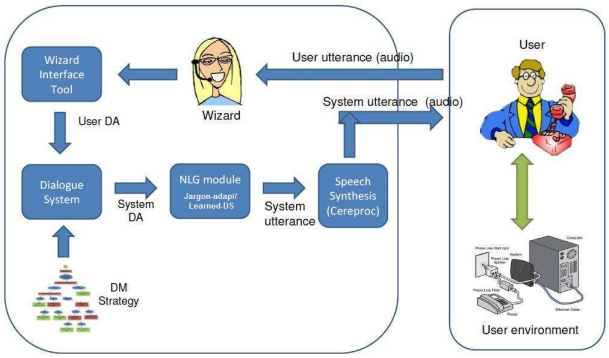


Figure 3: Wizarded Dialogue System

6.3 Metrics

In addition to the adaptation accuracy mentioned in section 5.1, we also measure other parameters from the conversation in order to show how learned adaptive policies compare with other policies on other dimensions. We also measure the learning effect on the users as (normalised) learning gain (LG) produced by using unknown jargon expressions. This is calculated using the pre- and post-test scores for the user domain knowledge (DK_u) as follows.

Metrics	Jargon-adapt	Learned-DS
AA	63.91	84.72 **
LG	0.59	0.61
DT	7.86	6.98 *
TC	84.7	99.8 **

* Statistical significance ($p < 0.05$).
** Statistical significance ($p < 0.001$).

Table 2: Evaluation with real users

$$Learning\ Gain\ LG = \frac{Post-Pre}{1-Pre}$$

Dialogue time (DT) is the actual time taken for the user to complete the task. We measured task completion (TC) by examining the user's broadband setup after the task was completed (i.e. the percentage of correct connections that they had made in their final set-up).

6.4 Results

We compare the performance of the two strategies on real users using objective parameters and subjective feedback scores. Tests for statistical significance were done using Mann-Whitney test for 2 independent samples (due to non-parametric nature of the data).

Table 2 presents the mean accuracy of adaptation (AA), learning gain (LG), dialogue time (DT), and task completion (TC), produced by the two strategies. The Learned-DS strategy produced more accurate adaptation than the Jargon-adapt strategy ($p<0.001$, U=9.0, r=-0.81). Higher accuracy of adaptation (AA) of the Learned-DS strategy translates to less dialogue time (U=73.0, $p<0.05$, r=-0.46) and higher task completion (U=47.5, $p<0.001$, r=-0.72) than the Jargon-adapt policy. However, there was no significant difference in learning gain (LG).

Table 3 presents how the users subjectively scored on a agreement scale of 1 to 4 (with 1 meaning 'strongly disagree'), different features of the system based on their conversations with the two different strategies. Users' feedback on different features of the systems were not very different from each other. However, users did feel that it was easier to identify domain objects with the Learned-DS strategy than the Jargon-adapt strategy (U=104.0, $p<0.05$, r=-0.34). To our knowledge, this is the first study to show a significant improvement in real user ratings for a learned policy in spoken dialogue systems (normally, objective metrics show an improvement, but not subjec-

Feedback questions	Jargon-adapt	Learned-DS
Q1. Quality of voice	3.11	3.36
Q2. Had to ask too many questions	2.23	1.89
Q3. System adapted very well	3.41	3.58
Q4. Easy to identify objects	2.94	3.37 *
Q5. Right amount of dialogue time	3.23	3.26
Q6. Learned useful terms	2.94	3.05
Q7. Conversation was easy	3.17	3.42
Q8. Future use	3.22	3.47

* Statistical significance ($p < 0.05$).

Table 3: Real user feedback

tive scores (Lemon et al., 2006)).

6.5 Analysis

The results show that the Learned-DS strategy is significantly better than the hand-coded Jargon-Adapt policy in terms of adaptation accuracy, dialogue time, and task completion rate. The initial knowledge of the users (mean pre-task recognition score) of the two groups were not significantly different from each other (Jargon-adapt = 7.33, Learned-DS = 7.45). Hence there is no bias on the user's pre-task score towards any strategy. While the Learned-DS system adapts well to its users globally, the Jargon-adapt system adapted only locally. This led to higher task completion rate and lower dialogue time.

The Learned-DS strategy enabled the system to adapt using the dependencies that it learned during the training phase. For instance, when the user asked for clarification on some referring expressions (e.g. "ethernet cable"), it used descriptive expressions for domain objects like ethernet light and ethernet socket. Such adaptation across referents enabled the Learned-DS strategy to score better than the Jargon-adapt strategy. Since the agent starts the conversation with no knowledge about the user, it learned to use information seeking moves (use jargon) at appropriate moments, although they may be inappropriate. But since it was trained to maximize the adaptation accuracy, the agent also learned to restrict such moves and start predicting the user's domain knowledge as soon as possible. By learning to trade-off between information-seeking and adaptation, the Learned-DS policy produced a higher adaptation with real users with different domain knowledge levels.

The users however did not generally rate the two policies differently. However, they did rate it (significantly) easier to identify objects when using the learned policy. For the other ratings, users seemed to be not able to recognize the nuances in the way the system adapted to them. They could have been satisfied with the fact that the system adapted better (Q3). This adaptation and the fact that the system offered help when the users were confused in interpreting the technical terms, could have led the users to score the system well in terms of future use (Q8), dialogue time (Q5), and ease of conversation (Q7), but in common with experiments in dialogue management (Lemon et al., 2006) it seems that users find it difficult to evaluate these improvements subjectively. The users were given only one of the two strategies and therefore were not in a position to compare the two strategies and judge which one is better. Results in table 3 lead us to conclude that perhaps users need to compare two or more strategies in order to judge the strategies better.

7 Conclusion

We presented new results from an evaluation with real users. In this study, we have shown that user-adaptive REG policies can be learned using an RL framework and data-driven user simulations. It learned to trade off between adaptive moves and information seeking moves automatically to maximize the overall adaptation accuracy. The learned policy started the conversation with information seeking moves, learned a little about the user, and started adapting dynamically as the conversation progressed. We also showed that the learned policy performs better than a reasonable hand-coded policy with real users in terms of accuracy of adaptation, dialogue time, task completion, and a subjective evaluation. Finally, this paper provides further evidence that evaluation results obtained

in simulated environments can transfer reliably to evaluations with real users (Lemon et al., 2006).

Whether the learned policy would perform better than a hand-coded policy which was painstakingly crafted by a domain expert (or learned using supervised methods from an expert-layperson corpus) is an interesting question that needs further exploration. Also, it would also be interesting to make the learned policy account for the user's learning behaviour and adapt accordingly. We also believe that this framework can be extended to include other decisions in NLG besides REG (Dethlefs and Cuayahuitl, 2010).

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSiC project `www.classic-project.org`) and from the EPSRC, project no. EP/G069840/1.

References

H. Ai and D. Litman. 2007. Knowledge consistent user simulations for dialog systems. In *Proceedings of Interspeech 2007, Antwerp, Belgium*.

T. Akiba and H. Tanaka. 1994. A Bayesian approach for User Modelling in Dialogue Systems. In *Proceedings of the 15th conference on Computational Linguistics - Volume 2, Kyoto*.

A. Bell. 1984. Language style as audience design. *Language in Society*, 13(2):145–204.

A. Cawsey. 1993. User Modelling in Interactive Explanations. *User Modeling and User-Adapted Interaction*, 3(3):221–247.

H. H. Clark and G. L. Murphy. 1982. Audience design in meaning and reference. In J. F. LeNy and W. Kintsch, editors, *Language and comprehension*. Amsterdam: North-Holland.

N. Dethlefs and H. Cuayahuitl. 2010. Hierarchical Reinforcement Learning for Adaptive Text Generation. In *Proc. INLG 2010*.

K. Georgila, J. Henderson, and O. Lemon. 2005. Learning User Simulations for Information State Update Dialogue Systems. In *Proc of Eurospeech/Interspeech*.

F. Hernandez, E. Gaudioso, and J. G. Boticario. 2003. A Multiagent Approach to Obtain Open and Flexible User Models in Adaptive Learning Communities. In *User Modeling 2003*, volume 2702/2003 of *LNCS*. Springer, Berlin / Heidelberg.

E. A. Issacs and H. H. Clark. 1987. References in conversations between experts and novices. *Journal of Experimental Psychology: General*, 116:26–37.

S. Janarthanam and O. Lemon. 2009a. A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies. In *Proc. SigDial'09*.

S. Janarthanam and O. Lemon. 2009b. A Wizard-of-Oz environment to study Referring Expression Generation in a Situated Spoken Dialogue Task. In *Proc. ENLG'09*.

S. Janarthanam and O. Lemon. 2009c. Learning Lexical Alignment Policies for Generating Referring Expressions for Spoken Dialogue Systems. In *Proc. ENLG'09*.

S. Janarthanam and O. Lemon. 2010. Learning to Adapt to Unknown Users: Referring Expression Generation in Spoken Dialogue Systems. In *Proc. ACL'10*.

O. Lemon, Georgila. K., and J. Henderson. 2006. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Evaluation. In *IEEE/ACL Spoken Language Technology*.

O. Lemon. 2010. Learning what to say and how to say it: joint optimization of spoken dialogue management and Natural Language Generation. *Computer Speech and Language*. (to appear).

E. Levin, R. Pieraccini, and W. Eckert. 1997. Learning Dialogue Strategies within the Markov Decision Process Framework. In *Proc. of ASRU97*.

V. Rieser and O. Lemon. 2009. Natural Language Generation as Planning Under Uncertainty for Spoken Dialogue Systems. In *Proc. EACL'09*.

V. Rieser and O. Lemon. 2010. Optimising information presentation for spoken dialogue systems. In *Proc. ACL*. (to appear).

J. Schatzmann, K. Weilhammer, M. N. Stuttle, and S. J. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement Learning of Dialogue Management Strategies. *Knowledge Engineering Review*, pages 97–126.

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. J. Young. 2007. Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc of HLT/NAACL 2007*.

D. Shapiro and P. Langley. 2002. Separating skills from preference: Using learning to program by reward. In *Proc. ICML-02*.

R. Sutton and A. Barto. 1998. *Reinforcement Learning*. MIT Press.

A unified account of the semantics of discourse particles

Markus Egg
Humboldt-Universität Berlin
markus.egg@anglistik.hu-berlin.de

Abstract

The paper investigates discourse particles on the example of German *doch*, assigning to them very specific semantic interpretations that still cover a wide range of their uses.

The analysis highlights the role of discourse particles in managing the *common ground* and crucially takes into account that discourse particles can refer not only to utterances they are a part of and to previously uttered utterances, but also to *felicity conditions* of these utterances.

1 Introduction

This paper advocates very specific semantic interpretations for discourse particles, concentrating on German *doch*. There is a very wide range of concrete usages of discourse particles in context (which has motivated analysing them as polysemous, e.g., in Helbig (1988)).

Assigning them a uniform semantic interpretation seems to be subject to two conflicting requirements:

- the interpretation must be sufficiently *specific* to allow deriving the interpretation of concrete uses
- it must be sufficiently *general* to cover a wide range of concrete uses

So far, research on the interpretation of *doch* focusses on the second requirement (e.g., Thurmair (1989), König (1997), Karagjosova (2004), or König and Requardt (1997)).

The meaning of *doch* emerges as a two-place relation between the utterance *doch* is a part of and a previous utterance to which the *doch*-utterance is a reaction.

This relation is described by the features KNOWN and CORRECTION in Thurmair (1989), i.e., *doch*-utterances *correct* a previous utterance by introducing *old* information. Karagjosova (2004) regards *doch*-utterances act as *reminders*, which present old information to hearers. According to König and Requardt (1997), *doch*-utterances point out *inconsistencies* between old information and a new piece of information or action.

Such general descriptions of *doch* apply to cases like Karagjosova's (1): B reminds A of Peter's illness, which seems inconsistent with A's announcement and therefore can act as a correction of A:

(1) A: *Peter wird auch mitkommen.* B: *Er ist doch krank.*
'A: Peter will come along, too. B: But he is ill.'

While these general descriptions (excepting Karagjosova (2004)) do not spell out in detail the way in which *doch* contributes to the meaning of larger discourses, they can capture a wide range of uses of these particles.

There remain a number of problematic cases, including *discourse-initial* uses of *doch*-utterances like König and Requardt's (2), which functions as an opening line in a conversation, it neither corrects nor reminds the hearer, nor is there an inconsistency between the utterance and the context:

(2) *Sie sind doch Paul Meier.*
'You must be Paul Meier.'

The proposed analysis of the particle *doch* is sufficiently general to account for a wide range of uses yet being specific enough to specify the semantic construction for discourses that comprise *doch*.

I follow much previous work in developing my analysis on the basis of simple examples like (1),

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 132–138,
The University of Tokyo, September 24-25, 2010.

and then extending it to cases like (2). Most examples consist of two utterances, the second utterance comprises a discourse particle and is a reaction to the first one. In the remainder of this paper, these two utterances are called 'involved utterances'.

In (1), the (propositional) semantic arguments of the particle are the meanings of these two utterances. But the semantic arguments of a discourse particle may differ from the meanings of the involved utterances, as illustrated by (3) (from Thurmair 1989):

(3) A: *Seit wann hast du denn den „Zauberberg"?* B: *Den hast du mir doch vor zwei Jahren geschenkt.*
A: 'Since when have you owned the 'Zauberberg'? B: But you gave it to me two years ago.'

B reacts to the implicit statement that A does not know the answer to his question. This statement is an argument of *doch* in (3), even though it is not the meaning of A's utterance. This shows that the semantic arguments of discourse particles must be distinguished from the meanings of their involved utterances.

Utterances with a discourse particle and preceding utterances to which they react are called 'p(article)-utterances' (or '*doch*-utterances') and 'a(ntecedent)-utterances'. They are distinguished from the semantic arguments of the particle, which are referred to as 'p-proposition' and 'a-proposition', respectively.

This is not just a question of nomenclature but reflects a fundamentally different view on the role of discourse particles. Instead of indicating the relation between two already identified propositions, the meaning of the particle applied to its first argument (very often but not always the interpretation of the p-utterance) determines the range of potential a-propositions in the context of utterance. From this range, the hearer selects the appropriate proposition.

This resembles the intuition of König and Requardt (1997) that discourse particles are 'metapragmatic instructions' which tell hearers how to deal with the p-utterance in a communicative situation.

Consequently, a- and p-utterances do not determine the semantic arguments for all uses of discourse particles, which might account for some problems of defining the semantics of the particles in the literature, which is characteristically based on the meanings of a- and p-utterance.

My claim is that there is a link between a- and p-proposition and a- and p-utterance, respectively, in that the propositions can either be the meanings of the utterances or emerge through the *felicity conditions* of the utterances.

E.g., in (3) the *doch*-proposition reminds A of the fact that the first preparatory condition for a question (that the speaker does not know the answer) does not hold, since A (as the one who gave the book to B) should know since when the book has been in B's possession.

The plan of the paper is to introduce background assumptions on discourse particles in section 2, to apply the proposed approach to the (unstressed) particle *doch* in section 3, and to conclude with an outlook on further research.

2 Formal background

This paper follows much previous work in assuming that discourse particles refer to the *common ground* (CG), e.g., König (1997), Karagjosova (2004), or Zimmermann (2009).

Common ground and the interlocutors' individual backgrounds are modelled as common or individual *belief* (Stalnaker, 2002). Individual belief is equated with the set of propositions that are true in all possible worlds compatible with the individual's beliefs; common belief, with the set of propositions believed by all members of the respective group of believers.

Stalnaker notes that this is an idealisation in that the CG might comprise propositions not shared by the background of every member of the group. But this idealisation is not a problem for the analysis presented in this paper.

Reasoning on CG and individual backgrounds often uses *defeasible deduction* (Asher and Lascarides, 2003). I.e., from statements of the form 'p defeasibly entails q' ($p > q$) together with p one can defeasibly deduce q.

This defeasible Modus Ponens applies if $\neg q$ does not hold and $\neg q$ cannot be deduced simultaneously (Asher and Lascarides, 2003). Defeasible deducability of p from a set of propositions C is written as '$C \mid\!\sim\ p$'.

Reference to the common ground makes the semantics of discourse particles context-dependent, because the CG is relative to the interlocutor(s) of

a- and p-utterances. This shows up in the shifting effects observed in Döring (2010). Consider e.g. what happens if one embeds (1) in a quotation like in (4):

(4) *A sagte, Peter komme auch mit. B entgegnete, er sei doch krank.*
'A said Peter would come along, too. B retorted that he was ill.'

The shift in (4) arises because *doch* presents a proposition (here, that Peter is ill) as part of the common ground, and the relevant CG is calculated with respect to A and B, not with respect to the interlocutors of (4). I.e., (4) does not express that Peter's illness is in the common ground of the speaker and hearer of (4).

3 The analysis

The proposed approach to discourse particle is now applied to *doch*, which introduces a notion of *tension* between the a- and the p-proposition.

3.1 Declarative a- and p-utterances

I will first illustrate this notion with simple examples in which the a-utterance expresses the a-proposition, and the meaning of the declarative p-utterance provides the p-proposition.

In (5) [= (1)] and (6), adapted from Karagjosova (2004), there is tension between being ill on the one hand and going out and living healthily on the other hand, respectively:

(5) A: *Peter wird auch mitkommen.* B: *Er ist doch krank.*
'A: Peter will come along, too. B: But he is ill.'

(6) *Ich bin oft krank. Dabei lebe ich doch gesund.*
'I am often ill. But I have a healthy lifestyle.'

The intuitive notion of tension between two propositions p and q is formalised as defeasible entailment $q > \neg p$. I.e., given q, one would expect p, but the propositions are not incompatible, even though q is a potential impediment for p.

The effect of *doch* q as a reaction to an a-proposition p against the common ground C is to remind the hearer that C comprises a potential impediment q for p, which expresses either surprise at the fact that p nevertheless holds or puts doubt on p. Still, p is not explicitly denied.

Formally, *doch* states that the common ground C defeasibly entails q and the fact that q defeasibly entails $\neg p$ (which by defeasible Modus Ponens would allow one to infer $\neg p$, if the conditions for defeasible Modus Ponens are met):

(7) $[\![\text{doch}]\!](q)(p)$ iff $C \mid\!\sim\ q \wedge C \mid\!\sim\ q > \neg p$

This analysis differs from the one of König (1997), who assumes that *doch* q points out a contradiction in the CG, in that p is incompatible with a consequence of q. In contrast, I regard this incompatibility as a *default* only. The status of q as derivable from the CG is also expressed in Karagjosova (2004) claim that *doch* introduces q as a reminder and in Thurmair's (1989) feature KNOWN.

In (5) and (6), p and q are the semantics of the a- and the *doch*-utterance. Being ill is a potential impediment for going out, so, by pointing out Peter's illness in (5), B expresses surprise or disbelief at A's announcement but does not necessarily correct it or refute it, because even ill people can go out in principle.

Similarly, the speaker of (6) is surprised at his frequent illness, in spite of his healthy lifestyle. (6) shows that q is only a default impediment: If q and p were contradictory, (6) would be nonsensical, but, intuitively, it is not.

The use of defeasible implications to model the tension between two propositions as indicated by *doch* is closely related to accounts of the discourse relation of CONCESSION in Grote et al. (1997), Oversteegen (1997), Lagerwerf (1998), and Knott (1996).

They assume the same kind of defeasible implication for this discourse relation and model it as a presupposition, which is compatible with giving it common ground status.

3.2 Non-declarative a-utterances

In (5) and (6), the a-proposition enters the CG as the meaning of an a-utterance. But the a-proposition can also emerge as a *felicity condition* of the a-utterance. Consider e.g. *doch*-utterances as reactions to questions, as in (8) [= (3)]:

(8) A: *Seit wann hast du denn den „Zauberberg"?* B: *Den hast du mir doch vor zwei Jahren geschenkt.*
A: 'Since when have you owned the 'Zauberberg'? B: But you gave it to me two years ago.'

Doch in (8) expresses surprise at the question being asked, since A himself gave the book to B and hence should know that B owns it.

The proposed analysis reconstructs this intuition: B's utterance expresses a proposition q (that A gave the book to B) and points out that q is part of the CG. It is also part of the CG that q is a potential obstacle for a specific a-proposition p (formally, the CG entails $q > \neg p$).

Such p-utterances restrict the range of potential a-propositions p, and their hearers try to identify the a-propositions in the given context. The a-utterance in (8), however, cannot directly contribute p in any context, since its meaning is not a proposition.

But due to the assumption that A is cooperative, the question introduces into the CG the felicity conditions for questions, among them the first preparatory condition, viz., that A does not know the answer to his question. This is a suitable p, because it is reasonable to assume that if A gave the book to B ($= q$), he should know the answer to the question ($= \neg p$).

The intuition that the semantic arguments of discourse particles need not be identical to the meanings of the involved utterances is related to suggestions to let discourse relations relate either to the content of the discourse segments that they link or to the corresponding intensions of the speaker or the intended effencts on the hearer, which is suggested by Sweetser (1990) and Knott (2001).

Doch-utterances in reaction to imperatives work analogously, e.g., (9):

(9) A: *Übersetze mir bitte diesen Brief.* B: *Ich kann doch kein Baskisch.*
A: 'Please translate this letter for me.' B: 'But I do not know Basque'.

Here B's lack of proficiency in Basque ($= q$) and A's belief that B can translate a Basque letter (i.e., the first preparatory condition of the request, our p) are in tension.

Now q can be deduced from the common ground either because it has been explicitly introduced before or because it makes sense to assume defeasibly that someone does not speak a less known language like Basque. In either case, A should not take for granted that B speaks Basque, i.e., has a reason not to require B to translate letters written in Basque.

This use of *doch* also shows up in reactions to declarative statements: The p-utterance of (10) states no potential impediment for the proposition expressed by A.

Rather, B's use of *doch* refers to A's surprise, suggesting that A should not be surprised at all. The felicity condition of expressing surprise that is cast into doubt by B is considering the fact about which one is surprised as something extraordinary, which would not have happened in a normal course of events.

(10) A: *Peter sieht schlecht aus.* B: *Er war doch lange im Krankenhaus.*
'A: Peter does not look healthy. B: But he has been in hospital for a long time.'

Peter's long stay in the hospital ($= q$) is no potential obstacle to looking unhealthy, on the contrary, it entails defeasibly that his looking unhealthy is quite normal ($= \neg p$). This would negate the abovementioned felicity condition for A's surprise ($= p$), hence suggests that A should not be surprised.

3.3 Non-declarative p-utterances

Another group of *doch*-utterances are imperative or interrogative (the latter adapted from Thurmair (1989)):

(11) *Verklag mich doch!*
'Go ahead and sue me.'

(12) *Komm doch nach Hause!*
'Do come home.'

(13) *Wie heißt doch diese Kneipe in der Sredzkistraße?*
'What is the name of this pub in the Sredzkistraße?'

(14) *Wie sagt Goethe doch so treffend?*
'What was this piercing remark of Goethe again?'

(15) *Du kommst doch?*
'I presume that you will be there.'

Doch is used provocatively in imperatives like (11); it suggests that the hearer cannot fulfil the request. In cases like (12), *doch* signals that the requested or suggested action is a very natural thing to do. *Doch*-questions refer to a piece of knowledge that the speaker knows or is supposed to

know (Thurmair, 1989), e.g., (13) indicates that the speaker knows the answer at least in principle, (14) announces a quotation, and (15) suggests that the answer can only be affirmative.

There are two issues in interpreting these sentences; the p-utterance does not denote a proposition (which could be the semantic argument of *doch*), and there need not be an a-utterance at all from which to derive the a-proposition.

But in all these utterances, speakers use *doch* to point out that they are aware of evidence from the CG which suggests that a felicity condition of the *utterance itself* does not hold. This can be modelled by identifying the p-proposition q (the argument of *doch*) with the fact that the sentences were uttered, which can be (trivially) deduced from the common ground C (the condition $C \mid\sim q$ in (7)).

Then the felicity conditions associated with different kinds of illocutionary acts emerge from the common ground C as default entailments from the utterance of the respective illocutionary type (the condition $C \mid\sim q > \neg p$ in (7); here $\neg p$ refers to one of the felicity conditions).

I.e., using *doch* in these cases triggers a search for a suitable a-proposition p in the CG which negates a felicity condition of the utterance. E.g., *doch* in (11) shows that the first preparatory condition of a request (the speaker believes that the hearer can do it) does not hold, even though this condition follows defeasibly from the fact that the request was made.

In (12), *doch* addresses the second preparatory condition of a request or advice (that it is not obvious to speaker and hearer that the hearer complies with the request in a normal course of events). Thus, *doch* suggests that it is obvious that the hearer will do anyway what is requested or advised, even though uttering (12) defeasibly entails the contrary. Consequently, (12) presents a request or advice as a very natural thing to do.

I.e., *doch*-imperatives do not correct unwanted behaviour by the hearer (pace Thurmair (1989)), which is confirmed by examples like (16), which can be uttered between future lovers during their courtship to take the process of courting one step further:

(16) *Komm doch mal vorbei!*
'Just drop by.'

(16) does not request the hearer to change his behaviour, because calling on the speaker was not an option yet. Instead, visiting the speaker is presented as a very natural thing to do for the hearer, i.e., once more the second preparatory condition of a request does not hold.

Using *doch* in questions also indicates that a felicity condition of the utterance does not hold, even though its validity could be deduced defeasibly from the fact that the question has been asked. The relevant condition is the first preparatory condition for questions (that the speaker does not know the answer already).

Doch signals that this condition is not fulfilled, either because the answer escapes the speaker only momentarily, as in (13), because he obviously knows, as in the conventionalised announcement (14), or because he would not accept a refusal, which settles the question, like in (15).

The analysis predicts that *doch* is not acceptable in ordinary questions, which is borne out e.g. by (17), because in these questions there is no tension between uttering the question and potential obstacles for its felicity conditions:

(17) **Wer schreibt dir doch?*
'But who is corresponding with you?'

Rhetorical questions are also incompatible with *doch*, but for a different reason. Consider e.g. the contrasting dialogue pairs (18a)/(18b) and (18a)/(18c):

(18) (a) A: *Ich werde meinen 30. Geburtstag mit einem großen Fest feiern.*
A: 'I'll throw a big party on the occasion of my 30th birthday.'
(b) B: *Es würde doch keiner zu deinem Fest kommen.*
B: 'But no one would come to your party.'
(c) B: **Wer würde doch zu deinem Fest kommen?*
B: 'But who would come to your party?'

The inacceptability of (18a)/(18c) - and of the rhetorical *doch*-question in particular - is not due to the function of the rhetorical question as a negated statement: In this case, (18b) should be an inacceptable response to (18a), too.

(18c) is inacceptable because rhetorical questions characterise statements as CG information (Egg, 2007). This is also one of the effects of *doch*; consequently, (18c) is as informative as

(18b) but more complex, hence, its use would not comply to conversation maxims (Grice, 1975; Krifka, 1989).

To sum up, non-declarative *doch*-utterances refer to their own felicity conditions; since they do not denote propositions, the first semantic argument of *doch* cannot be the meaning of the *doch*-utterance.

Instead, *doch* applies to the fact that the speaker uttered the sentence. In contrast, declarative *doch*-utterances like in (8) or (9) refer to a felicity condition of the non-declarative a-utterance.

This analysis of non-declarative *doch*-utterances also applies to the hitherto extremely problematic group of *discourse-initial doch*-utterances:

(19) *Morgen fahre ich doch nach Wien.*
'Well, I'll go to Vienna tomorrow.'

(20) *Du hast doch ein Auto.*
'Well, you have a car.'

These examples are characterised by *doch* as a *reminder*. This means that the p-utterance (the speaker's travel plans or the fact that the hearer has a car) contributes information semantically that is already in the CG. However, this information is not obviously in tension to any other information. This raises the question of what the semantic arguments of *doch* are in these cases.

Here *doch* addresses the first preparatory condition for statements, viz., that it is not obvious to the speaker that the hearer already knows what the speaker will say. Uttering the statement ($= q$) defeasibly implies this condition ($= \neg p$), but according to the CG the speaker knows that the hearer knows ($= p$).

(21) [= (2)] instantiates this case, too:

(21) *Sie sind doch Paul Meier.*
'You must be Paul Meier.'

Telling someone his name obviously violates the first preparatory condition for statements, whence the use of *doch*.

Another such case is the use of *doch* in expressions of outrage. Here *doch* signals that it is common knowledge that the hearer knows that the situation or action to which the speaker refers is outrageous:

(22) *Das ist doch die Höhe!*
'That is the limit!'

Finally, the sincerity condition of a statement can also be targeted by *doch*:

(23) *Da sagt er doch im letzten Moment ab!*
'I can't believe that he cancelled the appointment at the last moment.'

In (23), *doch* expresses disbelief of the speaker, he cannot believe what he is saying. This violates the sincerity condition for statements. The effect of *doch* here is one of expressing surprise.

The same effect shows up in exclamative *wh*-sentences:

(24) *Wie schön Amélie doch ist!*
'How beautiful Amélie is!'

Following analyses of these sentences like Zanuttini and Portner (2003) or Rett (2009), (24) characterises the degree of Amélie's beauty as unexpectedly or surprisingly high. Hence, *doch* naturally occurs in these exclamatives to deny a belief of the speaker in what he is stating.

In sum, I offered a uniform semantic analysis of *doch*, which still covers a wide range of its usages. *Doch* relates two propositions p and q iff q is derivable from the common ground as well as the fact that q defeasibly implies $\neg p$, i.e., q presents a potential impediment for p.

The correlation of p and q with utterances is flexible, however: Often q is the meaning of the *doch*-utterance, but for non-declarative and discourse-initial declarative *doch*-utterances, q is the fact that this utterance has been made.

The proposition p can be the meaning of a preceding a-utterance to which the *doch*-utterance is a reaction. But especially for non-declarative a-utterances, p can also be one of its felicity conditions, or, for discourse-initial *doch*-utterances, a felicity condition of the utterance itself.

4 Conclusion and outlook

The paper outlined a research programme for discourse particles that captures their meanings in very specific semantic descriptions that nevertheless account for the wide range of their uses. These two competing goals can be pursued simultaneously because *doch*-utterances can be integrated flexibly into the meaning of the surrounding discourse.

While discourse particles like *doch* uniformly relate two propositions semantically, the meaning of the utterance of which the particle is a part, and

the meaning of the utterance to which this first utterance reacts are not the only feasible semantic arguments of the particles: They can also have *felicity conditions* of these two utterances as semantic arguments.

This research programme was illustrated by investigating the particle *doch*. The next steps now are to extend the coverage of this analysis to other particles, in particular, *schon*, and to contrast 'minimal pairs' of discourses which differ only by discourse particles (e.g., *Komm schon!* as opposed to *Komm doch!*, which both require the hearer to come).

This analysis can also be used for investigations of stressed and unstressed forms of discourse particles and of the relation between them. Here it is particularly interesting to take prosody seriously and to look into the semantic effects of emphasising a discourse particle.

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of conversation*. Cambridge University Press, Cambridge.

Sophia Döring. 2010. On context shift in German discourse particles. BA thesis, Humboldt-Universität Berlin.

Markus Egg. 2007. Meaning and use of rhetorical questions. In Maria Aloni, Paul Dekker, and Floris Roelofsen, editors, *Proceedings of the 16th Amsterdam Colloquium*, pages 73–78. Universiteit van Amsterdam, ILLC.

Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry Morgan, editors, *Syntax and semantics 3: Speech acts*, pages 41–58. Academic Press, New York.

Brigitte Grote, Nils Lenke, and Manfred Stede. 1997. Ma(r)king concessions in English and German. *Discourse Processes*, 24:87–118.

Gerhard Helbig. 1988. *Lexikon deutscher Partikeln*. Verlag Enzyklopädie, Leipzig.

Elena Karagjosova. 2004. *The meaning and function of German modal particles*. Ph.D. thesis, Universität des Saarlandes.

Alistair Knott. 1996. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. Ph.D. thesis, University of Edinburgh.

Alistair Knott. 2001. Semantic and pragmatic relations and their intended effects. In T. Sanders, J. Schilperoord, and W. Spooren, editors, *Text representation: linguistic and psycholinguistic aspects*, pages 127–151. Benjamins, Amsterdam.

Ekkehard König and Susanne Requardt. 1997. A relevance-theoretic approach to the analysis of modal particles. *Multilingua*, 10:63–77.

Ekkehard König. 1997. Zur Bedeutung von Modalpartikeln im Deutschen: Ein Neuansatz im Rahmen der Relevanztheorie. *Germanistische Linguistik*, 136:57–75.

Manfred Krifka. 1989. *Nominalreferenz und Zeitkonstitution*. Fink, München.

Luuk Lagerwerf. 1998. *Causal connectives have presuppositions. Effects on coherence and discourse structure*. Ph.D. thesis, Universiteit van Tilburg.

Leonoor Oversteegen. 1997. On the pragmatic nature of causal and contrastive connectives. *Discourse Processes*, 24:51–85.

Jessica Rett. 2009. A degree account of exclamatives. In *Proceedings of SALT XVIII*, pages 601–608. CLC Publications.

Robert Stalnaker. 2002. Common ground. *Linguistics & Philosophy*, 25:701–721.

Eve Sweetser. 1990. *From etymology to pragmatics. Metaphorical and cultural aspects of semantic structure*. Cambridge University Press, Cambridge.

Maria Thurmair. 1989. *Modalpartikeln und ihre Kombinationen*. Niemeyer, Tübingen.

Raffaela Zanuttini and Paul Portner. 2003. Exclamative clauses: At the syntax-semantics interface. *Language*, 79:39–81.

Malte Zimmermann. 2009. Discourse particles. In Claudia Maienborn, Klaus von Heusinger, and Paul Portner, editors, *Semantics: an international handbook of natural language meaning*. Mouton de Gruyter, Berlin. In press.

The Effects of Discourse Connectives Prediction on Implicit Discourse Relation Recognition

Zhi Min Zhou[†], **Man Lan**[†,§], **Zheng Yu Niu**[‡], **Yu Xu**[†], **Jian Su**[§]
[†]East China Normal University, Shanghai, PRC.
[‡]Baidu.com Inc., Beijing, PRC.
[§]Institute for Infocomm Research, Singapore.
51091201052@ecnu.cn, lanman.sg@gmail.com

Abstract

Implicit discourse relation recognition is difficult due to the absence of explicit discourse connectives between arbitrary spans of text. In this paper, we use language models to predict the discourse connectives between the arguments pair. We present two methods to apply the predicted connectives to implicit discourse relation recognition. One is to use the sense frequency of the specific connectives in a supervised framework. The other is to directly use the presence of the predicted connectives in an unsupervised way. Results on PDTB2 show that using language model to predict the connectives can achieve comparable F-scores to the previous state-of-art method. Our method is quite promising in that not only it has a very small number of features but also once a language model based on other resources is trained it can be more adaptive to other languages and domains.

1 Introduction

Discourse relation analysis involves identifying the discourse relations (e.g., the comparison relation) between arbitrary spans of text, where the discourse connectives (e.g., "*however*", "*because*") may or may not explicitly exist in the text. This analysis is one important application both as an end in itself and as an intermediate step in various downstream NLP applications, such as text summarization, question answering etc.

As discussed in (Pitler and Nenkova., 2009b), although explicit discourse connectives may have two types of ambiguity, i.e., one is discourse or non-discourse usage ("*once*" can be either a temporal connective or a word meaning "formerly"), the other is discourse relation sense ambiguity ("*since*" can serve as either a temporal or causal connective), their study shows that for explicit discourse relations in Penn Discourse Treebank (PDTB) corpus, the most general 4 senses, i.e., Comparison (Comp.), Contingency (Cont.), Temporal (Temp.) and Expansion (Exp.), can be easily addressed by the presence of discourse connectives and a simple method only considering the sense frequency of connectives can achieve more than 93% accuracy. This indicates the importance of connectives for discourse relation recognition.

However, with implicit discourse relation recognition, there is no connective between the textual arguments, which results in a very difficult task. In recent years, a multitude of efforts have been employed to solve this task. One approach is to exploit various linguistically informed features extracted from human-annotated corpora in a supervised framework (Pitler et al., 2009a) and (Lin et al., 2009). Another approach is to perform recognition without human-annotated corpora by creating synthetic examples of implicit relations in an unsupervised way (Marcu and Echihabi, 2002).

Moreover, our initial study on PDTB implicit relation data shows that the averaged F-score for the most general 4 senses can reach 91.8% when we obtain the sense of test examples by mapping each implicit connective to its most frequent sense (i.e., sense recognition using gold-truth implicit connectives). This high F-score performance again proves that the connectives are very crucial source for implicit relation recognition.

In this paper, we present a new method to address the problem of recognizing implicit discourse relation. This method is inspired by the above observations, especially the two gold-truth results, which reveals that discourse connectives are very important signals for discourse relation recognition. Our basic idea is to recover the implicit connectives (not present in real text) between two spans of text with the use of a language

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 139–146,
The University of Tokyo, September 24-25, 2010.

model trained on large amount of raw data without any human-annotation. Then we use these predicted connectives to generate feature vectors in two ways for implicit discourse relation recognition. One is to use the sense frequency of the specific connectives in a supervised framework. The other is to directly use the presence of the predicted connectives in an unsupervised way.

We performed evaluation on explicit and implicit relation data sets in the PDTB 2 corpus. Experimental results showed that the two methods achieved comparable F-scores to the state-of-art methods. It indicates that the method using language model to predict connectives is very useful in solving this task.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 describes our methods for implicit discourse relation recognition. Section 4 presents experiments and results. Section 5 offers some conclusions.

2 Related Work

Existing works on automatic recognition of implicit discourse relations fall into two categories according to whether the method is supervised or unsupervised.

Some works perform relation recognition with supervised methods on human-annotated corpora, for example, the RST Bank (Carlson et al., 2001) used by (Soricut and Marcu, 2003), adhoc annotations used by (Girju, 2003) and (Baldridge and Lascarides, 2005), and the GraphBank (Wolf et al., 2005) used by (Wellner et al., 2006).

Recently the release of the Penn Discourse TreeBank (PDTB) (Prasad et al., 2006) has significantly expanded the discourse-annotated corpora available to researchers, using a comprehensive scheme for both implicit and explicit relations. (Pitler et al., 2009a) performed implicit relation classification on the second version of the PDTB. They used several linguistically informed features, such as word polarity, verb classes, and word pairs, showing performance increases over a random classification baseline. (Lin et al., 2009) presented an implicit discourse relation classifier in PDTB with the use of contextual relations, constituent Parse Features, dependency parse features and cross-argument word pairs. Although both of two methods achieved the state of the art performance for automatical recognition of implicit discourse relations, due to lack of human-annotated corpora, their approaches are not very useful in the real word.

Another line of research is to use the unsupervised methods on unhuman-annotated corpus.

(Marcu and Echihabi, 2002) used several patterns to extract instances of discourse relations such as contrast and elaboration from unlabeled corpora. Then they used word-pairs between arguments as features for building classification models and tested their model on artificial data for implicit relations.

Subsequently other studies attempt to extend the work of (Marcu and Echihabi, 2002). (Sporleder and Lascarides, 2008) discovered that Marcu and Echihabi's models do not perform as well on implicit relations as one might expect from the test accuracy on synthetic data. (Goldensohn, 2007) extended the work of (Marcu and Echihabi, 2002) by refining the training and classification process using parameter optimization, topic segmentation and syntactic parsing. (Saito et al., 2006) followed the method of (Marcu and Echihabi, 2002) and conducted experiments with a combination of cross-argument word pairs and phrasal patterns as features to recognize implicit relations between adjacent sentences in a Japanese corpus.

Previous work showed that with the use of some patterns, structures, or the pairs of words, relation classification can be performed using unsupervised methods.

In contrast to existing work, we investigated a new knowledge source, i.e., implicit connectives predicted using a language model, for implicit relation recognition. Moreover, this method can be applied in both supervised and unsupervised ways by generating features on labeled and unlabeled training data and then performing implicit discourse connectives recognition.

3 Methodology

3.1 Predicting implicit connectives via a language model

Previous work (Pitler and Nenkova., 2009b) showed that with the presence of discourse connectives, explicit discourse relations in PDTB can be easily identified with more than 90% F-score. Our initial study on PDTB human-annotated implicit relation data shows that the averaged F-score for the most general 4 senses can reach 91.8% when we simply map each implicit connective to

its most frequent sense. These high F-scores indicate that the connectives are very crucial source of information for both explicit and implicit relation recognition. However, for implicit relations, there are no explicitly discourse connectives in real text. This built-in absence makes the implicit relation recognition task quite difficult. In this work we overcome this difficulty by inserting connectives into the two arguments with the use of a language model.

Following the annotation scheme of PDTB, we assume that each implicit connective takes two arguments, denoted as *Arg1* and *Arg2*. Typically, there are two possible positions for most of implicit connectives, i.e., the position before *Arg1* and the position between *Arg1* and *Arg2*. Given a set of implicit connectives $\{c_i\}$, we generate two synthetic sentences, c_i+*Arg1*+*Arg2* and *Arg1*+c_i+*Arg2* for each c_i, denoted as $S_{c_i,1}$ and $S_{c_i,2}$. Then we calculate the *perplexity* (an intrinsic score) of these sentences with the use of a language model, denoted as $Ppl(S_{c_i,j})$. According to the value of $Ppl(S_{c_i,j})$ (the lower the better), we can rank these sentences and select the connectives in top N sentences as implicit connectives for this argument pair. Here the language model may be trained on any large amount of unannotated corpora that can be cheaply acquired. Typically, a large corpora with the same domain as the test data will be used for training language model. Therefore, we chose news corpora, such as North American News Corpora.

After that, we use the top N predicted connectives to generate different feature vectors and perform the classification in two ways. One is to use the sense frequency of predicted connectives in a supervised framework. The other is to directly use the presence of the predicted connectives in an unsupervised way. The two approaches are described as follows.

3.2 Using sense frequency of predicted discourse connectives as features

After the above procedure, we get a sorted set of predicted discourse connectives. Due to the presence of an implicit connective, the implicit discourse relation recognition task can be addressed with the methods for explicit relation recognition, e.g., sense classification based only on connectives (Pitler et al., 2009a). Inspired by their work, the first approach is to use sense frequency of predicted discourse connectives as features. We take the connective with the lowest perplexity value (i.e., top 1 connective) as the real connective for the arguments pair. Then we count the sense frequency of this connective on the training set. Figure 1 illustrates the procedure of generating predicted discourse connective from a language model and calculating its sense frequency from training data. Here the calculation of sense frequency of connective is based on the annotated training data which has labeled discourse relations, thus this method is a supervised one.

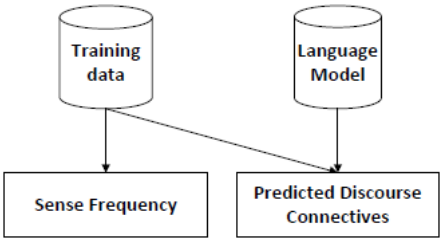


Figure 1: Procedure of generating a predicted discourse connective and its sense frequency from the training set and a language model.

Then we can directly use the sense frequency to generate a 4-feature vector to perform the classification. For example, the sense frequency of the connective *but* in the most general 4 senses can be counted from training set as 691, 6, 49, 2, respectively. For a given pair of arguments, if *but* is predicted as the top 1 connective based on a language model, a 4-dimension feature vector $(691, 6, 49, 2)$ is generated for this pair and used for training and test procedure. Figure 2 and 3 show the training and test procedure for this method.

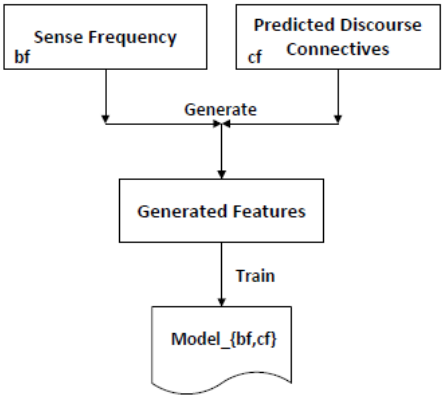


Figure 2: Training procedure for the first approach.

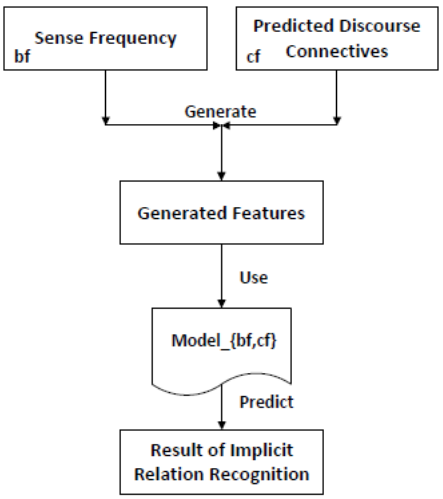


Figure 3: Test procedure for the first approach.

3.3 Using presence or absence of predicted discourse connective as features

(Pitler et al., 2008) showed that most connectives are unambiguous and it is possible to obtain high-accuracy in prediction of discourse senses due to the simple mapping relation between connectives and senses. Given two examples:
(E1) She paid less on her dress, *but* it is very nice.
(E2) We have to harry up *because* the raining is getting heavier and heavier.
The two connectives, i.e., *but* in E1 and *because* in E2, convey the Comparison and Contingency senses respectively. In most cases, we can easily recognize the relation sense by the appearance of a discourse connective since it can be interpreted in only one way. That means the ambiguity of the mapping between sense and connective is quite low. Therefore, the second approach is to use only the presence of the top N predicted discourse connectives to generate a feature vector for a given pair of arguments.

4 Experiment

4.1 Data sets

We used PDTB as our data set to perform the evaluation of our methods. The corpus contains annotations of explicit and implicit discourse relations. The first evaluation is performed on the annotated implicit data set. Following the work of (Pitler et al., 2009a), we used sections 2-20 as the training set, sections 21-22 as the test set and sections 0-1 as the development set for parameter optimization (e.g., *N* value). The second evaluation is performed on the annotated explicit data set. We follow the method used in (Sporleder and Lascarides, 2008) to remove the discourse connective from the explicit instances and consider these processed instances as implicit ones.

We constructed four binary classifiers to recognize each main senses (i.e., Cont., Cont., Exp., Temp.) from the rest. For each sense we used equal numbers of positive and negative instances in training set. The negative instances were chosen at random from the rest of training set. For both evaluations all instances in sections 21-22 were used as test set. Table 1 lists the numbers of positive and negative instances for each sense in training, development and test sets of implicit and explicit relation data sets.

4.2 Evaluation and classifier

To evaluate the performance of above systems, we used two widely-used measures, F-score (i.e., F_1) and accuracy. In addition, in this work we used the LIBSVM toolkit to construct four linear SVM classifiers for each sense.

4.3 Preprocessing

We used the SRILM toolkit to build a language model and calculated the *perplexity* value for each training and test sample. The steps are described as follows. First, since perplexity is an intrinsic score to measure the similarity between training and test samples, in order to fit the restriction of perplexity we chose 3 widely-used corpora in the Newswire domain to train the language model, i.e., (1) the New York part of BLLIP North American News Text (Complete), (2) the Xin and (3) the Ltw parts of the English Gigaword Fourth Edition. For the BLLIP corpus with 1,796,386 automatically parsed English sentences, we converted the parsed sentences into original textual data. Some punctuation marks such as commas, periods, minuses, right/left parentheses are converted into their original form. For the Xin and Ltw parts, we only used the Sentence Detector toolkit in OpenNLP to split each sentence. Finally we constructed 3-, 4- and 5-grams language models from these three corpora. Table 2 lists statistics of different n-grams in the different language models and different corpora.

Next, for each instance we combined its *Arg1* and *Arg2* with connectives obtained from PDTB. There are two types of connectives, single connectives (e.g. "*because*" and "*but*") and paral-

Table 1: Statistics of positive and negative instances for each sense in training, development and test sets of implicit and explicit relation data sets.

	Implicit				Explicit			
	Comp.	Cont.	Exp.	Temp.	Comp.	Cont.	Exp.	Temp.
Train(Pos/Neg)	1927/1927	3375/3375	6052/6052	730/730	4080/4080	2732/2732	4609/4609	2663/2663
Dev(Pos/Neg)	191/997	292/896	651/537	54/1134	438/1071	295/1214	514/995	262/1247
Test(Pos/Neg)	146/912	276/782	556/502	67/991	388/1025	235/1178	501/912	289/1124

Table 2: Statistics of different n-grams in the different language models and different corpora.

n-gram	BLLIP - New York	Gigaword-Xin	Gigaword-Ltw
1-gram	1638156	2068538	2276491
2-grams	26156851	23961796	33504873
3-grams	80876435	77799100	101855639
4-grams	127142452	134410879	159791916
5-grams	146454530	168166195	183794771

lel connectives (such as "*not only ...*, *but also*"). Since discourse connectives may appear not only ahead of the *Arg1*, but also between *Arg1* and *Arg2*, we considered this case. Given a set of possible implicit connectives $\{c_i\}$, for a single connective c_i, we constructed two synthetic sentences, c_i+*Arg1*+*Arg2* and *Arg1*+c_i+*Arg2*. In case of parallel connectives, we constructed one synthetic sentence like $c_{i,1}$+*Arg1*+$c_{i,2}$+*Arg2*.

As a result, we obtain 198 synthetic sentences ($|c_i| * 2$ for single connective or $|c_i|$ for parallel connective) for each pair of arguments. Then we converted all words to lower cases and used the language model trained in the above step to calculate its perplexity (the lower the better) value on sentence level. The sentences were ranked from low to high according to their perplexity scores. For example, given a sentence with arguments pair as follows:
Arg1: it increased its loan-loss reserves by $93 million after reviewing its loan portfolio,
Arg2: before the loan-loss addition it had operating profit of $10 million for the quarter.
we got the perplexity (*Ppl*) values for this arguments pair in combination with two connectives (*but* and *by comparison*) in two positions as follows:

1. *but* + *Arg1* + *Arg2*: *Ppl*= 349.622
2. *Arg1* + *but* + *Arg2*: *Ppl*= 399.339
3. *by comparison* + *Arg1* + *Arg2*: *Ppl*= 472.206
4. *Arg1* + *by comparison* + *Arg2*: *Ppl*= 543.051

In our second approach described in Section 3.3, we considered the combination of connectives and their position as final features like mid_*but*, first_*but*, where the features are binary, that is, the presence or absence of the specific connective. According to the value of $Ppl(S_{c_i,j})$, we tried various N values on development set to get the optimal N value.

4.4 Results

Table 3 summarizes the best performance achieved using gold-truth implicit connectives, the previous state-of-art performance achieved by (Pitler et al., 2009a) and our approaches. The first line shows the result by mapping the gold-truth implicit connectives directly to the relation's sense. The second line presents the best result of (Pitler et al., 2009a). One thing worth mentioning here is that for the Expansion relation, (Pitler et al., 2009a) expanded both training and test sets by including EntRel relation as positive examples, which makes it impossible to perform direct comparison. The third and fourth lines show the best results using our first approach, where the sense frequency is counted on explicit and implicit training set respectively. The last line shows the best result of our second approach only considering the presence of top N connectives.

Table 4 summarizes the best performance using gold-truth explicit connectives reported in (Pitler and Nenkova., 2009b) and our two approaches.

Figure 4 shows the curves of averaged F-scores on implicit connective classification with different n-gram language models. From this figure we can see that all 4-grams language models achieved around 0.5% better averaged F-score than 3-grams models. And except for Ltw corpus, other 5-grams models achieved lower averaged F-score than 4-grams models. Specially the 5-grams result of New York corpus is much lower than its 3-grams result.

Figure 5 shows the averaged F-scores of different top N on the New York corpus with 3-, 4- and 5-grams language models. The essential

Table 3: Best result of implicit relations compared with state-of-art methods.

System	Comp. vs. Not F_1 (Acc)	Cont. vs. Other F_1 (Acc)	Exp. vs. Other F_1 (Acc)	Temp. vs. Other F_1 (Acc)	Averaged F_1 (Acc)
Sense recognition using gold-truth implicit connectives	94.08(98.30)	98.19(99.05)	97.79(97.64)	77.04(97.07)	91.78(98.02)
Best result in (Pitler et al., 2009a)	21.96(56.59)	47.13(67.30)	76.42(63.62)	16.76(63.49)	40.57(62.75)
Use sense frequency in explicit training set	26.02(52.17)	35.72(51.70)	64.94(53.97)	13.76(41.97)	35.10(49.95)
Use sense frequency in implicit training set	24.55(63.99)	16.26(70.79)	60.70(53.50)	14.75(70.51)	29.07(64.70)
Use presence of top N connectives only	21.91(52.84)	39.53(50.85)	68.84(52.93)	11.91(6.33)	35.55(40.74)

Table 4: Best result of explicit relation conversion to implicit relation compared with results using the same method.

System	Comp. vs. Not F_1 (Acc)	Cont. vs. Other F_1 (Acc)	Exp. vs. Other F_1 (Acc)	Temp. vs. Other F_1 (Acc)	Average F_1 (Acc)
Sense recognition using gold-truth explicit connectives in (Pitler et al., 2009a)	N/A	N/A	N/A	N/A	N/A(93.67)
Use sense frequency in explicit training set	41.62(50.96)	27.46(59.24)	48.44(50.88)	35.14(54.28)	38.17(53.84)
Use presence of top N connectives only	42.92(55.77)	31.83(56.05)	47.26(55.77)	37.89(58.24)	39.98(56.46)

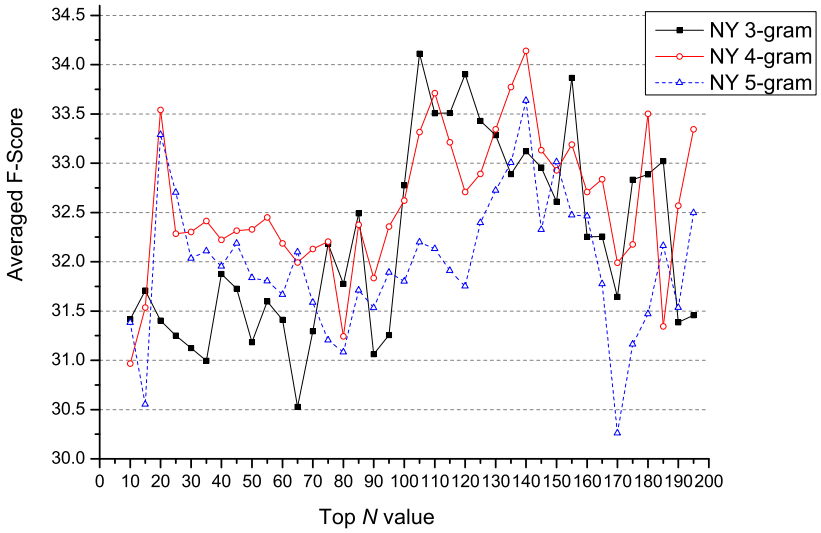


Figure 5: Curves of averages F-score on New York 3-, 4- and 5-grams language models with different top N values.

trend of these curves cannot be summarized in one sentence. But we can see that the best averaged F-scores mostly appeared in the range from $100 - 160$. For 4-grams and 5-grams models, the system achieved the top averaged F-scores when $N = 20$ as well.

4.5 Discussion

Experimental results on PDTB showed that using predicted connectives achieved the comparable F-scores of the state-of-art method.

From Table 3 we can find that our results are closely to the best performance of previous state-of-art methods in terms of averaged F-score. On the Comparison sense, our first approach has an improvement of more than 4% F-score on the previous state-of-art method (Pitler et al., 2009a). As we mentioned before, for the Expansion sense, they included EntRel relation to expand the training set and test set, which makes it impossible to perform a direct comparison. Since the positive instances size has been increased by 50%, they may achieve a higher F-score than our approach. For other relations, our best performance is slightly lower than theirs. While bearing in mind that our approach only uses a very small amount of features for implicit relation recognition. Compared

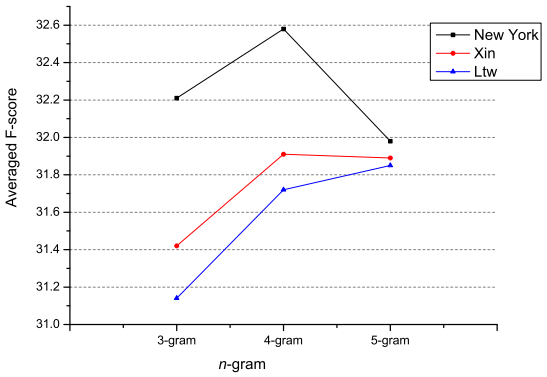


Figure 4: Curves of averaged F-score on implicit connective classification with *n*-Gram language model.

with other approaches involving thousands of features, our method is quite promising.

From Table 4 we observe comparable averaged F-score (39.98% F-score) on explicit relation data set to that on implicit relation data set. Previously, (Sporleder and Lascarides, 2008) also used the same conversion method to perform implicit relation recognition on different corpora and their best result is around 33.69% F-score. Although the two results cannot be compared directly due to different data sets, the magnitude of performance quantities is comparable and reliable.

By comparing with the above different systems, we find several useful observations. First, our method using predicted implicit connectives via a language model can help the task of implicit discourse relation recognition. The results are comparable to the previous state-of-art studies. Second, our method has a lot of advantages, i.e., a very small amount of features (several or no more than 200 vs. ten thousand), easy computation (only based on the trained language model vs. using a lot of NLP tools to extract a large amount of linguistically informed features) and fast running, which makes it more practical in real world application. Furthermore, since the language model can be trained on many corpora whether annotated or unannotated, this method is more adaptive to other languages and domains.

5 Conclusions

In this paper we have presented an approach to implicit discourse relation recognition using predicted implicit connectives via a language model. The predicted connectives have been used for implicit relation recognition in two ways, i.e., supervised and unsupervised framework. Results on the Penn Discourse Treebank 2.0 show that the predicted discourse connectives can help implicit relation recognition and the two algorithms achieve comparable F-scores with the state-of-art method. In addition, this method is quite promising due to its simple, easy to retrieve, fast run and increased adaptivity to other languages and domains.

Acknowledgments

We thank the reviewers for their helpful comments and Jonathan Ginzburg for his mentoring. This work is supported by grants from National Natural Science Foundation of China (No.60903093), Shanghai Pujiang Talent Program (No.09PJ1404500) and Doctoral Fund of Ministry of Education of China (No.20090076120029).

References

J. Baldridge and A. Lascarides. 2005. *Probabilistic head-driven parsing for discourse structure*. Proceedings of the Ninth Conference on Computational Natural Language Learning.

L. Carlson, D. Marcu, and Ma. E. Okurowski. 2001. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Proceedings of the Second SIG dial Workshop on Discourse and Dialogue.

B. Dorr. LCS Verb Database. *Technical Report Online Software Database, University of Maryland, College Park, MD*,2001.

R. Girju. 2003. *Automatic detection of causal relations for question answering*. In ACL 2003 Workshops.

S. Blair-Goldensohn. 2007. *Long-Answer Question Answering and Rhetorical-Semantic Relations*. Ph.D. thesis, Columbia Unviersity.

M. Lapata and A. Lascarides. 2004. *Inferring Sentence-internal Temporal Relations*. Proceedings of the North American Chapter of the Assocation of Computational Linguistics.

Z.H. Lin, M.Y. Kan and H.T. Ng. 2009. *Recognizing Implicit Discourse Relations in the Penn Discourse Treebank*. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing.

D. Marcu and A. Echihabi. 2002. *An Unsupervised Approach to Recognizing Discourse Relations*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.

E. Pitler, M. Raghupathy, H. Mehta, A. Nenkova, A. Lee, A. Joshi. 2008. *Easily Identifiable Discourse Relations*. Coling 2008: Companion volume: Posters.

E. Pitler, A. Louis, A. Nenkova. 2009. *Automatic sense prediction for implicit discourse relations in text*. Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics.

E. Pitler and A. Nenkova. 2009. *Using Syntax to Disambiguate Explicit Discourse Connectives in Text*. Proceedings of the ACL-IJCNLP 2009 Conference Short Papers.

M. Porter. An algorithm for suffix stripping. In *Program*, vol. 14, no. 3, pp.130-137, 1980.

R. Prasad, N. Dinesh, A. Lee, A. Joshi, B. Webber. 2006. *Annotating attribution in the Penn Discourse TreeBank*. Proceedings of the COLING/ACL Workshop on Sentiment and Subjectivity in Text.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, B. Webber. 2008. *The Penn Discourse TreeBank 2.0*. Proceedings of LREC'08.

M. Saito, K.Yamamoto, S.Sekine. 2006. *Using Phrasal Patterns to Identify Discourse Relations*. Proceeding of the HLTCNA Chapter of the ACL.

R. Soricut and D. Marcu. 2003. *Sentence Level Discourse Parsing using Syntactic and Lexical Information*. Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference.

C. Sporleder and A. Lascarides. 2008. *Using automatically labelled examples to classify rhetorical relations: an assessment*. Natural Language Engineering, Volume 14, Issue 03.

B. Wellner , J. Pustejovsky, C. H. R. S., A. Rumshisky. 2006. *Classification of discourse coherence relations: An exploratory study using multiple knowledge sources*. Proceedings of the 7th SIGDIAL Workshop on Discourse and Dialogue.

F. Wolf, E. Gibson, A. Fisher, M. Knight. 2005. *The Discourse GraphBank: A database of texts annotated with coherence relations*. Linguistic Data Consortium.

Discourse indicators for content selection in summarization

Annie Louis, Aravind Joshi, Ani Nenkova
University of Pennsylvania
Philadelphia, PA 19104, USA
{lannie,joshi,nenkova}@seas.upenn.edu

Abstract

We present analyses aimed at eliciting which specific aspects of discourse provide the strongest indication for text importance. In the context of content selection for single document summarization of news, we examine the benefits of both the graph structure of text provided by discourse relations and the semantic sense of these relations. We find that structure information is the most robust indicator of importance. Semantic sense only provides constraints on content selection but is not indicative of important content by itself. However, sense features complement structure information and lead to improved performance. Further, both types of discourse information prove complementary to non-discourse features. While our results establish the usefulness of discourse features, we also find that lexical overlap provides a simple and cheap alternative to discourse for computing text structure with comparable performance for the task of content selection.

1 Introduction

Discourse relations such as *cause*, *contrast* or *elaboration* are considered critical for text interpretation, as they signal in what way parts of a text relate to each other to form a coherent whole. For this reason, the discourse structure of a text can be seen as an intermediate representation, over which an automatic summarizer can perform computations in order to identify important spans of text to include in a summary (Ono et al., 1994; Marcu, 1998; Wolf and Gibson, 2004). In our work, we study the content selection performance of different types of discourse-based features.

Discourse relations interconnect units of a text and discourse formalisms have proposed different resulting structures for the full text, i.e. *tree* (Mann and Thompson, 1988) and *graph* (Wolf and Gibson, 2005). This *structure* is one source of information from discourse which can be used to compute the importance of text units. The *semantics* of the discourse relations between sentences could be another indicator of content importance. For example, text units connected by "cause" and "contrast" relationships might be more important content for summaries compared to those conveying "elaboration". While previous work have focused on developing content selection methods based upon individual frameworks (Marcu, 1998; Wolf and Gibson, 2004; Uzda et al., 2008), little is known about which aspects of discourse are actually correlated with content selection power.

In our work, we separate out structural and semantic features and examine their usefulness. We also investigate whether simpler intermediate representations can be used in lieu of discourse. More parsimonious, easy to compute representations of text have been proposed for summarization. For example, a text can be reduced to a set of highly descriptive topical words, the presence of which is used to signal importance for content selection (Lin and Hovy, 2002; Conroy et al., 2006). Similarly, a graph representation of the text can be computed, in which vertices represent sentences, and the nodes are connected when the sentences are similar in terms of word overlap; properties of the graph would then determine the importance of the nodes (Erkan and Radev, 2004; Mihalcea and Tarau, 2005) and guide content selection.

We compare the utility of discourse features for single-document text summarization from three frameworks: Rhetorical Structure Theory (Mann and Thompson, 1988), Graph Bank (Wolf and Gibson, 2005), and Penn Discourse Treebank (PDTB) (Prasad et al., 2008). We present a detailed analysis of the predictive power of different types of discourse features for content selection

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 147–156,
The University of Tokyo, September 24-25, 2010.

and compare discourse-based selection to simpler non-discourse methods.

2 Data

We use a collection of Wall Street Journal (WSJ) articles manually annotated for discourse information according to three discourse frameworks. The Rhetorical Structure Theory (RST) and Graph Bank (GB) corpora are relatively small compared to the Penn Discourse Treebank (PDTB) annotations that cover the 1 million word WSJ part of the Penn Treebank corpus (Marcus et al., 1994). Our evaluation requires gold standard summaries written by humans, so we perform our experiments on a subset of the overlapping documents for which we also have human summaries available.

2.1 RST corpus

RST (Mann and Thompson, 1988) proposes that coherent text can be represented as a *tree* formed by the combination of text units via discourse relations. The RST corpus developed by Carlson et al. (2001) contains discourse tree annotations for 385 WSJ articles from the Penn Treebank corpus. The smallest annotation units in the RST corpus are sub-sentential clauses, also called elementary discourse units (EDUs). Adjacent EDUs combine through rhetorical relations into larger spans such as sentences. The larger units recursively participate in relations with others, yielding one hierarchical tree structure covering the entire text.

The discourse units participating in a RST relation are assigned either nucleus or satellite status; a nucleus is considered to be more central, or important, in the text than a satellite. Relations composed of one nucleus and one satellite are called *mononuclear* relations. On the other hand, in *multinuclear* relations, two or more text units participate, and all are considered equally important. The RST corpus is annotated with 53 mononuclear and 25 multinuclear relations. Relations that convey similar meaning are grouped, resulting in 16 classes of relations: *Cause, Comparison, Condition, Contrast, Attribution, Background, Elaboration, Enablement, Evaluation, Explanation, Joint, Manner-Means, Topic-Comment, Summary, Temporal and Topic-Change*.

2.2 Graph Bank corpus

Sometimes, texts cannot be described in a tree structure as hypothesized by the RST. For example, crossing dependencies and nodes with multiple parents appear frequently in texts and do not allow a tree structure to be built (Lee et al., 2008). To address this problem, general graph representation was proposed by Wolf and Gibson (2005) as a more realistic model of discourse structure.

Graph annotations of discourse are available for 135 documents (105 from AP Newswire and 30 from the WSJ) as part of the Graph Bank corpus (Wolf and Gibson, 2005). Clauses are the basic discourse segments in this annotation. These units are represented as the nodes in a graph, and are linked with one another through 11 different rhetorical relations: *Cause-effect, Condition, Violated expectation, Elaboration, Example, Generalization, Attribution, Temporal sequence, Similarity, Contrast and Same*. The edge between two nodes representing a relation is directed in the case of asymmetric relations such as *Cause* and *Condition* and undirected for symmetric relations like *Similarity* and *Contrast*.

2.3 Penn Discourse Treebank

The Penn Discourse Treebank (PDTB) (Prasad et al., 2008) is theory-neutral and does not make any assumptions about the form of the overall discourse structure of text. Instead, this approach focuses on local and lexically-triggered discourse relations. Annotators identify explicit signals such as discourse connectives: 'but', 'because', 'while' and mark the text spans which they relate. The relations between these spans are called *explicit* relations. In addition, adjacent sentences in a discourse are also semantically related even in the absence of explicit markers. In the PDTB, these are called *implicit* relations and are annotated between adjacent sentences in the same paragraph.

For both implicit and explicit relations, senses are assigned from a hierarchy containing four top-level categories: *Comparison* (contrast, pragmatic contrast, concession, pragmatic concession), *Contingency* (cause, pragmatic cause, condition, pragmatic condition) , *Expansion* (conjunction, instantiation, restatement, alternative, exception, list) and *Temporal* (asynchronous, synchronous). The top level senses are divided into types and subtypes that represent more fine grained senses—the second level senses are listed in parentheses above.

PDTB also provides annotations for the text spans of the two arguments (referred to Arg1 and Arg2) involved in a relation. In explicit relations, the argument syntactically bound to the discourse connective is called Arg2. The other argument is

referred to as Arg1. For implicit relations, the argument occurring first in the text is named Arg1, the one appearing later is called Arg2.

2.4 Human summaries

Human summaries are available for some of the WSJ articles. These summaries are *extractive*: human judges identified and extracted important text units from the source articles and used them as such to compose the summary.

The RST corpus contains summaries for 150 documents. Two annotators selected the most important EDUs from these documents and created summaries that contain about square root of the number of EDUs in the source document. For convenience, we adopt sentences as the common unit for comparison across all frameworks. So, we mapped the summary EDUs to the sentences which contain them. Two variable length summaries for each document were obtained in this way. In some documents, it was not possible to align EDUs automatically with gold standard sentence boundaries given by the Penn Treebank and these were not used in our work. We perform our experiments on the remaining 124 document-summary pairs. These documents consisted of 4,765 sentences in total, of which 1,152 were labeled as important sentences because they contained EDUs selected by at least one annotator.

The Graph Bank corpus also contains human summaries. However, only 15 are for documents for which RST and PDTB annotations are also available. These summaries were created by fifteen human annotators who ranked the sentences in each document on a scale from 1 (low importance) to 7 (very important for a summary). For each document, we ordered the sentences according to the average rank from the annotators, and created a summary of 100 words using the top ranked sentences. The number of summary (important) sentences is 67, out of a total of 308 sentences from the 15 documents.

3 Features for content selection

In this section, we describe two sets of discourse features–structural and semantic. The structure features are derived from RST trees and do not involve specific relations. Rather they compute the importance of a segment as a function of its position in the *global* structure of the entire text. On the other hand, semantic features indicate the sense of a relation between two sentences and do not involve structure information. We compute these from the PDTB annotations. To understand the benefits of discourse information, we also study the performance of some non-discourse features standardly used in summarization.

3.1 Structural features: RST-based

Prior work in text summarization has developed content selection methods using properties of the RST tree: the nucleus-satellite distinction, notions of salience and the level of an EDU in the tree.

In early work, Ono et al. (1994) suggested a penalty score for every EDU based on their nucleus-satellite status. Since satellites of relations are considered less important than the corresponding nuclei, spans that appear as satellites can be assigned a lower score than the nucleus spans. This intuition is implemented by Ono et al. (1994) as a penalty value for each EDU, defined as the number of satellite nodes found on the path from the root of the tree to that EDU. Figure 1 shows the RST tree (Carlson et al., 2002) for the following sentence which contains four EDUs.

1. [Mr. Watkins said] 2. [volume on Interprovincial's system is down about 2% since January] 3. [and is expected to fall further,] 4. [making expansion unnecessary until perhaps the mid-1990s.]

The spans of individual EDUs are represented at the leaves of the tree. At the root of the tree, the span covers the entire text. The path from EDU 1 to the root contains one satellite node. It is therefore assigned a penalty of 1. Paths to the root from all other EDUs involve only nucleus nodes and subsequently these EDUs do not incur any penalty.

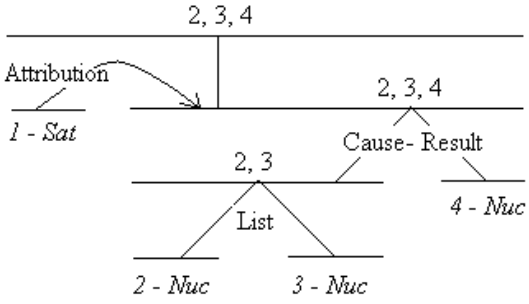


Figure 1: RST tree for the example sentence in Section 3.1.

Marcu (1998) proposed another method to utilize the nucleus-satellite distinction, rewarding nucleus status instead of penalizing satellite. He put forward the idea of a *promotion set*, consisting of

salient/important units of a text span. The nucleus is the more salient unit in the full span of a mononuclear relation. In a multinuclear relation, all the nuclei are salient units of the larger span. For example, in Figure 1, EDUs 2 and 3 participate in a multinuclear (List) relation. As a result, both EDUs 2 and 3 appear in the promotion set of their combined span. The salient units (promotion set) of each text span are shown above the horizontal line which represents the span. At the leaves, salient units are the EDUs themselves.

For the purpose of identifying important content, units in the promotion sets of nodes close to the root are hypothesized to be more important than those at lower levels. The highest promotion of an EDU occurs at the node closest to the root which contains that EDU in its promotion set. The depth of the tree from the highest promotion is assigned as the score for that EDU. Hence, the closer to the root an EDU is promoted, the better its score. Since EDUs 2, 3 and 4 are promoted all the way up to the root of the tree, the score assigned to them is equal to 4, the total depth of the tree. EDU 1 receives a depth score of 3.

However, notice that EDUs 2 and 3 are promoted to the root from a greater depth than EDU 4 but all three receive the same depth score. But an EDU promoted successively over multiple levels should be more important than one which is promoted fewer times. In order to make this distinction, a promotion score was also introduced by Marcu (1998) which is a measure of the number of levels over which an EDU is promoted. Now, EDUs 2 and 3 receive a promotion score of three while the score of EDU 4 is only two.

For our experiments, we use the nucleus-satellite penalty, depth and promotion based scores as features. Because all these scores depend on the length of the document, another set of the same features normalized by number of words in the document are also included. The penalty/score for a sentence is computed as the maximum of the penalties/scores of its constituent EDUs.

3.2 Semantic features: PDTB-based

These features represent sentences purely in terms of the relations which they participate in. For each sentence, we use the PDTB annotations to encode the sense of the relation expressed by the sentence and the type of realization (explicit or implicit).

For example, the sentence below expresses a *Contingency* relation.

In addition, its machines are easier to operate, so customers require less assistance from software.

For such sentences that contain both the arguments of a relation ie., *expresses* the relation by itself, we set the feature "expresses relation". For the above sentence, the binary feature "expresses Contingency relation" would be true.

Alternatively, sentences participating in multi-sentential relations will have one of the following features on: "contains Arg1 of relation" or "contains Arg2 of relation". Therefore, for the following sentences in an Expansion relation, we record the feature "contains Arg1 of Expansion relation" for sentence (1) and for sentence (2), "contains Arg2 of Expansion relation".

(1) Wednesday's dominant issue was Yasuda & Marine Insurance, which continued to surge on rumors of speculative buying. (2) It ended the day up 80 yen to 1880 yen.

We combine the implicit/explicit type distinction of the relations with the other features described so far, doubling the number of features. We also added features that use the second level sense of a relation. So, the relevant features for sentence (1) above would be "contains Arg1 of Implicit Expansion relation" as well as "contains Arg1 of Implicit Restatement relation" (*Restatement* is a type of *Expansion* relation (Section 2.3)).

In addition, we include features measuring the number of relations shared by a sentence (implicit, explicit and total) and the distance between arguments of explicit relations (the distance of Arg1 when the sentence contains Arg2).

3.3 Non-discourse features

We use standard non-discourse features used in summarization: length of the sentence, whether the sentence is paragraph initial or the first sentence of a document, and its offsets from document beginning as well as paragraph beginning and end (Edmundson, 1969). We also include the average, sum and product probabilities of the content words appearing in sentences (Nenkova et al., 2006) and the number of topic signature words in the sentence (Lin and Hovy, 2000).

4 Predictive power of features

We used the human summaries from the RST corpus to study which features strongly correlate with the important sentences selected by humans. For binary features such as "does the sentence con-

tain a Contingency relation", a chi-square test was computed to measure the association between a feature and sentence class (in summary or not in summary). For real-valued features, comparison between important and unimportant/non-summary sentences was done using a two-sided t-test. The significant features from our different classes are reported in the Appendix–Tables 5, 6 and 7. A brief summary of the results is provided below.

Significant features that have higher values for *sentences selected in a summary* are:

Structural: depth score and promotion score–both normalized and unnormalized.

Semantic-PDTB-level1[1]**:** contains Arg1 of Explicit Expansion, contains Arg1 of Implicit Contingency, contains Arg1 of Implicit Expansion, distance of other argument

Non-discourse: length, is the first sentence in the article, is the first sentence in the paragraph, offset from paragraph end, number of topic signature terms present, average probability of content words, sum of probabilities of content words

Significant features that have higher values for *sentences not selected in a summary* are:

Structural: Ono penalty–normalized and unnormalized.

Semantic-PDTB-level1: expresses Explicit Expansion, expresses Explicit Contingency, contains Arg2 of Implicit Temporal relation, contains Arg2 of Implicit Contingency, contains Arg2 of Implicit Expansion, contains Arg2 of Implicit Comparison, number of shared implicit relations, total shared relations

Non-discourse: offset from paragraph beginning, offset from article beginning, sentence probability based on content words.

All the structural features prove to be strong indicators for content selection. RST depth and promotion scores are higher for important sentences. Unimportant sentences have high penalties.

On the other hand, note that most of the significant sense features are descriptive of the majority class of sentences—those *not important* or *not selected* to appear in the summary (refer Table 7). For example, the second arguments of all the first level implicit PDTB relations are not preferred in human summaries. Most of the second level sense features also serve as indicators for what content should not be included in a summary. Such features can be used to derive constraints on what content is not important, but there are only few indicators associated with important sentences. Overall, out of the 25 first and second level sense features which turned out to be significantly related to a sentence class, only 8 are those indicative of important content.

Another compelling observation is that highly cognitively salient discourse relations such as *Contrast* and *Cause* are *not* indicative of important sentences. Of the features that indicate the occurrence of a particular relation in a sentence, only two are significant, but they are predictive of non-important sentences. These are "expresses Explicit Expansion" (also subtypes Conjunction and List) and "expresses Explicit Contingency".

An additional noteworthy fact is the differences between implicit and explicit relations that hold across sentences. For implicit relations, the tests show a strong indication that the second arguments of Implicit Contingency or Expansion would not be included in a summary, their first arguments however are often important and likely to appear in a summary. At the same time, for explicit relations, there is no regularity for any of the relations of which of the two arguments is more important.

All the non-discourse features turned out highly significant (Table 6). Longer sentences, those in the beginning of an article or its paragraphs and sentences containing frequent content words are preferred in human summaries.

5 Classification performance

We now test the strengths and complementary behavior of these features in a classification task to predict important sentences from input texts.

5.1 Comparison of feature classes

Table 1 gives the overall accuracy, as well as precision and recall for the important/summary sentences. Features classes were combined using logistic regression. The reported results are from 10-fold cross-validation runs on sentences from the 124 WSJ articles for which human summaries are available in the RST corpus. For the classifier using sense information from the PDTB, *all* the features described in Section 3.2 were used.

The best class of features turn out to be the structure-based ones. They outperform both non-discourse (ND) and sense features by a large margin. F-measure for the RST-based classifier is 33.50%. The semantic type of relations, on the other hand, gives no indication of content importance obtaining an F-score of only 9%. Non-discourse features provide an F-score of 19%,

[1]Features based on the PDTB level 1 senses. The significant features based on the level 2 senses are reported in the appendix.

which is much better than the semantic class but still less than structural discourse features.

The structure and semantic features are complementary to each other. The performance of the classifier is substantially improved when both types of features are used (line 6 in Table 1). The F-score for the combined classifier is 40%, which amounts to 7% absolute improvement over the structure-only classifier.

Discourse information is also complementary to non-discourse. Adding discourse structure or sense features to non-discourse (ND) features leads to better classification decisions (lines 4, 5 in Table 1). Particularly notable is the improvement when sense and non-discourse features are combined–over 10% better F-score than the classifier using only non-discourse features. The overall best classifier is the combination of discourse—structure as well as sense—and non-discourse features. Here, recall for important sentences is 34% and the precision of predictions is 62%.

We also evaluated the features using ROUGE (Lin and Hovy, 2003; Lin, 2004). ROUGE computes ngram overlaps between human reference summaries and a given system summary. This measure allows us to compare the human summaries and classifier predictions at word level rather than using full sentence matches.

To perform ROUGE evaluation, summaries for our different classes of features were obtained as follows. Important sentences for each document were predicted using a logistic regression classifier trained on all other documents. When the number of sentences predicted to be important was not sufficient to meet the required summary length, sentences predicted with lowest confidence to be non-important were selected. All summaries were truncated to 100 words. Stemming was used, and stop words were excluded from the calculation. Both human extracts were used as references.

The results from this evaluation are shown in Table 2. They closely mirror the results obtained using precision and recall. The sense features perform worse than the structural and non-discourse features. The best set of features is the one combining structure, sense and non-discourse features, with ROUGE-1 score (unigram overlap) of 0.479. Overall, combining types of features considerably improves results in all cases. However, unlike in the precision and recall evaluation, structural and non-discourse features perform very similarly.

Features used	**Acc**	**P**	**R**	**F**
structural	78.11	63.38	22.77	33.50
semantic	75.53	44.31	5.04	9.05
non-discourse (ND)	77.25	67.48	11.02	18.95
ND + semantic	77.38	59.38	20.62	30.61
ND + structural	78.51	63.49	26.05	36.94
semantic + structural	77.94	58.39	30.47	40.04
structural + semantic + ND	78.93	61.85	34.42	44.23

Table 1: Accuracy (Acc) and Precision (P), Recall (R) and F-score (F) of important sentences.

Features	**ROUGE**	**Features**	**ROUGE**
structural + semantic + ND	0.479	ND	0.432
structural + ND	0.468	LEAD	0.411
structural + semantic	0.453	semantic	0.369
semantic + ND	0.444	TS	0.338
structural	0.433		

Table 2: ROUGE-1 recall scores

Their ROUGE-1 recall scores are 0.433 and 0.432 respectively. The top ranked sentences by both sets of features appear to contain similar content.

We also evaluated sentences chosen by two baseline summarizers. The first, LEAD, includes sentences from the beginning of the article up to the word limit. This simple method is a very competitive baseline for single document summarization. The second baseline ranks sentences based on the proportion of topic signature (TS) words contained in the sentences (Conroy et al., 2006). This approach leads to very good results in identifying important content for multi-document summaries where there is more redundancy, but it is the worst when measured by ROUGE-1 on this single document task. Structure and non-discourse features outperform both these baselines.

5.2 Tree vs. graph discourse structure

Wolf and Gibson (2004) showed that the Graph Bank annotations of texts can be used for summarization with results superior to that based on RST trees. In order to derive the importance of sentences from the graph representation, they use the PageRank algorithm (Page et al., 1998). These scores, similar to RST features, are based only on the link structure; the semantic type of the relation linking the sentences is not used. In Table 3, we report the performance of structural features from RST and Graph Bank on the 15 documents with overlapping annotations from the two frameworks.

As discussed by Wolf and Gibson (2004), we find that the Graph Bank discourse representation (GB) leads to better sentence choices than using RST trees. The F-score is 48% for the GB clas-

Features	Acc	P	R	F	ROUGE
RST-struct.	81.61	63.00	31.56	42.05	0.569
GB-struct.	82.58	62.50	39.16	48.15	0.508

Table 3: Tree vs graph-based discourse features

sifier and 42% for the RST classifier. The better performance of GB method comes from higher recall scores compared to RST. Their precision values are comparable. But, in terms of ngram-based ROUGE scores, the results from RST (0.569) turn out slightly better than GB (0.508). Overall, discourse features based on structure turn out as strong indicators of sentence importance and we find both tree and graph representations to be equally useful for this purpose.

6 Lexical approximation to discourse structure

In prior work on summarization, graph models of text have been proposed that do not rely on discourse. Rather, lexical similarity between sentences is used to induce graph structure (Erkan and Radev, 2004; Mihalcea and Tarau, 2005). PageRank-based computation of sentence importance have been used on these models with good results. Now, we would like to see if the discourse graphs from the Graph Bank (GB) corpus would be more helpful for determining content importance than the general text graph based on lexical similarity (LEX). We perform this comparison on the 15 documents that we used in the previous section for evaluating tree versus graph structures. We used cosine similarity to link sentences in the lexical graph. Links with similarity less than 0.1 were removed to filter out weak relationships.

The classification results are shown in Table 4. The similarity graph representation is even more helpful than RST or GB: the F-score is 53% compared to 42% for RST and 48% for GB. The most significant improvement from the lexical graph is in terms of precision 75% which is more than 10% higher compared to RST and GB features. Using ROUGE as the evaluation metric, the lexical similarity graph, LEX (0.557), gives comparable performance with both GB (0.508) and RST (0.569) representations (refer Table 3). Therefore, for use in content selection, lexical overlap information appears to be a good proxy for building text structure in place of discourse relations.

Features	Acc	P	R	F	ROUGE
LEX-struct.	83.23	75.17	41.14	53.18	0.557

Table 4: Performance of lexrank summarizer

7 Discussion

We have analyzed the contribution of different types of discourse features–structural and semantic. Our results provide strong evidence that discourse structure is the most useful aspect. Both tree and graph representations of discourse can be used to compute the importance of text units with very good results. On the other hand, sense information from discourse does not provide strong indicators of good content but some constraints as to which content should not be included in a summary. These sense features complement structure information leading to improved performance. Further, both these types of discourse features are complementary to standardly used non-discourse features for content selection.

However, building automatic parsers for discourse information has proven to be a hard task overall (Marcu, 2000; Soricut and Marcu, 2003; Wellner et al., 2006; Sporleder and Lascarides, 2008; Pitler et al., 2009) and the state of current parsers might limit the benefits obtainable from discourse. Moreover, discourse-based structure is only as useful for content selection as simpler text structure built using lexical similarity. Even with gold standard annotations, the performance of structural features based on the RST and Graph Bank representations is not better than that obtained from automatically computed lexical graphs. So, even if robust discourse parsers exist to use these features on other test sets, it is not likely that discourse features would provide better performance than lexical similarity. Therefore, for content selection in summarization, current systems can make use of simple lexical structures to obtain similar performance as discourse features.

But it should be remembered that summary quality does not depend on content selection performance alone. Systems should also produce linguistically well formed summaries and currently systems perform poorly on this aspect. To address this problem, discourse information is vital. The most comprehensive study of text quality of automatically produced summaries was performed by Otterbacher et al. (2002). A collection of 15 automatically produced summaries was manually edited in order to correct any problems. The study

found that discourse and temporal ordering problems account for 34% and 22% respectively of all the required revisions. Therefore, we suspect that for building summarization systems, most benefits from discourse can be obtained with regard to text quality compared to the task of content selection. We plan to focus on this aspect of discourse use for our future work.

References

L. Carlson, D. Marcu, and M. E. Okurowski. 2001. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of SIGdial*, pages 1–10.

L. Carlson, D. Marcu, and M. E. Okurowski. 2002. Rst discourse treebank. *Corpus number LDC 2002T07, Linguistic Data Consortium, Philadelphia*.

J. Conroy, J. Schlesinger, and D. O'Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of ACL*.

H.P. Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

G. Erkan and D. Radev. 2004. Lexrank: Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

A. Lee, R. Prasad, A. Joshi, and B. Webber. 2008. Departures from Tree Structures in Discourse: Shared Arguments in the Penn Discourse Treebank. In *Proceedings of the Constraints in Discourse Workshop*.

C. Lin and E. Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of COLING*, pages 495–501.

C. Lin and E. Hovy. 2002. Manual and automatic evaluation of summaries. In *Proceedings of the ACL Workshop on Automatic Summarization*.

C. Lin and E. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of HLT-NAACL*.

C. Lin. 2004. ROUGE: a package for automatic evaluation of summaries. In *Proceedings of ACL Text Summarization Workshop*.

W.C. Mann and S.A. Thompson. 1988. Rhetorical structure theory: Towards a functional theory of text organization. *Text*, 8.

D. Marcu. 1998. To build text summaries of high quality, nuclearity is not sufficient. In *Working Notes of the the AAAI-98 Spring Symposium on Intelligent Text Summarization*, pages 1–8.

D. Marcu. 2000. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3):395–448.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1994. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

R. Mihalcea and P. Tarau. 2005. An algorithm for language independent single and multiple document summarization. In *Proceedings of IJCNLP*.

A. Nenkova, L. Vanderwende, and K. McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of SIGIR*.

K. Ono, K. Sumita, and S. Miike. 1994. Abstract generation based on rhetorical structure extraction. In *Proceedings of COLING*, pages 344–348.

J.C. Otterbacher, D.R. Radev, and A. Luo. 2002. Revisions that improve cohesion in multi-document summaries: a preliminary study. In *Proceedings of ACL Text Summarization Workshop*, pages 27–36.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project.

E. Pitler, A. Louis, and A. Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of ACL-IJCNLP*, pages 683–691.

R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of LREC*.

R. Soricut and D. Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of HLT-NAACL*.

C. Sporleder and A. Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14:369–416.

V.R. Uzda, T.A.S. Pardo, and M.G. Nunes. 2008. Evaluation of automatic text summarization methods based on rhetorical structure theory. *Intelligent Systems Design and Applications*, 2:389–394.

B. Wellner, J. Pustejovsky, C. Havasi, A. Rumshisky, and R. Saurí. 2006. Classification of discourse coherence relations: An exploratory study using multiple knowledge sources. In *Proceedings of SIGdial*, pages 117–125.

F. Wolf and E. Gibson. 2004. Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance. In *Proceedings of ACL*, pages 383–390.

F. Wolf and E. Gibson. 2005. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 31(2):249–288.

Appendix: Feature analysis

This appendix provides the results from statistical tests for identifying predictive features from the different classes (RST-based structural features–Table 5, Non-discourse features–Table 6 and PDTB-based sense features–Table 7).

For real-valued features, we performed a two sided t-test between the corresponding feature values for important versus non-important sentences. For features which turned out significant in each set, the value of the test statistic and significance levels are reported in the tables.

For binary features, we report results from a chi-square test to measure how indicative a feature is for the class of important or non-important sentences. For results from the chi-square test, a (+/-) sign is enclosed within parentheses for each significant feature to indicate whether the observed number of times the feature was *true* in important sentences is greater (+) than the expected value (indication that this feature is frequently associated with important sentences). When the observed frequency is less than the expected value, a (-) sign is appended.

RST Features	**t-stat**	**p-value**
Ono penalty	-21.31	2.2e-16
Depth score	16.75	2.2e-16
Promotion score	16.00	2.2e-16
Normalized penalty	-11.24	2.2e-16
Normalized depth score	17.24	2.2e-16
Normalized promotion score	14.36	2.2e-16

Table 5: Significant RST-based features

Non-discourse features	**t-stat**	**p-value**
Sentence length	3.14	0.0017
Average probability of content words	9.32	2.2e-16
Sum probability of content words	11.83	2.2e-16
Product probability of content words	-5.09	3.8e-07
Number of topic signature terms	9.47	2.2e-16
Offset from article beginning	-12.54	2.2e-16
Offset from paragraph beginning	-28.81	2.2e-16
Offset from paragraph end	7.26	5.8e-13
	χ^2	**p-value**
First sentence?	224.63 (+)	2.2e-16
Paragraph initial?	655.82 (+)	2.2e-16

Table 6: Significant non-discourse features

PDTB features	**t-stat**	**p-value**
No. of implicit relations involved	-9.13	2.2e-16
Total relations involved	-6.95	4.9e-12
Distance of Arg1	3.99	6.6e-05

Based on level 1 senses

	χ^2	**p-value**
Expresses explicit Expansion	12.96 (-)	0.0003
Expresses explicit Contingency	7.35 (-)	0.0067
Arg1 explicit Expansion	12.87 (+)	0.0003
Arg1 implicit Contingency	13.84 (+)	0.0002
Arg1 implicit Expansion	29.10 (+)	6.8e-08
Arg2 implicit Temporal	4.58 (-)	0.0323
Arg2 implicit Contingency	60.28 (-)	8.2e-15
Arg2 implicit Expansion	134.60 (-)	2.2e-16
Arg2 implicit Comparison	27.59 (-)	1.5e-07

Based on level 2 senses

	χ^2	**p-value**
Expresses explicit Conjunction	8.60 (-)	0.0034
Expresses explicit List	4.41 (-)	0.0358
Arg1 explicit Conjunction	10.35 (+)	0.0013
Arg1 implicit Conjunction	5.26 (+)	0.0218
Arg1 implicit Instantiation	18.94 (+)	1.4e-05
Arg1 implicit Restatement	15.35 (+)	8.9-05
Arg1 implicit Cause	12.78 (+)	0.0004
Arg1 implicit List	5.89 (-)	0.0153
Arg2 explicit Asynchronous	4.23 (-)	0.0398
Arg2 explicit Instantiation	10.92 (-)	0.0009
Arg2 implicit Conjunction	51.57 (-)	6.9e-13
Arg2 implicit Instantiation	12.08 (-)	0.0005
Arg2 implicit Restatement	28.24 (-)	1.1e-07
Arg2 implicit Cause	58.62 (-)	1.9e-14
Arg2 implicit Contrast	30.08 (-)	4.2e-08
Arg2 implicit List	12.31 (-)	1.9e-14

Table 7: Significant PDTB-based features

Comparing Spoken Language Route Instructions for Robots across Environment Representations

Matthew Marge
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
mrmarge@cs.cmu.edu

Alexander I. Rudnicky
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
air@cs.cmu.edu

Abstract

Spoken language interaction between humans and robots in natural environments will necessarily involve communication about space and distance. The current study examines people's close-range route instructions for robots and how the presentation format (schematic, virtual or natural) and the complexity of the route affect the content of instructions. We find that people have a general preference for providing metric-based instructions. At the same time, presentation format appears to have less impact on the formulation of these instructions. We conclude that understanding of spatial language requires handling both landmark-based and metric-based expressions.

1 Introduction

Spoken language interaction between humans and robots in natural environments will necessarily involve communication about space and distance. It is consequently useful to understand the nature of the language that humans would use for this purpose. In the present study we examine this question in the context of formulating route instructions given to robots. For practical purposes, we are also interested in understanding how presentation format affects such language. Instructions given in a physical space might differ from those given in a virtual world, which in turn may differ from those given when only a schematic representation (e.g., a map or drawing) is available.

There is general agreement that landmarks play an important role in spatial language (Daniel and Denis, 2004; Klippel and Winter, 2005; Lovelace et al., 1999; MacMahon, 2007; Michon and Denis, 2001; Nothegger et al., 2004; Raubal and Winter, 2002; Weissensteiner and Winter, 2004). However, landmarks might not necessarily be used uniformly in instructions across presentation formats. For example, people may use objects in the environment as landmarks more often when they do not have a good sense of distance in the environment. Behaviors related to spatial language may change based on the complexity of the route that a robot must take. This could be due to a combination of factors, including ease of use and personal assessment of a robot's ability to interpret specific distances over landmarks.

Several studies have investigated written or typed spatial language (e.g., MacMahon et al., 2006; Koulori and Lauria, 2009; Kollar et al., 2010). In addition, Ross (2008) studied models of spoken language interpretation in schematic views of areas. In the current study we focus on close-range spoken language route instructions.

2 Related Work

Interpreting spatial language is an important capability for systems (e.g., mobile robots) that share space with people. Human-human communication of spatial language has been extensively studied. Talmy (1983) proposed that the nature of language places constraints on how people communicate about space with others (i.e., *schematization*). Spatial descriptions are primarily influenced by how reference objects fit along fundamental axes that exhibit clear relationships with the target, and secondly by the salience of references (Carlson and Hill, 2008). People also tend to keep their spatial descriptions consistent after making an initial choice of strategy based on any existing relationships between the target to be described and other references (Vorwerg, 2009).

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 157–164,
The University of Tokyo, September 24-25, 2010.

(a) (b) (c)

Figure 1. Stimuli from the (a) schematic, (b) virtual, and (c) real-world scene experiments. Each scenario has 2 robots, Mok (left) and Aki (right). Mok is the actor in all scenarios. Outlined are possible destinations for Mok.

Studies involving spatial language with robots have thus far focused on scenarios where one robot is moved around an area using spatial prepositions (Stopp et al., 1994; Moratz et al., 2003) and further with landmarks (Skubic et al., 2002; Perzanowski et al., 2003). A number of these approaches, however, were crafted by the designers of the robots themselves and not necessarily based on an understanding of what comes naturally to people. Indeed, Shi and Tenbrink (2009) found that a person's internal linguistic representations may differ significantly from what a robot is capable of interpreting. Bugmann et al. (2004) motivated the concept of *corpus-based robotics*, where spontaneous spoken commands are collected and in turn used for designing the functionality of robots. They collected natural language instructions from people commanding robots in a miniature of a real-world environment. Our approach follows this same reasoning; we explore naturally occurring spatial language through route instructions to robots in three distinct formats (schematic, virtual, and natural environments).

3 Method

We designed and conducted three experiments using a navigation task that required the participant to "tell" a robot how to move to a target location. We varied the presentation formats of the stimuli (two-dimensional schematics, three-dimensional virtual scenes, real-world areas in-person). In each variant, the participant observed a static scene depicting two robots ("Mok" and "Aki") and a destination marker. The participant's task was to move Mok to the target destination using spoken instructions. Participants were told to act as if they were an observer of the scene but that were themselves not present in the scene; put otherwise, the robots could hear participants but not see them (and thus the participant could not figure in the instructions).

The experiment instructions directed participants to assume that Mok would understand natural language and were told to use natural expressions to specify instructions (that is, there was no "special language" necessary). Participants were told that they could take the orientations of the robots into account when they formulated their instructions. They were moreover asked to include all necessary steps in a single utterance (i.e., a turn composed of one or more spatial language commands). The robots did not move in the experiments.

Since our aim was to learn about spoken language route instructions, all participants recorded their requests using a simple recorder interface that could be activated while viewing the scene. A standard headset microphone was used. To avoid self-correction while speaking, the instructions directed participants to think about their instructions before recording. Participants could playback their instructions, and re-record them if they deemed them unsatisfactory. All interface activity was time-stamped and logged.

3.1 General variations

In their work, Hayward and Tarr (1995) found that people used spatial language with reference to landmarks most often and found it most suitable when the objects in a scene were horizontally or vertically aligned. We systematically varied three elements of the stimuli in this study: the orientations of the two robots, Mok and Aki, and the location of the destination marker. Each robot's orientation was varied four ways: directly pointing forward, right, left, or backward. The

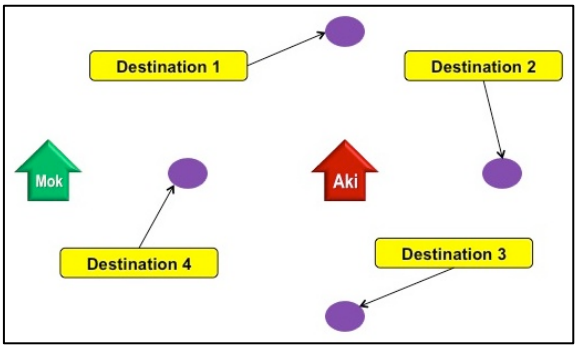


Figure 2. Specified are four potential goal destinations for Mok, the actor in all scenarios. Only one of the destinations is shown on a particular trial.

destination marker was also varied four ways: directly in front of, behind, right of, or left of Aki. These three dimensions were varied using a factorial design, yielding 64 different configurations that were presented in randomized order. Thus each participant produced 64 sets of instructions. Participants received a break at the halfway point of the session.

3.2 Schematic (2-D) Scene Experiment

Participants observed two-dimensional configurations of schematics that contained two robots (Mok and Aki) and a destination marker in this experiment. Each participant viewed a single monitor displaying a recording interface overlaid by static slides that contained the stimuli. After each participant was shown the speech recording interface and had tried it out, they proceeded through a randomly ordered slide set. In this experiment, participants viewed an overhead perspective of the scene, with the robots represented as arrows and the destination marked by purple circles (see Figures 1a and 2). The robots were represented by arrows that were meant to indicate their orientations in the scene.

3.3 Virtual (3-D) Scene and Distance Awareness Variation Experiment

In this experiment, we crafted stimuli with a three-dimensional map builder and USARSim, a virtual simulation platform designed for conducting experiments with robots (Carpin et al., 2007). The map was designed such that trials were "rooms" in a multi-room environment. Participants did not walk through the environment; they only viewed static configurations. Included in the map were instances of two Pioneer P2AT robots. All visual stimuli were presented at an eye-level view, with eyes at a height of 5'10" (see Figure 1b). The room was designed such that walls would be too far away to serve as landmarks. Visual stimuli for this experiment required full-screen access to the game engine, so the recording interface was moved to an adjoining monitor.

We included an additional condition: informing participants (or not) of the distance between the two robots. We recruited fourteen participants for this study, seven in each of two conditions. In one condition (*no-dist*), participants were not given any information related to the scale of the robots and area in the stimuli. This is equivalent to what participants experienced in the schematic scene experiment. In the second condition (*dist*), the instructions indicated that the two robots, Mok and Aki, were seven feet apart. However, no scale information (e.g., a ruler) was provided in the scene itself. This would provide the option to cast instructions in terms of absolute distances. The option to use Aki as a landmark reference point remained the same as in the first experiment. We hypothesize that participants that are not given a sense of scale will use landmarks much more often than those participants that are provided distance information.

3.4 Real-World Scene Experiment

In natural environments, it can be assumed that people generally have a good sense of scale. In this experiment, participants viewed similar stimuli to the virtual scenarios (eye-level view), but in-person (see Figure 1c). Bins were used to represent the two robots, with two eyes placed on top of each bin to indicate orientation. As in the previous experiments, participants were told to give instructions to one robot (Mok) so that it would arrive at the destination. We recorded participant instructions for 8 different configurations of the two robots (destination varied four ways, Mok's orientation varied two ways, right and left; Aki's orientation did not change). We simplified the number of orientations because we found that orientations of Mok and Aki did not influence landmark use in the previous experiments. After each instruction, participants were asked to close their eyes as the experimenter changed the orientations. Since they were not at a computer screen for this experiment, only verbal instructions were recorded, with no task times.

3.5 Participation

A total of 35 participants were recruited for this study, 10 in the schematic scene experiment, 14

Environment	Type	Spoken language route instruction (transcribed with fillers removed)
2-D	Mixed	Mok turn left / and stop at the right hand side of *Aki*.
2-D	Mixed	Turn right about sixty degrees / then go forward until you're in front of *Aki*.
3-D no-dist	Mixed	Mok turn to your left / move towards *Aki* when you are pretty close to *Aki* stop there / turn to your right / continue moving in a straight line path you will find a blue dot to your left at some point stop there / turn to your left / and reach the blue dot which is your destination.
3-D no-dist	Relative	Go forward half the distance between you and *Aki*.
3-D dist	Absolute	Rotate to your right / move forward about five feet / rotate again to your left / and move forward about seven feet.
3-D dist	Absolute	Turn to your right / move forward one foot / turn to your left / move forward ten feet / turn to your left again / move forward one foot.
Real-world	Absolute	Okay Mok I want you to go straight ahead for about five feet / then turn to your right forty five degrees / and go ahead and you're gonna hit the spot in about four feet from there.
Real-world	Mixed	Mok move to *Aki* / turn left / and move forward three feet.

Table 1. Spoken language route instructions for Mok, the moving robot, were transcribed and divided into absolute and relative steps (absolute step / relative step). Absolute steps are explicit instructions that contain metric or metric-like distances, while relative steps include Aki (the static robot) as a reference.

in the virtual scene experiment, and 11 in the real-world scene experiment. Participants ranged in age from 19 to 61 (M = 28.4 years, SD = 9.9). Of all participants, 22 were male and 14 were female. All participants were self-reported fluent English speakers.

4 Data

The first study (schematic stimuli) yielded a total of 640 route instructions (64 from each of 10 participants). All of these instructions were transcribed in-house using the CMU Communicator guidelines (Bennett and Rudnicky, 2002). In addition to the recorded instructions, we also logged participants' interactions with the speech recording interface. Since the experiment instructions ask participants to think about what they plan to say before recording their speech, we assessed their "thinking time" from this logging information.

In the second study (virtual stimuli), more participants were recruited, but they were divided into two conditions (presence/absence of an explicitly stated metric distance between the two robots in the stimuli). A total of 896 route instructions were collected in the second study (64 from each of 14 participants). Of the 14 participants recruited for this study, 12 were transcribed using Amazon's Mechanical Turk (Marge et al., 2010) with the same guidelines as the first study. In the real-world study, 8 route instructions were recorded from 11 participants and transcribed, yielding a total of 88 utterances.

5 Measurements

Several outcomes were analyzed in this study, including the time needed to formulate directions to the robot and the number of discrete steps that participants included in their instructions. We analyzed two measures, "thinking time" and word count. Thinking time represents the time between starting viewing a stimulus and pressing the "Record" button. We measured utterance length by counting the number of words spoken by participants for each instruction. Utterance-level restarts and mispronunciations were excluded from this count.

We also coded the instructions in terms of the number of discrete "steps" (see Table 1). We defined a "step" as any action where motion by Mok (the moving robot) was required to complete a sub-goal. For example, "*turn left and*

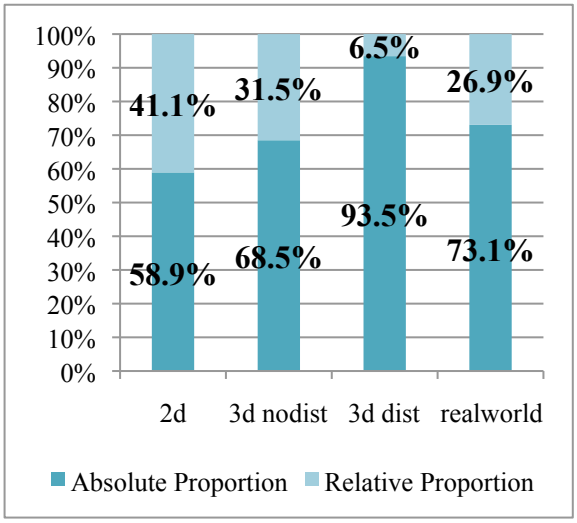


Figure 3. Mean proportion of relative steps to absolute steps across distance-naïve 2-D (schematic), distance-naïve 3-D (virtual), distance-aware 3-D (virtual), and real-world scenarios (with a 1% margin of error).

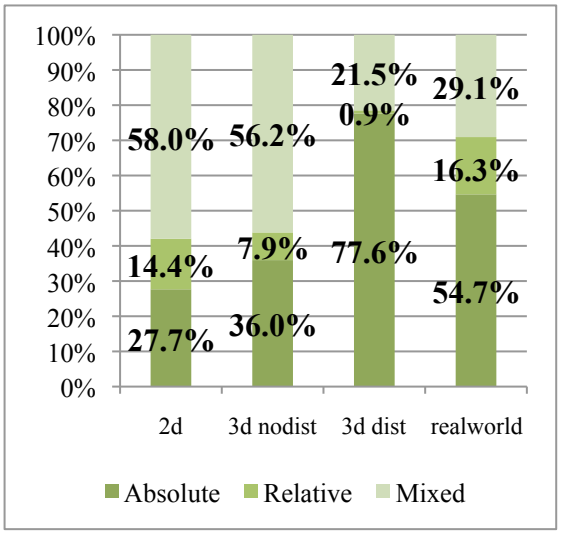


Figure 4. Proportions of instruction types across distance-naïve 2-D (schematic), distance-naïve 3-D (virtual), distance-aware 3-D (virtual), and real-world scenarios.

move forward five feet" consists of two steps: (1) a ninety degree turn to the left and (2) a movement forward of five feet to get to a new location. We divided steps into two categories, *absolute steps* and *relative steps* (similar to Levinson's (1996) *absolute* and *intrinsic* reference systems). An *absolute step* is one with explicit instructions that contain metric or metric-like distances (e.g., "*move forward two feet*", "*turn right ninety degrees*", "*move forward three steps*"). We assume that simple turns (e.g., "*turn right*") are turns of 90 degrees, and thus are absolute steps. We define a *relative step* as one that includes Aki, the static robot, in the reference (e.g., "*move forward until you reach Aki*", "*turn right until you face Aki*").

6 Results

We conducted analyses based on measures of thinking time, word count, and the number of discrete "steps" in participants' spoken language route instructions. Among the folds of the data we examined were observations from schematics without distance information (i.e., "*2-D no-dist*"), virtual scenes without giving participants distance information (i.e., "*3-D no-dist*"), virtual scenes with giving participants initial distance information (i.e., "*3-D dist*"), and real-world scenes (i.e., "*realworld*"). Since we collected an equal number of route instructions in the two virtual scene conditions (i.e., with and without being told about the distance in the environment), we directly compared properties of these instructions.

In Sections 6.2 and 6.3, absolute steps, relative steps, word count (log-10 transformed), and thinking timing (log-10 transformed) were the dependent measures in mixed-effects models of analysis of variance (for significance testing). ParticipantID was modeled as a random effect. We are interested in the population from which participants were drawn.

6.1 Adjusting Spatial Information

Landmark use was affected by participants' awareness of scale. The fewer scale cues available, the greater the number of references to landmarks. Thus, landmarks were most prevalent in instructions generated for schematic scenarios and least prevalent in the condition that explicitly specified a scale. See Figure 3 for the actual proportions. We did not inform participants of scale in the real-world condition. Interestingly, their absolute/relative mix was closer to the no-scale conditions even though they were observing an actual scene and could presumably make inferences about distances. Figure 4 shows that presentation format also affected participants' use of instructions that were entirely absolute in nature. There were fewer mixed instructions (i.e., instructions where absolute instructions were supported by landmarks) in conditions where participants had a sense of scale.

Though distances may be self-evident in real-world scenarios, they often are not in virtual en-

vironments. Participants behaved differently from real-world scenarios when we presented a non-trivial indication of scale. Participants' instructions were dominated by absolute instructions when they had a sense of scale in a virtual environment. This suggests that despite similarities in scale awareness, people formulate spatial language instructions differently when they cannot for themselves determine a sense of distance in an environment.

6.2 Sense of Distance in Virtual Stimuli

We directly compared participants' spoken language route instructions with respect to the presence (i.e., "*dist*") or absence (i.e., "*no-dist*") of distance information in the virtual environment. Though participants already had an initial preference toward using metric-based instructions, these became dominant when participants were aware of the distance in the virtual environment.

Participants that were not given a sense of distance referred to Aki as a landmark much more than when participants were given a sense of distance, confirming our initial hypothesis. We observed that the mean number of relative steps in the *no-dist* condition was nearly four times greater (1.0 relative steps per instruction) than the *dist* condition (0.2 relative steps per instruction) ($F[1, 12] = 4.6$, $p = 0.05$). As expected, participants used absolute references more in the *dist* condition, given the lack of landmark use. The mean number of absolute steps was greater in the *dist* condition (3.3 per instruction) compared to the *no-dist* condition (mean 2.4 absolute steps per instruction) ($F[1, 12] = 5.5$, $p < 0.05$).

As shown in Figure 3, the proportions of absolute to relative steps in participants' instructions show clear differences in strategy. When participants received distance information, an overwhelming majority of steps were absolute in nature (i.e., steps containing metric or metric-like distances). Aki was mentioned in steps only 6.5% of the time in the dist condition (i.e., relative steps). The proportions were more balanced in the *no-dist* condition, with 68% of steps being *absolute*. The remaining 32% of steps referred to Aki. The difference between proportions from the *no-dist* and *dist* conditions was statistically significant ($F[1,12] = 7.5$, $p < 0.05$). From these analyses we can see that distance greatly influenced participants' language instructions in virtual environments.

We further classified participants' instructions as entirely absolute, relative, or mixed in nature. When participants used landmarks, they tended to mix them with absolute steps in their instructions. Participants in the *dist* condition comprised most instructions with only absolute steps. However, even though 6.5% of steps were absolute in nature, they were distributed among one-fifth of instructions. In the *no-dist* condition, though relative steps comprised only 31.5% of total steps, they were distributed among a majority of the instructions. These results suggest that sequences of absolute steps may be sufficient on their own, but relative steps, when used, depend on the presence of some absolute terms.

6.3 Goal Location and Orientation Results

Our analysis showed that the goal location in scenarios impacted participants' instructions. For word count, participants used significantly different numbers of words based on the goal location ($F[3, 1580] = 252.2$, $p < 0.0001$). Upon further analysis, across all experiments, when the goal was closest to the Mok, the moving robot, people spoke fewer words (14 fewer words on average) compared to other locations (analysis conducted with a Tukey pairwise comparisons test). Participants also had significantly different thinking times based on the goal location ($F[3, 1502] = 6.21$, $p < 0.05$). Thinking time for the destination closest to Mok was lowest overall (on average at least 1.3s lower) and significantly different from two of the three remaining goal locations (via a Tukey pairwise comparisons test). There were no significant differences in word count and thinking time when varying Mok's orientation or Aki's orientation.

We also observed patterns in the steps people gave in their instructions. A landmark's placement, when directly interfering with a goal, increased its reference in spatial language instructions. When the goal location was blocked by Aki, we observed a high proportion of relative steps. For schematic stimuli, participants often required Mok to move past Aki in order to get to the destination. After observing the proportions of absolute steps and relative steps out of the total number of steps across destination, we found that stimuli with this destination yielded an average of 45% relative steps to 55% absolute steps. This is a greater proportion than any of the other destinations (their relative step proportions ranged from 33% to 38%).

7 Summary and Conclusions

We presented a study that examines people's close-range spoken language route instructions

for robots and how the presentation format and the complexity of the route influenced the content of instructions. Across all presentation formats, people preferred providing instructions that were absolute in nature (i.e., metric-based). Despite this preference, landmarks were used on occasion. When they were, participants' use of them was influenced by the presentation format (schematic, virtual or natural). When participants had a general sense of distance in scenes, they were much more acclimated to using specific distances to give route instructions to a robot.

Our results indicate that the goal location can influence participant effort (i.e., time to formulate) and the pattern (absolute/relative) in spoken language route instructions to robots. Several of these were predictable (e.g., least effort when goal location was closest to moving robot). When participants viewed these configurations in virtual environments, there were clear differences in their instructions based on whether or not they were given a sense of scale.

We compared the natural language instructions from the real-world condition to those from virtual stimuli. Figure 3 shows that in general, real-world participants' instructions contained similar proportions of landmarks to the *3d no-dist* (virtual) condition. However, there was a greater preference to use absolute steps in the real-world than in the virtual world; participants apparently access their own sense of scale when formulating these instructions. With respect to spatial language instructions, participants tended to treat virtual environments much like real-world environments.

This study provides useful information about methodology in the study of spatial language and also suggests principles for the design of spatial language understanding capabilities for robots in human environments. Specifically, virtual world representations, under suitable conditions, elicit language similar to that found under real-world situations, although the more information people have about the metric properties of the environment the more likely they are to use them. But even in the absence of unambiguous metrics people seem to want to use such language in the instructions that they produce. These observations can be used to inform the design of spatial language understanding for robot systems as well as guide the development of requirements for a spatial reasoning component.

Acknowledgments

This work was supported by the Boeing Company and a National Science Foundation Graduate Research Fellowship. The authors would like to thank Carolyn Rosé, Satanjeev Banerjee, Aasish Pappu, and the anonymous reviewers for their helpful comments on this work. The views and conclusions expressed in this document only represent those of the authors.

References

C. Bennett and A. I. Rudnicky. 2002. *The Carnegie Mellon Communicator Corpus*, ICSLP, 2002.

G. Bugmann, E. Klein, S. Lauria, and T. Kyriacou. 2004. *Corpus-based robotics: A route instruction example*, Intelligent Autonomous System, pp. 96-103.

L. A. Carlson and P. L. Hill. 2008. *Processing the presence, placement and properties of a distractor during spatial language tasks*, Memory and Cognition, 36, pp. 240-255.

S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper. 2007. *USARSim: A Robot Simulator for Research and Education*, International Conference on Robotics and Automation, 2007, pp. 1400-1405.

M. P. Daniel and M. Denis. 2004. *The production of route directions: Investigating conditions that favour conciseness in spatial discourse*, Applied Cognitive Psychology, 18, pp. 57-75.

W. G. Hayward and M. J. Tarr. 1995. *Spatial language and spatial representation*, Cognition, 55 (1), pp. 39-84.

A. Klippel and S. Winter. 2005. *Structural Salience of Landmarks for Route Directions*, COSIT 2005, pp. 347-362.

T. Kollar, S. Tellex, D. Roy, and N. Roy. 2010. *Toward Understanding Natural Language Directions*, Human Robot Interaction Conference (HRI-2010), pp. 259-266.

T. Koulouri and S. Lauria. 2009. *Exploring Miscommunication and Collaborative Behaviour in Human-Robot Interaction*, SIGdial 2009, pp. 111-119.

S. C. Levinson. 1996. *Frames of reference and Molyneux's question: cross-linguistic evidence*, in P. Bloom, M. Peterson, L. Nadel, and M. Garrett (Eds.), Language and space, pp. 109-169.

K. Lovelace, M. Hegarty, and D. R. Montello. 1999. *Elements of good route directions in familiar and unfamiliar environments*, in C. Freksa and D. M. Mark (Eds.), Spatial information theory: Cognitive and computational foundations of geographic information science. Berlin: Springer.

M. MacMahon. 2007. *Following Natural Language Route Instructions*, Ph.D. Thesis, University of Texas at Austin.

M. MacMahon, B. Stankiewicz, and B. Kuipers. 2006. *Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions*, 21st

National Conf. on Artificial Intelligence (AAAI), 2006, pp. 1475-1482.

M. Marge, S. Banerjee, and A. I. Rudnicky. 2010. *Using the Amazon Mechanical Turk for Transcription of Spoken Language*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010. Dallas, TX.

P. E. Michon and M. Denis. 2001. *When and why are visual landmarks used in giving directions?* in D. R. Montello (Ed.), Spatial information theory: Foundations of geographic information science, pp. 292-305. Berlin: Springer.

R. Moratz, T. Tenbrink, J. Bateman, and K. Fischer. 2003. *Spatial knowledge representation for human-robot interaction*, Spatial Cognition III. Berlin: Springer-Verlag.

C. Nothegger, S. Winter, and M. Raubal. 2004. *Selection of salient features for route directions*, Spatial Cognition and Computation, 4 (2), pp. 113-136.

D. Perzanowski, D. Brock, W. Adams, M. Bugajska, A. C. Schultz, and J. G. Trafton. 2003. *Finding the FOO: A Pilot Study for a Multimodal Interface*, IEEE Systems, Man, and Cybernetics Conference, 2003. Washington, D.C.

M. Raubal and S. Winter. 2002. *Enriching wayfinding instructions with local landmarks*, in M. J. Egenhofer and D. M. Mark (Eds.), Geographic information science, pp. 243-259. Berlin: Springer.

R. Ross. 2008. *Tiered Models of Spatial Language Interpretation*, International Conference on Spatial Cognition, 2008. Freiburg, Germany.

H. Shi and T. Tenbrink. 2009. *Telling Rolland where to go: HRI dialogues on route navigation*, in K. Coventry, T. Tenbrink, and J. Bateman (Eds.), Spatial Language and Dialogue (pp. 177-190). Oxford University Press.

M. Skubic, D. Perzanowski, A. Schultz, and W. Adams. 2002. *Using Spatial Language in a Human-Robot Dialog*, IEEE International Conference on Robotics and Automation, 2002, pp. 4143-4148. Washington, D.C.

E. Stopp, K. P. Gapp, G. Herzog, T. Laengle, and T. Lueth. 1994. *Utilizing Spatial Relations for Natural Language Access to an Autonomous Mobile Robot*, 18th German Annual Conference on Artificial Intelligence, 1994, pp. 39-50. Berlin.

L. Talmy. 1983. *How language structures space*, in H. Pick, and L. Acredolo (Eds.), Spatial Orientation: Theory, Research and Application.

C. Vorwerg. 2009. *Consistency in successive spatial utterances*, in K. Coventry, T. Tenbrink, and J. Bateman (Eds.), Spatial Language and Dialogue. Oxford University Press.

E. Weissensteiner and S. Winter. 2004. *Landmarks in the communication of route instructions*, in M. Egenhofer, C. Freksa, and H. Miller (Eds.), GIScience. Berlin: Springer.

The Dynamics of Action Corrections in Situated Interaction

Antoine Raux
Honda Research Institute USA
Mountain View, CA, USA.
araux@honda-ri.com

Mikio Nakano
Honda Research Institute Japan
Wako, Japan
nakano@jp.honda-ri.com

Abstract

In spoken communications, correction utterances, which are utterances correcting other participants utterances and behaviors, play crucial roles, and detecting them is one of the key issues. Previously, much work has been done on automatic detection of correction utterances in human-human and human-computer dialogs, but they mostly dealt with the correction of erroneous utterances. However, in many real situations, especially in communications between humans and mobile robots, the misunderstandings manifest themselves not only through utterances but also through physical actions performed by the participants. In this paper, we focus on action corrections and propose a classification of such utterances into Omission, Commission, and Degree corrections. We present the results of our analysis of correction utterances in dialogs between two humans who were engaging in a kind of on-line computer game, where one participant plays the role of the remote manager of a convenience store, and the other plays the role of a robot store clerk. We analyze the linguistic content, prosody as well as the timing of correction utterances and found that all features were significantly correlated with action corrections.

1 Introduction

Recent progress in robot technology made it a reality to have robots work in offices and homes, and spoken dialog is considered to be one of the most desired interface for such robots. Our goal is to build a spoken dialog interface for robots that can move around in an office or a house and execute tasks according to humans' requests.

Building such spoken dialog interface for robots raises new problems different from those of traditional spoken/multimodal dialog systems. The intentions behind human utterances may vary depending on the situation where the robot is and the situation changes continuously not only because the robot moves but also because humans and objects move, and human requests change. In this sense human-robot interaction is *situated*.

Of the many aspects of situated interaction, we focus on the timing structure of interaction. Although traditional spoken dialog systems deal with some timing issues such as turn-taking and handling barge-ins, timing structure in human-robot interaction is far more complex because the robot can execute physical actions and those actions can occur in parallel with utterances.

In this work we are concerned specifically with corrections in situated interaction. In joint physical tasks, human corrective behavior, which allows to repair discrepancies in participants' mutual understanding, is tightly tied to actions.

While past work on non-situated spoken dialog systems has shown the necessity and feasibility of detecting and handling corrections (Kitaoka et al., 2003; Litman et al., 2006; Gieselman and Ostendorf, 2007; Cevik et al., 2008), most of these models assume that corrections target past utterances and rely on a strict turn-based structure which is frequently violated in situated interaction. When dialog is interleaved with physical actions, the specific timing of an utterance relative to other utterances and actions is more relevant than the turn sequence.

In this paper, we propose a classification of errors and corrections in physical tasks and analyze the properties of different types of corrections in the context of human-human task-oriented interactions in a virtual environment. The next section gives some characteristics of corrections in situated interaction. Section 3 describes our experi-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 165–174,
The University of Tokyo, September 24-25, 2010.

```
Alice : Put it right above    (1)
        the lamp stand
Bob   : Here?                 (2)
Alice : A little bit more     (3)
        to the right.
(Bob Moves the frame left)    (4)
Alice : No the right!         (5)
(Bob Moves the frame right)   (6)
Alice : More...               (7)
        Alright, that's ...   (8)
(A bee flies next to Bob)     (9)
Alice : Watch out! That bee is
        going to sting you!  (10)
```

Figure 1: *Example dialog from a situated task.*

mental set up and data collection effort. Section 4 presents the results of our analysis of corrections in terms of timing, prosodic, and lexical features. These results are discussed in Section 5.

2 Corrections in Situated Tasks

2.1 Situated Tasks

We define a situated task as one for which two or more agents interact in order to perform physical actions in the (real or virtual) world. Physical actions involve moving from one place to another, as well as manipulating objects. In many cases, interaction happens simultaneously with physical actions and can be affected by them, or by other external events happening in the world. For example, Figure 1 shows an extract of a (constructed) dialog where one person (Alice) assists another (Bob) while he hangs a picture frame on a wall.

This interaction presents some similarities and differences with unimodal, non-situated dialogs. In addition to standard back-and-forth turn-taking as in turns 1-3, this example features utterances by Alice which are not motivated by Bob's utterances, but rather by (her perception of) his actions (e.g. utterance 5 is a reaction to action 4), as well as external events such as 9, which triggered response 10 from Alice. Therefore the **content** of Alice's utterances is dependent not only on Bob's, but also on events happening in the world. Similarly, the **timing** of Alice's utterances is not only conditioned on Bob's speech, prosody, etc, but also on asynchronous world events.

Robots and other agents that interact with people in real world situations need to be able to account for the impact of physical actions and world events on dialog. In the next section and the rest of this paper, we focus on correction utterances and how situational context affects how and when speakers produce them.

2.2 Corrections

Generally speaking, a correction is an utterance which aims at signaling or remedying a misunderstanding between the participants of a conversation. In other word, corrections help (re)establish common ground (Clark, 1996).

2.2.1 Previous Work

There are many dimensions along which corrections can be analyzed and many researchers have addressed this issue. Conversational Analysis (CA) has, from its early days, 1 concerned itself with corrections (usually called repairs in CA work) (Schegloff et al., 1977).

More recently, spoken dialog systems researchers have investigated ways to automatically recognize corrections. For instance, Litman et al. (2006) exploited features to automatically detect correction utterances. In addition, several attempts have been made to exploit the similarity in speech sounds and speech recognition results of a correction and the previous user utterance to detect corrections (Kitaoka et al., 2003; Cevik et al., 2008).

Going beyond a binary correction/non-correction classification scheme, Levow (1998) distinguished corrections of misrecognition errors from corrections of rejection errors and found them to have different prosodic features. Rogdriguez and Schlangen (2004) and Rieser and Moore (2005) classify corrections according to their form (e.g. Repetition, Paraphrase, Addition of information...) and function. The latter aspect is mostly characterized in terms of the source of the problem that is being corrected using models of communication such as that of Clark (1996).

In all of this very rich literature, corrections are assumed to target utterances from another participant (or even oneself, in the case of self-repair) that conflict with the hearer's expectations. While some work on embodied conversational agents (Cassell et al., 2001; Traum and Rickel, 2002) does consider physical actions as possible cues to errors and corrections, the actions are typically communicative in nature (e.g. nods, glances, gestures). Comparatively, there is extremely little work on corrections that target task actions.

A couple of exceptions are Traum et al. (1999), who discuss the type of context representation needed to handle action corrections, and Funakoshi and Tokunaga (2006), who present a model for identifying repair targets in human-robot command-and-control dialogs. While important, these papers focus on theoretical planning aspects of corrections whereas this paper focuses on an empirical analysis of human conversational behavior.

2.2.2 Action Corrections in Situated Interaction

As seen above, the vast majority of prior work on corrections concerned corrections of previous (erroneous) utterances (i.e. utterance corrections). In contrast, in this paper, we focus exclusively on corrections that target previous physical actions (i.e. action corrections).

While some classification schemes of utterance corrections are applicable to task corrections (e.g. those based on the form of the correction itself), we focus on differences that are specific to action corrections.

Namely, we distinguish three types of action errors and their related action corrections:

Commission errors occur when Bob performs an action that conflicts with Alice's expectation. Action 4 of Figure 1 is a commission error, which is corrected in turn 5.

Omission errors occur when Bob fails to react to one of Alice's utterances. A typical way for Alice to correct an omission error is to repeat the utterance to which Bob did not react.

Degree errors occur when Bob reacts with an appropriate action to Alice's utterance but fails to completely fulfill Alice's goal. This is illustrated by Alice's use of "More" in turn 7 in response to Bob's insufficient action 6.

Figure 3 illustrates the three error categories based on extracts from the corpus.

In some ways, the dichotomy Commission errors/Omission errors parallels that of Misrecognitions/Rejections by Levow (1998). This type of classification is also commonly used to analyze human errors in human factors research (Wickens et al., 1998). In addition to these two categories, we added the Degree category based on our observation of the data we collected. This aspect is somewhat specific to certain kinds of physical actions (those that can be performed to different degrees, as opposed to binary actions such as "opening the door"). However, it seems general enough to be applied to many collaborative tasks relevant to robots such as guidance, tele-operation, and joint construction.

For an automated agent, being able to classify a user utterance into one of these four categories (including non-action-correction utterances) could be very useful to make fast, appropriate decisions such as canceling the current action, or asking a clarification question to the user. This is important because in human-robot interaction, responsiveness to a correction can be critical in avoiding physical accidents. For instance, if the robot detects that the user issued a commission error correction, it can stop performing its current action even before understanding the details of the correction.

In the rest of the paper, we analyze some lexical, prosodic and temporal characteristics of action corrections in the context of human-human conversations in a virtual world.

3 The Konbini Domain and System

3.1 Simulated Environments for Human-Robot Interaction Research

One obstacle to the empirical study of situated interaction is that it requires a fully functional sophisticated robot to collect data and conduct experiments. Most such complex robots are still fragile and thus it is typically challenging to run user studies with naive subjects without severely limiting the tasks or the scope of the interaction. Another issue which comes with real world interaction is that it is difficult for the experimenter to control or monitor the events that affect the interaction. Most of the time, an expensive manual annotation of events and actions is required (see Okita et al. (2009) for an example of such an experimental setup).

To avoid these issues, robot simulators have been used. Koulouri and Lauria (2009) developed a simulator to collect dialogs between human and simulated robot using a Wizard-of-Oz method. The human can see a map of a town and teaches the robot a route and the operator operates the robot but he/she can see only a small area around the robot in the map. However, the dialog

is keyboard-based, and the situation does not dynamically change in this setting, making this approach unsuitable to the study of timing aspects. Byron and Fosler-Lussier (2006) describe a corpus of spoken dialogs collected in a setting very similar to the one we are using but, again, the environment appears to be static, thus limiting the importance of the timing of actions and utterances.

In this section, we describe a realistic, PC-based virtual world that we used to collect human-human situated dialogs.

3.2 Experimental Setup

In our experiment, two human participants collaborate in order to perform certain tasks pertaining to the management of a small convenience store in a virtual world. The two participants sit in different rooms, both facing a computer that presents a view of the virtual store. One of the participants, the Operator (O) controls a (simulated) humanoid robot whose role is to answer all customer requests. The other participant plays the role of a remote Manager (M) who sees the whole store but can only interact with O through speech.

Figure 2 shows the Operator and Manager views. M can see the whole store at any time, including how many customers there are and where they are. In addition, M knows when a particular customer has a request because the customer's character starts blinking (initially green, then yellow, then red, as time passes). M's role is then to guide O towards the customers needing attention.

On the other hand, O sees the world through the "eyes" of the robot, whose vision is limited both in terms of field of view (90 degrees) and depth (degradation of vision with depth is produced by adding a virtual "fog" to the view). When approaching a customer who has a pending request, O's view display the customer's request in the form of a caption.[1] O can act upon the virtual world by clicking on certain object such as items on the counter (to check them out), machines in the store (to repair them when needed), and various objects littering the floor (to clean them up). Each action takes a certain amount of time to perform (between 3 and 45 seconds), indicated by a progress bar that decreases as O keeps the pointer on the target object and the left mouse button down. Once the counter goes to zero the action is completed and the participants receive 50 (for partially fulfilling a customer request) or 100 points (for completely fulfilling a request).

When the session begins, customers start entering the store at random intervals, with a maximum of 4 customers in the store at any time. Each customer follows one of 14 predefined scenarios, each involving between 1 and 5 requests. Scenarios represent the customer's moves in terms of fixed way points. As a simplification, we did not implement any reactive path planning. Rather, the experimenter, sitting in a different room than either subject has the ability to temporarily take control of any customer to make them avoid obstacles.

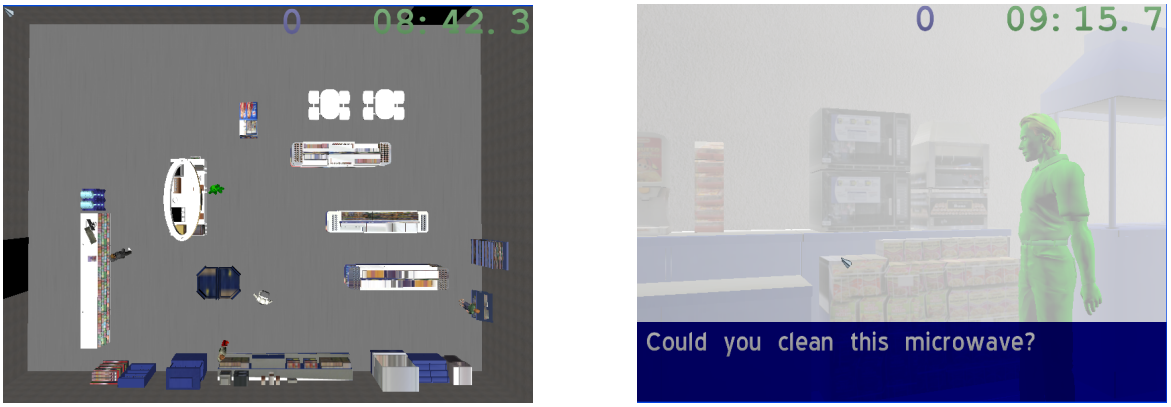
3.3 System Implementation: the Siros architecture

The experimental system described above was implemented using Siros (SItuated RObot Simulator) a new client/server architecture developed at Honda Research Institute USA to conduct human-robot interaction research in virtual worlds. Siros is similar to the architectures used by certain online video games. The server's role is to manage the virtual world and broadcast world updates to all clients so that they can be rendered to the human participants. The server receives commands from the Operator client (robot moves), runs the simulated customers according to the scenarios, and maintains the timer and the score. Anytime the trajectory of an entity (robot, customer, object) changes, the server broadcasts the related information, including entity location, orientation, and speed, to all clients.

Clients are in charge of rendering a given view of the virtual world. Rendering itself is performed by the open source Ogre 3D engine (Open Source 3D Graphics Engine, 2010). In addition, clients handle all required user interaction such as robot control and mouse-based object selection. All network messages and user actions are logged into a text file for further processing. Finally, clients have the ability to log incoming audio to a wave file, allowing synchronization between the audio signal, the user actions, and virtual world events. Spoken communication itself is handled by an external VoIP client.[2]

[1] No actual speech interaction happens between the Operator and the simulated customers.

[2] We used the open source Mumble/Murmur (Mumble/Murmur Project, 2010) system.



(a) Manager View

(b) Robot View

Figure 2: Screenshots of the Konbini data collection system.

3.4 The Konbini Corpus

3.4.1 Data Collection

Using the system described above, we collected data from 18 participants. There were 15 male and 3 female participants. All were adults living in the United States, fluent in English. All were regular users of computers but their experience with online games was diverse (from none at all to regular player). All were co-workers (associates or interns) at Honda Research Institute USA, and thus they knew each other fairly well.

Participants were randomly paired into teams. After being given the chance to read a brief introduction to the experiment's design and goals, the participants did two two-minute practice sessions to familiarize themselves with the task and control. To avoid providing too much information about the layout of the store from the start, the practice sessions used a different virtual world than the experimental sessions. The participants switched roles between the two practice sessions to get a sense of what both roles entailed (Manager and Operator). After these sessions, the team did one 10-minute experimental session, then switched roles once again and did another 10-minute session. Because the layout of the store was kept the same between the two experimental sessions, the first session represents a condition in which the Operator learns the store layout as they are performing the tasks, whereas the second session corresponds to a case where the Operator already has knowledge of the layout. Overall, 18 10-minute sessions were collected, including audio recordings as well as timestamped logs of world updates and operator actions.

3.4.2 Annotation

All recordings were orthographically transcribed and checked. The first author then segmented the transcripts into dialog acts (DAs). A DA label was attached to each act, though this information is not used in the present paper.

Subsequently, the first author annotated each semantic unit with the action correction labels described in section 2: Non-correction, Omission Correction, Commission Correction, Degree Correction. This annotation was done using the Anvil video annotation tool,[3] which presented audio recordings, transcripts, a timeline of operator actions, as well as videos of the computer screens of the participants. Only Manager DAs were annotated for corrections. The second author also annotated a subset of the data in the same way to evaluate inter-annotator agreement. Cohen's kappa between the two annotators was 0.67 for the 4-class task, and 0.76 for the binary task of any-action-correction vs non-action-correction, which is reasonable, though not very high, indicating that correction annotation on this type of dialogs is a non-trivial task, even for human annotators.

4 Analysis of Action Corrections

4.1 Overview

The total number of DAs in the corpus is 6170. Of those, 826 are corrections and 5303 are non-corrections. Overall, corrections account thus for 13.4% of the dialog acts. The split among the different correction classes is roughly equal as shown in Table 5 given in appendix. We found however significant differences across participants, in terms

[3] http://www.anvil-software.de

of total number of DAs (form 162 to 516), proportion of corrections among those DAs (from 6.8% to 30.6%), as well as distribution among the three types of action corrections.

In this section, we present the results of our statistical analysis of the correlation between a number of features and correction type. To evaluate statistical significance, we performed a one-way ANOVA using each feature as dependent variable and the correction type as independent variable. All features described here were found to significantly correlate with correction type.

4.2 Features Affecting Corrections

4.2.1 Timing

For each manager DA, we computed the time since the beginning/end of the previous manager and operator DAs, as well as of operator's actions (walk/turn). To account for reaction time, and based on our observations we ignored events happening less than 1 second before a DA.

Table 1 shows the mean durations between these events and a Manager DA, depending on the act's correction class. All corrections happen closer to Manager dialog acts than non-corrections, which reflects the fact that corrections typically occur in phases when the Manager gives instructions, as well as the fact that the Manager often repeats corrections. Commission and Degree corrections are produced closer to Operator actions than either non-corrections or Omission corrections. This reflects the fact that both Commission and Degree corrections are a reaction to an event that occurred (the Operator moved or stopped moving unexpectedly), whereas Omission corrections address a *lack* of action from the Operator, and act therefore as a "time-out" mechanism.

To better understand the relationship between moves and the timing of corrections, we computed the probability of a given DA to be an Omission, Commission and Degree correction as a function of the time elapsed since the Operator last started to move. Figure 4 shows the results.

Time since last...	NC	O	C	D
Mgr. DA	3.4 s	2.4 s	2.8 s	2.6 s
Ope. DA	5.8 s	6.7 s	6.5 s	7.5 s
Ope. move start	3.8 s	3.1 s	2.3 s	2.5 s
Ope. move end	3.9 s	3.3 s	2.7 s	2.3 s

Table 1: *Mean duration between dialog acts and Operator movements and the beginning of different corrections.*

Feature	Non-Corr	Om	Com	Deg
Perc. voiced	0.48	0.46	0.55	0.53
Min F0	-0.61	-0.41	-0.40	-0.56
Max F0	0.81	0.68	1.02	0.46
Mean F0	-0.03	0.12	0.28	-0.05
Min Power	-1.35	-1.24	-1.18	-1.55
Max Power	0.85	0.89	1.14	0.62
Mean Power	-0.03	0.09	0.24	-0.2

Table 2: *Mean Z-score of prosodic features for different correction classes.*

The probability of a DA being an Omission correction is relatively stable over time. This is consistent with the fact that Omission corrections are related to lack of action rather than to a specific action to which the Manager reacts. On the other hand, the probability of a Commission, and to lesser extent, Degree correction sharply decreases with time after an action.

4.2.2 Prosody

We extracted F0 and intensity from all manager audio files using the Praat software (Boersma and Weenink, 2010). We then normalized pitch and intensity for each speaker using a Z-transform in order to account for individual differences in mean and standard deviation. For each DA, we computed the minimum, maximum, and mean pitch and intensity, using values from voiced frames.

Table 2 shows the mean Z-score of the prosodic features for the different correction classes. Commission corrections feature higher pitch and intensity than all other classes. This is due to the fact that such corrections typically involve a higher emotional level, when the Manager is surprised or even frustrated by the behavior of the Operator. In contrast, Degree corrections, which represent a smaller mismatch between the Operator's action and the Manager's expectations are more subdued, with mean power and intensity values lower than even those of non-corrections.

4.2.3 Lexical Features

In order to identify potential lexical characteristics of correction utterances, we created binary variables indicating that a specific word from the vocabulary (804 distinct words in total) appears in a given DA based on the manual transcripts. We computed the mutual information of those binary

variables with DA's correction label.

Figure 3 shows the 10 words with highest mutual information. Not surprisingly, negative words ("NO", "DON'T"), continuation words ("MORE", "KEEP") are correlated with respectively commission and degree corrections. On the other hand, positive words ("OKAY", "YEAH") are strong indicators that a DA is not a correction.

Another lexical feature we computed was a flag indicating that a certain Manager DA is an exact repetition of the immediately preceding Manager DA. The intuition behind this feature is that corrections often involve repetitions (e.g. "Turn left [Operator turns right] Turn left!"). Overall, 10.6% of the DAs are repetitions. This number is only 6.4% on non-corrections but jumps to 45.6%, 22.5%, and 43.4% on, respectively, Omission, Commission, and Degree corrections. This confirms that, as for utterance corrections, detecting exact repetitions could prove useful for correction classification.

4.2.4 ASR Features

Since our goal is to build artificial agents, we investigated features related to automatic speech recognition. We used the Nuance Speech Recognition System v8.5. Using cross-validation, we trained a statistical language model for each correction category on the transcripts of the training portion of the data. We then ran the recognizer sequentially with all 4 language models, which generated a confidence score for each category.

Table 4 shows the mean confidence scores obtained on DAs of each class using a language model trained on specific classes. While the matching LM gives the highest score for any given class, some classes have consistently higher scores than others. In particular, Commission corrections receive low confidence scores, which might hurt the effectiveness of these features. Indeed, lexical content alone might not be not enough to distinguish non-corrections and various categories of corrections since the same expression (e.g. "Turn left") can express a simple instruction, or any kind of correction, depending on context.

LM \ Corr.	NC	O	C	D
Non-Correction	32.3	28.5	25.0	29.5
Omission	24.0	30.0	23.3	27.2
Commission	26.6	29.8	25.7	27.9
Degree	24.2	28.7	23.9	32.6

Table 4: *Mean ASR confidence score using class-specific LMs.*

5 Discussion

The results provide support for the correction classification scheme we proposed. Not only do corrections differ in many respects from non-correction utterances, but there are also significant differences between Omission, Commission, and Degree corrections. Timing features seem most useful to distinguish Commission and Degree corrections from Omission corrections and non-corrections. Emphasized prosody (high pitch and energy) is a particularly strong indicator of Commission, as well as Omission corrections. Lexical cues could be useful to all categories, provided the speech recognizer is accurate enough to recognize them, which is particularly challenging on this data given the very conversational nature of the speech. Finally, ASR scores are also potentially useful features, particularly for Omission and Degree corrections.

One advantage of timing over all other features discussed here is that timing information is available *before* the correction is actually uttered. This means that such information could be used to allow fast reaction, or to prime the speech recognizer based on the instantaneous probability of the different classes of correction.

6 Conclusion

In this paper, we analyzed correction utterances in the context of situated spoken interaction within a virtual world. We proposed a classification of action correction utterances into Omission, Commission, and Degree corrections. Our analysis of human-human data collected using a PC-based simulated environment shows that the three types of corrections have unique characteristics in terms of prosody, lexical features, as well as timing with regards to physical actions. These results can serve as the basis for further investigations into automatic detection and understanding of correction utterances in situated interaction.

References

Paul Boersma and David Weenink. 2010. Praat: doing phonetics by computer, http://www.fon.hum.uva.nl/praat.

Word (W)	$P(Non-Corr\|W)$	$P(Om\|W)$	$P(Com\|W)$	$P(Deg\|W)$
MORE	0.41	0.01	0.02	0.56
NO	0.55	0.04	0.33	0.07
RIGHT	0.67	0.15	0.04	0.14
TURN	0.69	0.17	0.06	0.08
LEFT	0.65	0.18	0.07	0.10
OKAY	0.99	0.00	0.00	0.00
YEAH	0.99	0.00	0.00	0.00
DON'T	0.59	0.01	0.33	0.07
WAY	0.49	0.05	0.42	0.04
KEEP	0.79	0.03	0.03	0.15

Table 3: *Keywords with highest mutual information with correction category.*

Donna K. Byron and Eric Fosler-Lussier. 2006. The OSU Quake 2004 corpus of two-party situated problem-solving dialogs. In *Proc. 15th Language Resource and Evaluation Conference (LREC'06)*.

Justine Cassell, Timothy Bickmore, Hannes Hgni Vilhjámsson, and Hao Yan. 2001. More Than Just a Pretty Face: Conversational Protocols and the Affordances of Embodiment. *Knowledge-Based Systems*, 14:55–64.

Mert Cevik, Fuliang Weng, and Chin hui Lee. 2008. Detection of repetitions in spontaneous speech dialogue sessions. In *Proc. Interspeech 2008*, pages 471–474.

Herbert Clark. 1996. *Using Language*. Cambridge University Press.

Kotaro Funakoshi and Takenobu Tokunaga. 2006. Identifying repair targets in action control dialogue. In *Proc. EACL 2006*, pages 177–184.

Petra Gieselman and Mari Ostendorf. 2007. Problem-Sensitive Response Generation in Human-Robot Dialogs. In *Proc. SIGDIAL 2002*.

Norihide Kitaoka, Naoko Kakutani, and Seiichi Nakagawa. 2003. Detection and Recognition of Correction Utterance in Spontaneously Spoken Dialog. In *Proc. Eurospeech 2003*, pages 625–628.

Theodora Koulouri and Stanislao Lauria. 2009. Exploring miscommunication and collaborative behaviour in human-robot interaction. In *Proc. SIGDIAL 2009*, pages 111–119.

Gina-Anne Levow. 1998. Characterizing and recognizing spoken corrections in human-computer dialogue. In *Proc. COLING-ACL '98*, pages 736–742.

Diane Litman, Julia Hirschberg, and Marc Swerts. 2006. Characterizing and predicting corrections in spoken dialogue systems. *Computational Linguistics*, 32(3):417–438.

The Mumble/Murmur Project. 2010. http://mumble.sourceforge.net.

Sandra Y Okita, Victor Ng-Thow-Hing, and Ravi K Sarvadevabhatla. 2009. Learning Together: ASIMO Developing an Interactive Learning Partnership with Children. In *Proc. RO-MAN 2009*.

OGRE Open Source 3D Graphics Engine. 2010. http://www.ogre3d.org.

Verena Rieser and Johanna Moore. 2005. Implications for generating clarification requests in task-oriented dialogues. In *Proc. 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, pages 239–246.

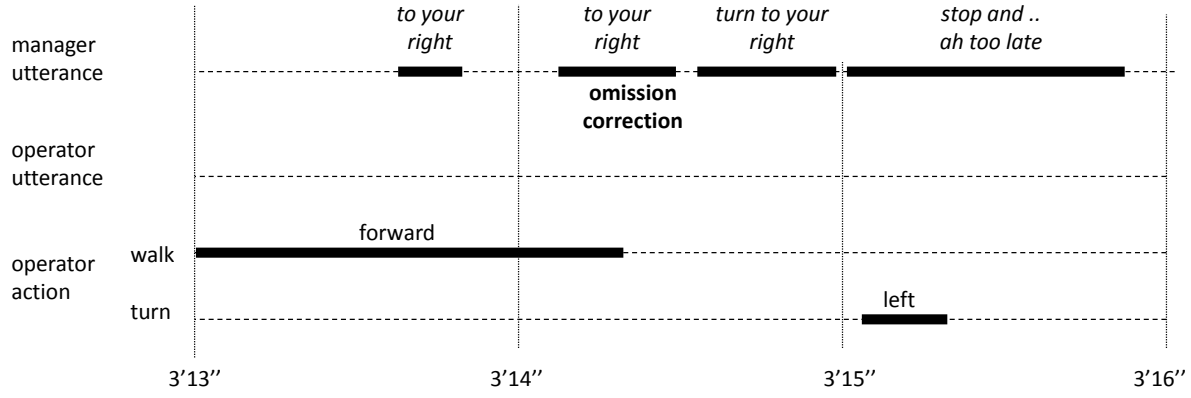
Kepa Josepa Rogdriguez and David Schlangen. 2004. Form, intonation and function of clarification requests in german task-oriented spoken dialogues. In *Proc. 8th Workshop on the Semantics and Pragmatics of Dialogue (CATALOG'04)*.

Emanuel A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. The preference for self-correction in the organization of repair in conversation. *Language*, 53(2):361–382.

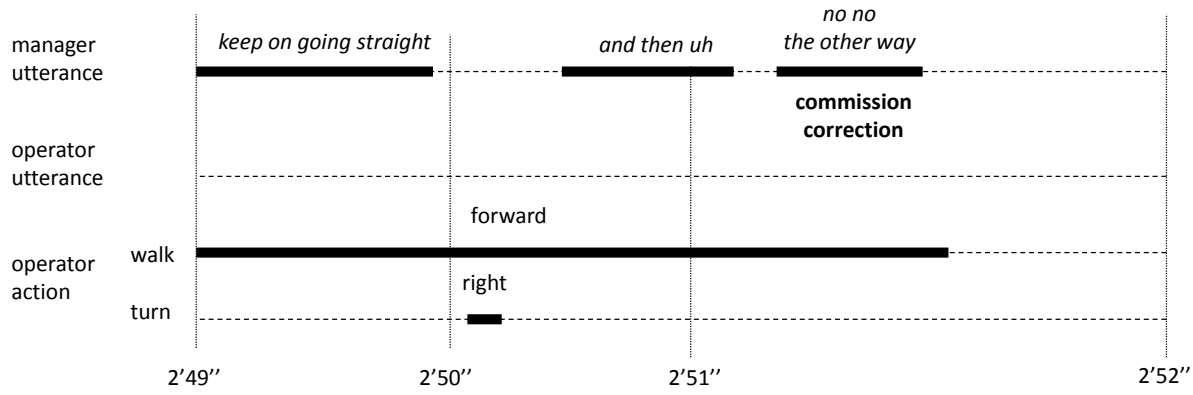
David Traum and Jeff Rickel. 2002. Embodied agents for multiparty dialogue in immersive virtual worlds. In *Proc. International Joint Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002)*, pages 766–773.

David R. Traum, Carl F. Andersen, Waiyian Chong, Darsana P. Josyula, Yoshi Okamoto, Khemdut Purang, Michael O'Donovan-Anderson, and Donald Perlis. 1999. Representations of Dialogue State for Domain and Task Independent Meta-Dialogue. *Electron. Trans. Artif. Intell.*, 3(D):125–152.

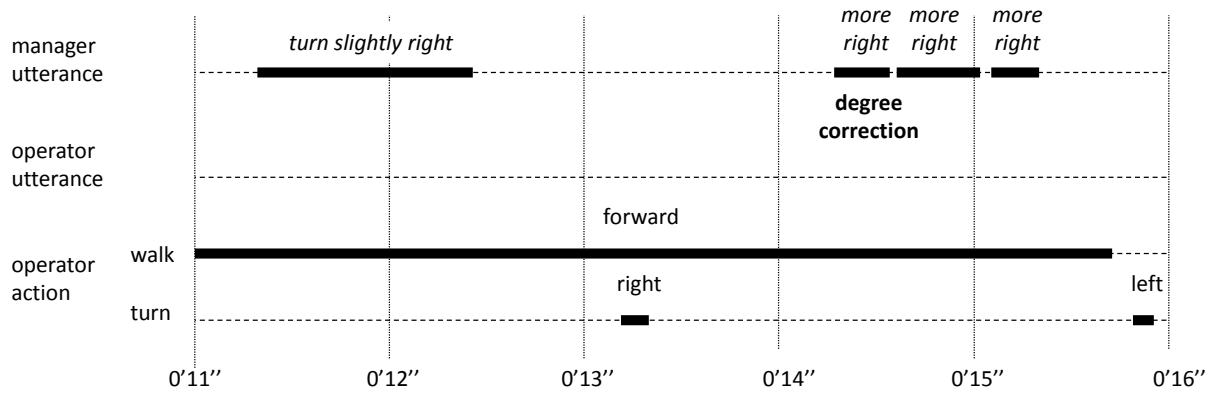
Christopher D. Wickens, Sallie E. Gordon, and Yili Liu. 1998. *An Introduction to Human Factors Engineering*. Addison-Wesley Educational Publishers Inc.



(a) Omission correction

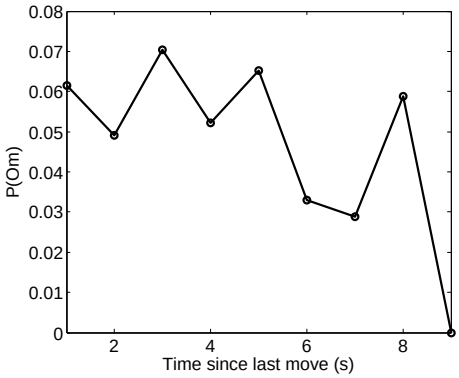


(b) Commission correction

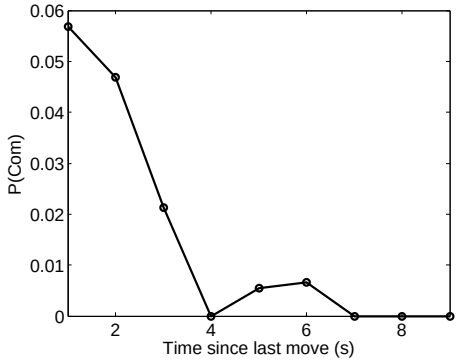


(c) Degree correction

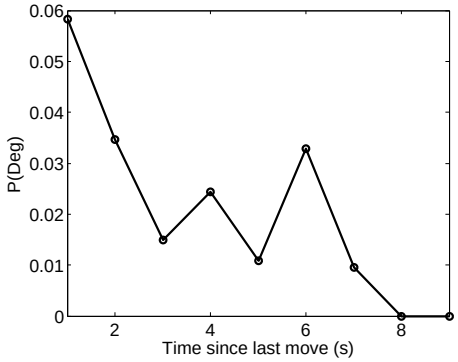
Figure 3: Example omission, commission, and degree errors and corrections. The corresponding videos can be found at `http://sites.google.com/site/antoineraux/konbini`.



(a) Omissions



(b) Commissions



(c) Degree

Figure 4: Evolution of the probability of occurrence of corrections over time after an Operator move.

Participant	Total	Non-Corr	Omission	Commission	Degree
Total	6170	5303 (86.6%)	298 (4.8%)	277 (4.5%)	251 (4.1%)
1	338	299 (88.5%)	19 (5.6%)	15 (4.4%)	5 (1.5%)
2	249	232 (93.1%)	10 (4.0%)	2 (0.8%)	0 (0.0%)
3	440	383 (87.0%)	25 (5.7%)	11 (2.5%)	9 (2.0%)
4	265	247 (93.2%)	4 (1.5%)	8 (3.0%)	0 (0.0%)
5	313	270 (86.3%)	15 (4.8%)	5 (1.6%)	17 (5.4%)
6	238	198 (83.2%)	22 (9.2%)	13 (5.5%)	5 (2.1%)
7	426	361 (84.7%)	30 (7.0%)	10 (2.3%)	23 (5.4%)
8	244	218 (89.3%)	3 (1.2%)	13 (5.3%)	9 (3.7%)
9	162	137 (84.6%)	4 (2.5%)	13 (8.0%)	8 (4.9%)
10	229	202 (88.2%)	6 (2.6%)	3 (1.3%)	12 (5.2%)
11	380	326 (85.8%)	16 (4.2%)	19 (5.0%)	19 (5.0%)
12	427	385 (90.2%)	16 (3.7%)	11 (2.6%)	15 (3.5%)
13	327	281 (85.9%)	5 (1.5%)	14 (4.3%)	27 (8.3%)
14	516	358 (69.4%)	38 (7.4%)	79 (15.3%)	39 (7.6%)
15	362	332 (91.7%)	13 (3.6%)	6 (1.7%)	11 (3.0%)
16	392	321 (81.9%)	34 (8.7%)	27 (6.9%)	10 (2.6%)
17	362	338 (85.4%)	19 (4.8%)	22 (5.6%)	17 (4.3%)
18	466	415 (89.1%)	19 (4.1%)	6 (1.3%)	25 (5.4%)

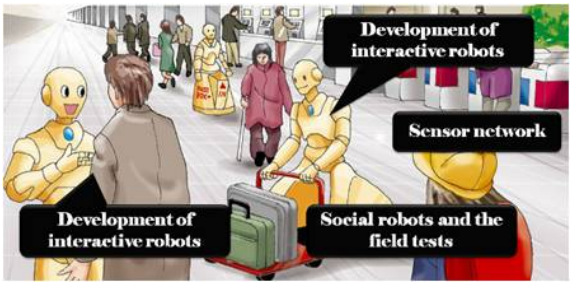
Table 5: *Frequency of the different types of corrections per participant.*

Invited Talk

Understanding Humans by Building Androids

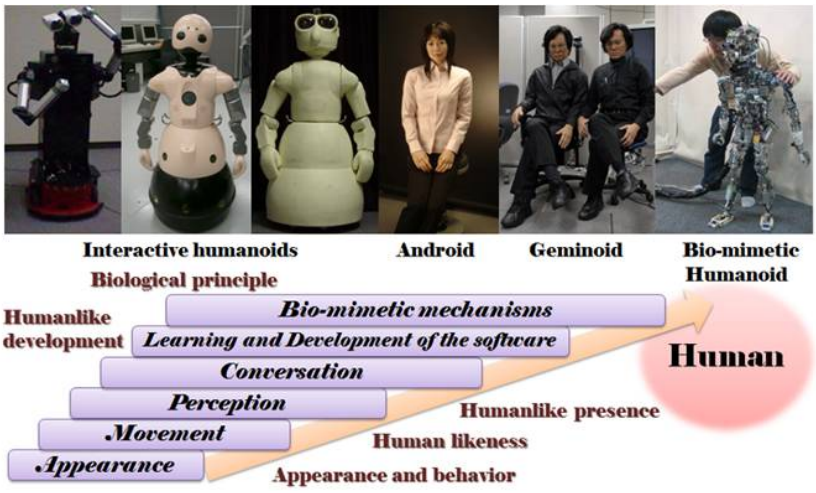
Hiroshi Ishiguro

Professor of Department of Systems Innovation, Osaka University
ATR Fellow and visiting group leader of ATR Intelligent Robotics and Communication laboratories



In the near future, we are going to use interactive humanoids in our daily life. They will provide various services by using the communication functions. In order to realize the robot society, we have to investigate many issues.

The ideal humanoid is a human since the human brain has functions to recognize humans. This means that we need to study human for building more humanlike robots. Once we get deeply into this fundamental question, the many issues come up, such as humanlike appearance, humanlike movement, and so on.



This talk introduces a series of androids that have been developed for tackling the issues and the research topics. Although it is difficult to solve completely the issues, they brings us new ideas on what human is. In other words, this research approach bridges robotics and not only cognitive science and neuroscience but also philosophy.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, page 175,
The University of Tokyo, September 24-25, 2010.

Non-humanlike Spoken Dialogue: A Design Perspective

Kotaro Funakoshi
Honda Research Institute Japan Co., Ltd.
8-1 Honcho, Wako
Saitama, Japan
funakoshi@jp.honda-ri.com

Mikio Nakano
Honda Research Institute Japan Co., Ltd.
8-1 Honcho, Wako
Saitama, Japan
nakano@jp.honda-ri.com

Kazuki Kobayashi
Shinshu University
4-17-1 Wakasato, Nagano
Nagano, Japan
kby@shinshu-u.ac.jp

Takanori Komatsu
Shinshu University
3-15-1 Tokida, Ueda
Nagano, Japan
tkomat@shinshu-u.ac.jp

Seiji Yamada
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda
Tokyo, Japan
seiji@nii.ac.jp

Abstract

We propose a non-humanlike spoken dialogue design, which consists of two elements: non-humanlike turn-taking and non-humanlike acknowledgment. Two experimental studies are reported in this paper. The first study shows that the proposed non-humanlike spoken dialogue design is effective for reducing speech collisions. It also presents pieces of evidence that show quick humanlike turn-taking is less important in spoken dialogue system design. The second study supports a hypothesis found in the first study that user preference on response timing varies depending on interaction patterns. Upon receiving these results, this paper suggests a practical design guideline for spoken dialogue systems.

1 Introduction

Speech and language are owned by humans. Therefore, spoken dialogue researchers tend to pursue a humanlike spoken dialogue. Only a few researchers positively investigate restricted (*i.e.*, non-humanlike) spoken dialogue design such as (Fernández et al., 2007).

Humanlikeness is a very important concept and sometimes it is really useful to design machines / interactions. Machines are, however, not humans. We believe humanlikenss cannot be the dominant factor, or gold-standard, for designing spoken dialogues.

Pursuing humanlikeness has at least five critical problems. (1) *Cost*: in general, humanlikeness demands powerful and highly functional hardware and software, and highly integrated systems requiring top-grade experts both for development and maintenance. All of them lead to cost overrun. (2) *Performance*: sometimes, humanlikeness forces performance to be compromised. For example, achieving quick turn-taking which humans do in daily conversations forces automatic speech recognizers, reasoners, etc. to be compromised to enable severe real-time processing. (3) *Applicability*: differences in cultures, genders, generations, situations limit the applicability of a humanlike design because it often accompanies a rigid character. For example, Shiwa et al. (2008) succeeded in improving users' impression for slow responses from a robot by using a filler but obviously use of such a filler is limited by social appropriateness. (4) *Expectancy*: humanlike systems induce too much expectancy of users that they are as intelligent as humans. It will result in disappointments (Komatsu and Yamada, 2010) and may reduce users' willingness to use systems. Keeping high willingness is quite important from the viewpoint of both research (for collecting data from users to improve systems) and business (for continuously selling systems with limited functionality). (5) *Risk*: Although it is not verified, what is called the uncanny valley (Bartneck et al., 2007) probably exists. It is commonly observed that people hate imperfect humanlike systems.

We try to avoid these problems rather than overcome them. Our position is positively exploring non-humanlike spoken dialogue design. This pa-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 176–184,
The University of Tokyo, September 24-25, 2010.

per focuses on its two elements, *i.e.*, decelerated dialogues as non-humanlike turn-taking and an artificial subtle expression (ASE) as non-humanlike acknowledgment[1], and presents two experimental studies regarding these two elements. ASEs, defined by the authors in (Komatsu et al., 2010), are simple expressions suitable for artifacts, which intuitively notify users about artifacts' internal states while avoiding the above five problems.

In Section 2, the first study, which was previously reported in (Funakoshi et al., 2010), is summarized and shows that the proposed non-humanlike spoken dialogue design is effective for reducing speech collisions. It also presents pieces of evidence that shows quick humanlike turn-taking is less important in designing spoken dialogue systems (SDSs). In Section 3, the second study, which is newly reported in this paper, shows a tendency supporting a hypothesis found in the first study that user preference on response timing varies depending on interaction patterns. Upon receiving the results of the two experiments, a design guideline for SDSs is suggested in Section 4.

2 Study 1: Reducing Speech Collisions with an Artificial Subtle Expression in a Decelerated Dialogue

An important issue in SDSs is the management of turn-taking. Failures of turn-taking due to systems' end-of-turn misdetection cause undesired speech collisions, which harm smooth communication and degrade system usability.

There are two approaches to reducing speech collisions due to end-of-turn misdetection. The first approach is using machine learning techniques to integrate information from multiple sources for accurate end-of-turn detection in early timing. The second approach is to make a long interval after the user's speech signal ends and before the system replies simply because a longer interval means no continued speech comes. As far as the authors know, all the past work takes the first approach (*e.g.*, (Kitaoka et al., 2005; Raux and Eskenazi, 2009)) because the second approach deteriorates responsiveness of SDSs. This choice is based on the presumption that users prefer a responsive system to less responsive systems. The presumption is true in most cases if the system's performance is at human level. However, if the system's performance is below human level, high responsiveness might not be vital or even be harmful. For instance, Hirasawa et al. (1999) reported that immediate overlapping backchannels can cause users to have negative impressions. Kitaoka et al. (2005) also reported that the familiarity of an SDS with backchannels was inferior to that without backchannels due to a small portion of errors even though the overall timing and frequency of backchannels was fairly good (but did not come up to human operators). Technologies are advancing but they are still below human level. We challenge the past work that took the first approach.

Figure 1: Interface robot with an embedded LED

The second approach is simple and stable against user differences and environmental changes. Moreover, it can afford to employ more powerful but computationally expensive speech processing or to build systems on small devices with limited resources. A concern with this approach is debasement of user experience due to poor responsiveness as stated above. Another issue is speech collisions due to users' following-up utterances such as repetitions. Slow responses tend to induce such collision-eliciting speech.

This section shows the results of the experiment in which participants engaged in hotel reservation tasks with an SDS equipped with an ASE-based acknowledging method, which intuitively notified a user about the system's internal state (processing). The results suggest that the method can reduce speech collisions and provide users with positive impressions. The comparisons of evaluations between systems with a slow reply speed and a moderate reply speed suggest that users of SDSs do not care about slow replies. These results indicate that decelerating spoken dialogues is not a bad idea.

2.1 Experiment

System An SDS that can handle a hotel reservation domain was built. The system was equipped

[1]In this paper, *acknowledgment* denotes that at the level 1 of the joint action ladder (Clark, 1996), which communicates the listener's identifying the signal presented by the speaker.

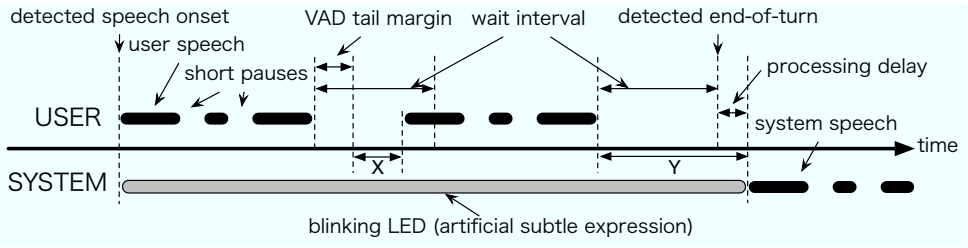


Figure 2: Behavior of the dialogue system along a timeline

with an interface robot with an LED attached to its chest (see Figure 1). Participants' utterances were recognized by an automatic speech recognizer Julius[2], and interpreted by an in-house language understander. The robot's utterances were voiced by a commercial speech synthesizer. The LCD monitor in Figure 1 was used only to show reservation details at last.

Julius output a recognition result to the system at 400 msec after an input speech signal ended, but the system awaited the next input for a fixed interval (*wait interval*, whose length is given as an experimental factor). If the system received an additional input, it awaited the next input for the same interval again. Otherwise, the system replied.

The LED started blinking at 1/30 sec even-intervals when a speech signal was detected and stopped when the system started replying. The basic function of the blinking light expression is similar to hourglass icons used in GUIs. A big difference is that basically GUIs can ignore any input while they are showing those icons, but SDSs must accept successive speech while it is blinking an LED. What we intend to do is to suppress only collision-eliciting speech such as repetitions (we call them *follow-ups*) which are negligible but difficult to be automatically distinguished from barge-ins. Barge-ins are not negligible.

Conditions and participants Two experimental factors were set-up, that is, the reply speed factor (moderate or slow reply speed) and the blinking light factor (with or without a blinking light), resulting in four conditions:

A: **slow** reply speed, **with** a blinking light,
B: **slow** reply speed, **without** a blinking light,
C: **moderate** reply speed, **with** a blinking light,
D: **moderate** reply speed, **without** a blinking light.

We randomly assigned 48 Japanese participants (mean age 30.9) to one of the four conditions.

A reply speed depends on a wait interval for which the dialogue system awaits the next input. Shiwa et al. (2008) showed that the best reply speed for a conversational robot was one second. Thus we chose 800 msec as the wait interval for the moderate reply speed because an actual reply speed was the accumulation of the wait interval and a delay for processing a user request, and 800 msec is simply twice the default length (the VAD tail margin) by which the Julius speech recognizer recognizes the end of a speech. For the slow reply speed, we chose 4 sec as the wait interval. Wait intervals include the VAD tail margin.

Figure 2 shows how the system and the LED work along with user speech. In this figure, a user utters a continuous speech with a rather long pause that is longer than the VAD tail margin but shorter than the wait interval. If the system detects the end of the user's turn and starts speaking within the interval marked with an 'X', a speech collision would occur. If the user utters a follow-up within the interval marked with a 'Y', a speech collision would occur, too. We try to suppress the former speech collision by decelerating dialogues and the latter by using a blinking light as an ASE.

Method The experiment was conducted in a room for one participant at one time. Participants entered the room and sat on a chair in front of a desk as shown in Figure 1.

The experimenter gave the participants instructions so as to reserve hotel rooms five times by talking with the robot in front of them. All of them were given the same five tasks which require them to reserve several rooms (one to three) at the same time. The meaning of the blinking light expression was not explained to them. After giving the instructions, the experimenter left the participants, and they began tasks when the robot started to talk to them. Each task was limited to up to three minutes. After finishing the tasks, the participants an-

[2]http://julius.sourceforge.jp/

swered a questionnaire. Figure 5 and Figure 6 in the appendix show one of the five task instructions, and a dialogue on that task, respectively.

2.2 Results

Reply speeds Averages of observed reply speeds were calculated from the timestamps in transcripts. They were 4.53 sec for the slow conditions and 1.42 sec for the moderate conditions.

Task completion The average number of completed tasks in the four conditions A, B, C, and D were 4.00, 3.83, 3.83, and 4.33, respectively. An ANOVA did not find any significant difference.

Speech collisions We counted speech collisions for which the SDS was responsible, that is, the cases where the robot spoke while participants were talking (*i.e.*, end-of-turn misdetections). Of course, there were speech collisions for which participants were responsible, that is, the cases where participants intentionally spoke while the robot was talking (*i.e.*, barge-ins). These speech collisions were not the targets, hence they were not included in the counts.

Speech collisions due to participants' backchannel feedbacks were not included, either. We think that it is possible to filter out such feedback because feedback utterances are usually very short and variations are small. On the other hand, as we mentioned above, it is not easy to automatically distinguish negligible speech such as repetitions from barge-ins. We want to suppress only such speech negligible but hard to distinguish from other not negligible speech.

The number of observed speech collisions in the four conditions A, B, C, and D were 5, 11, 45, and 30, respectively. First we performed an ANOVA on the number of collisions. The interaction effect was not significant ($p = 0.24$). A significant difference on the reply speed factor was found ($p < 0.005$). This result confirms that decelerating dialogues reduces collisions. The effect of the blinking light factor was not significant ($p = 0.60$).

Next we performed a Fisher's exact test (one-side) on the number of participants who had speech collisions between the two conditions of the slow reply speed (3 out of 12 for A and 8 out of 12 for B). The test found a significant difference ($p < 0.05$). This result indicates that the blinking light can reduce speech collisions by suppressing users' follow-ups in decelerated dialogues.

Impression on the dialogue and robot The participants rated 38 positive-negative adjective pairs (such as smooth vs. rough) for evaluating both the dialogue and the robot. The ratings are based on a seven-point Likert scale.

An ANOVA found a positive marginal significance ($p = 0.07$) for the blinking light in the *comfortableness* factor extracted by a factor analysis for the impression on the dialogue. In addition, an ANOVA found a positive marginal significance ($p = 0.07$) for the slow reply speed in the *modesty* factor extracted by a factor analysis for the impression on the robot. Surprisingly, no significant negative effect for the slow reply speed was found.

System evaluations The participants evaluated the SDS in two measures on a scale from 1 to 7, that is, the convenience of the system and their willingness to use the system. The greater the evaluation value is, the higher the degree of convenience or willingness.

The average scores of convenience in the four conditions A, B, C, and D were 3.50, 3.17, 3.17, and 3.92, respectively. Those of willingness were 3.58, 2.58, 2.83, and 3.42, respectively. ANOVAs did not find any significant difference among the four conditions both for the two measures.

Discussion on user preference The analysis of the questionnaire suggests that the blinking light expression gives users a comfortable impression on the dialogue. The analysis also suggests that the slow reply speed gives users a modest impression on the interface robot. Meanwhile, no negative impression with a statistical significance is found on the slow reply speed.

Although no statistically significant difference is found between the four conditions, numbers of completed tasks and convenience are strongly correlated. However, users' willingness to use the systems, which is the most important measure for systems, is inverted between condition A and D. Convenience will be primarily dominated by what degree a user's purpose (reserving rooms) is achieved, thus, it is reasonable that convenience scores correlate with the number of completed tasks. On the other hand, willingness will be dominated by not only practical usefulness but also overall usability or experience. Therefore, we can interpret that the improvements in impressions and reduction in aversive speech collisions

let condition A have the highest score for willingness. These results indicate that decelerating spoken dialogues is not a bad idea in contradiction to the common design policy in human-computer interfaces (HCIs), and they suggest to exploit merits provided by decelerating dialogues rather than pursuing quickly responding humanlike systems.

Our finding contradicts not only the common design policy in HCIs but also the design policy in human-robot interaction found by Shiwa et al. (2008), that is, *the best response timing of a communication robot is at one second*. We think this contradiction is superficial and is ascribable to the following four major differences between their study and our study.

- They adopted a within-subjects experimental design while we adopted a between-subjects design. A within-subjects design makes subjects do relative evaluations and tends to emphasis differences.
- Their question was specific in terms of response timing. Our questions were overall ratings of the system such as convenience.
- They assumed a perfect machine (Wizard-of-Oz experiment). Our system was elaborately crafted but still far from perfect.
- Our system quickly returns non-verbal responses even if verbal responses are delayed.

From these differences, we hypothesize that response timing has no significant impact on the usability of SDSs in an absolute and holistic context at least in the current state of the art spoken dialogue technology, even though users prefer a system which responds quickly to a system which responds slowly when they compare them with each other directly, given an explicit comparison metric on response timing with perfect machines.

3 Study 2: Uncovering Comfortableness of Response Timing under Different Interaction Patterns

Our conclusion in Section 2 is that SDSs do not need to quickly respond verbally as long as they quickly respond non-verbally by showing their internal states with an ASE, while many researchers try to make them verbally respond as fast as possible. Decelerating a dialogue has many practical advantages as stated above.

However, through the experiment, we have also suspected that this conclusion is not valid in some specific cases. That is, we think in some situations users feel uncomfortable with slow verbal responses primordially, and those situations are such as when users simply reply to systems' yes-no-questions or greetings. Our hypothesis is that users expect quick verbal responses (and hate slow verbal responses) only when users expect that it is not difficult for systems to understand their responses or to decide next actions. This section reports the experiment validating this hypothesis.

3.1 Experiment

To validate the hypothesis described above, we conducted a Wizard-of-Oz experiment using fixed scenarios. Participants engaged in short interactions with an interface robot and evaluated response timing of the robot. Three experimental factors were interaction patterns, response timing (wait interval), and existence of a blinking light.

Interaction patterns Five interaction patterns were setup to see the differences between situations. Each pattern consisted of three utterances. The first utterance was from the system. Upon receiving the utterance, a participant as a user of the system replied with the second utterance. Then the system responded after the given wait interval (1 sec or 4 sec) with the third utterance. Participants evaluated this interval between the second utterance and the third utterance in a measure of comfortableness.

The patterns with scenarios are shown in Figure 3. They will be referred to by abbreviations (PGG, QYQ, QNQ, PSQ, PLQ) in what follows. Note that the scenarios are originally in Japanese. Here, RequestS and RequestL mean a short request and a long request, respectively. YNQuestion and WhQuestion mean a yes-no-question and a wh-question, respectively. According to the hypothesis, we can predict that the reported comfortableness for the longer wait interval (4 sec) are worse for short and formulaic cases such as PGG and QYQ than for the long request case (*i.e.*, PLQ). In addition, we can predict that the reported comfortableness for longer intervals improves for PLQ if the robot's light blinks, while that does not improve for PGG and QYQ.

System We used the same interface robot and the LCD monitor as study 1. The experiment in this study, however, was conducted using a WOZ system.

Prompt-Greeting-Greeting (PGG)
S: Welcome to our Hotel. May I help you?
U: Hello.
S: Hello.

YNQuestion-Yes-WhQuestion (QYQ)
S: Welcome to our Hotel. Will you stay tonight?
U: Yes.
S: Can I ask your name?

YNQuestion-No-WhQuestion (QNQ)
S: Welcome to our Hotel. Will you stay tonight?
U: No.
S: How may I help you?

Prompt-RequestS-WhQuestion (PSQ)
S: Welcome to our Hotel. May I help you?
U: I would like to reserve a room from tomorrow.
S: How long will you stay?

Prompt-RequestL-WhQuestion (PLQ)
S: Welcome to our Hotel. May I help you?
U: I would like to reserve rooms with breakfast from tomorrow, one single room and one double room, non-smoking and smoking, respectively.
S: How long will you stay?

Figure 3: Interaction patterns and scenarios

First the WOZ system presents an instruction to the participant on the LCD monitor, which reveals the robot's first utterance of the given scenario (*e.g.*, "Welcome to our Hotel. May I help you?") and indicates the participant's second utterance (*e.g.*, "Hello."). Two seconds after the participant clicks the OK button on the monitor with a computer mouse, the system makes the robot utter the first utterance. Then, the participant replies, and the operator of the system end-points the end of participant's speech by clicking a button shown in another monitor for the operator in the room next to the participant's room. After the end-pointing, the system waits for the wait interval (one second or four seconds) and makes the robot utter the third utterance of the scenario. One second after, the system asks the participant to evaluate the comfortableness of the response timing of the robot's third utterance on a scale from 1 to 7 (1:very uncomfortable, 4:neutral, 7:very comfortable) on the LCD monitor.

Conditions and participants Forty participants (mean age 28.8, 20 males and 20 females) engaged in the experiment. No participant had engaged in study 1. They were randomly assigned to one of two groups (gender was balanced). The groups correspond to one of two levels of the experimental factor of the existence of a blinking light. For one group, the robot blinked its LED when it was waiting. For the other group, the robot did not blink the LED. We refer to the former group (condition) as BL (Blinking Light, n=20) and the later as NL (No Light, n=20). In summary, this experiment is within-subjects design with regard to interaction patterns and response timing and is between-subjects design with regard to the blinking light.

Method The experiment was conducted in a room for one participant at one time. Participants entered the room and sat on a chair in front of a desk as shown in Figure 1, but they did not wear headphones this time.

The experimenter gave the participants instructions so as to engage in short dialogues with the robot in front of them. They engaged in each of five scenarios shown in Figure 3 six times (three times with a 1 sec wait interval and three with 4 sec), resulting in 30 dialogues ($5 \times 3 \times 2 = 30$). The order of scenarios and intervals was randomized. The existence and meaning of the blinking light expression was not explained to them. They were not told that the system was operated by a human operator, either. After giving the instructions, the experimenter left the participants, and they practiced one time. This practice used a Prompt-RequestM-WhQuestion[3] type scenario with a wait interval of two seconds. Then, thirty dialogues were performed. Short breaks were inserted after ten dialogues. Each dialogue proceeded as explained above.

3.2 Results

End-pointing errors End-pointing was done by a fixed operator. We obtained 1,184 dialogues out of 1,200 ($= 30 \times 40$) after removing dialogues in which end-pointing failed (failures were self-reported by the operator). We sampled 30 dialogues from the 1,184 dialogues and analyzed end-pointing errors in the recorded speech data. The average error was 84.6 msec (SD=89.6).

Comfortableness This experiment was designed to grasp a preliminary sense on our hypothesis as much as possible with a limited number of participants in exchange for abandonment of use of statistical tests, because this study involved multiple factors and the interaction pattern factor was complex by itself. Therefore, in the following discussion on comfortableness, we do not refer to statistical significances.

[3]The request utterance is longer than that of RequestS and shorter than that of RequestL.

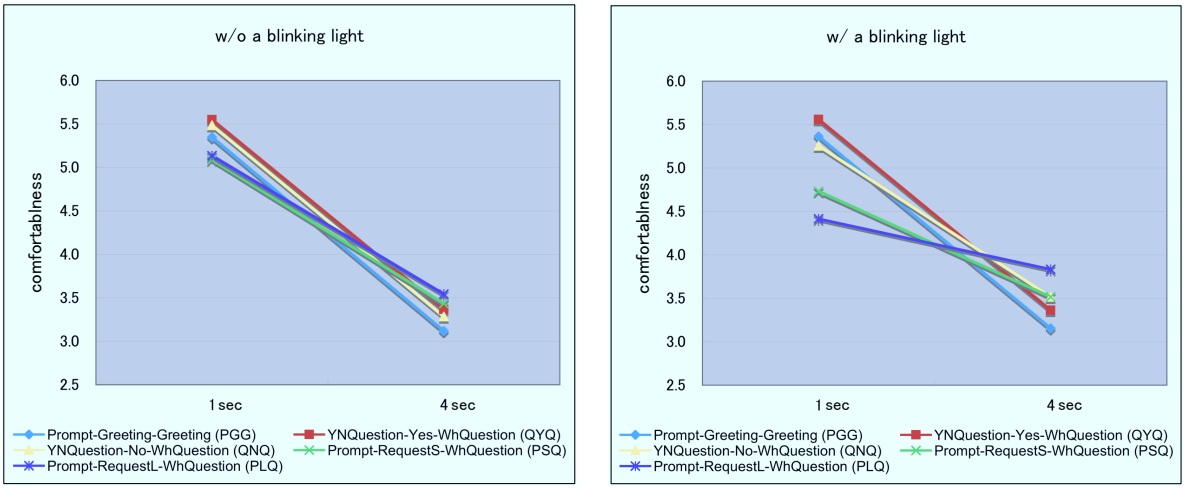


Figure 4: Comfortableness (Left: without a blinking light (NL), right: with a blinking light (BL))

Figure 4 shows regression lines obtained from the 1,184 dialogues in the two graphs for NL and BL (Detailed values are shown in Table 1). The X axes in the graphs correspond to response timing, that is, the two wait intervals of 1 sec and 4 sec. The Y axes correspond to comfortableness reported in a scale from 1 to 7. Obviously, with or without a blinking light effected comfortableness.

The results shown in the graphs support the predictions made in Section 3.1. The scores of PGG and QYQ are worse than that of PLQ at 4 sec. PGG and QYQ show no difference between NL and BL. QNQ and PSQ show differences. PLQ shows the biggest difference. In case of PLQ, the reported comfortableness at 4 sec shifted to almost the neutral position (score 4) by presenting a blinking light. This indicates that a blinking light ASE can allay the debasement of impression due to slow responses only in non-formulaic cases.

Interestingly, the blinking light expression attracted comfortableness scores to neutral both at 1 sec and at 4 sec. We can make two hypotheses on this result. One is that the blinking light expression has a negative effect which degrades comfortableness at 1 sec. The other is that the blinking light expression makes participants difficult to see differences between 1 sec and 4 sec, therefore, reported scores converge to neutral. At this stage we think that the later is more probable than the former because the scores of PGG and QYQ should be degraded at 1 sec if the former is true.

4 A Practical Design Guideline for SDSs

Summarizing the results of the experiments presented in Section 2 and Section 3, we suggest a twofold design guideline for SDSs, especially for task-oriented systems. Some interaction-oriented systems such as chatting systems are out of scope of this guideline. In what follows, first the guideline is presented and then a commentary on the guideline is described.

The guideline

(1) Never be obsessed with quick turn-taking but acknowledge users immediately

Quick turn-taking will not recompense your efforts, resources inputted, etc. Pursue it only after accomplishing all you can do without compromising performance in other elements of dialogue systems and only if it does not make system development and maintenance harder. However, quick (possibly non-verbal) acknowledgment is a requisite. You can compensate for the debasement of user experience due to slow verbal responses just by using an ASE such as a tiny blinking LED to acknowledge user speech. No instruction about the ASE is needed for users.

(2) Think of users' expectations

Users expect rather quick verbal responses to their greetings and yes-answers. ASEs will be ineffective for them. Thus it is recommended to enable your systems to quickly respond verbally to such utterances. Fortunately it is easy to anticipate such utterances. Greetings usually occur only at the beginning of dialogues or after tasks were accomplished. Yes-answers will come only after yes-no-questions. Therefore it will be able to implement an SDS that quickly responds verbally to greeting and yes-answers both without increasing development / maintenance costs and without decreasing

recognition performance, etc.

However, you should keep in mind that too quick verbal responses (0 sec interval or overlapping) may not be welcomed (Hirasawa et al., 1999; Shiwa et al., 2008). They may also induce too much expectancy in users and result in disappointments to your systems after some interactions.

Commentary on the guideline

The guideline was constructed so as to avoid the five problems pointed out in Section 1. The first point of the guideline is induced mainly from the results of study 1, and the second point is induced mainly from the results of study 2.

Although the results of study 2 indicate users prefer quick responses to slow ones as presupposed in past literature, note that the experiment in study 2 is within-subjects design with regard to the response timing factor and that within-subjects design tends to emphasis differences as discussed at the end of Section 2. The results of study 1 suggested that such an emphasized difference (*i.e.*, preference for quick responses) has no significant impact on the usability of SDSs on the whole.

5 Conclusion

This paper proposed a non-humanlike spoken dialogue design, which consists of two elements: non-humanlike turn-taking and acknowledgment. Two experimental studies were reported regarding these two elements. The first study showed that the proposed non-humanlike spoken dialogue design is effective for reducing speech collisions. This study also presented pieces of evidence that show quick humanlike turn-taking is less important in spoken dialogue system (SDS) design. The second study showed a tendency supporting a hypothesis found in the first study that user preference on response timing varies depending on interaction patterns in terms of comfortableness. Upon receiving these results, a practical design guideline for SDSs was suggested, that is, (1) never be obsessed with quick turn-taking but acknowledge users immediately and (2) think of users' expectations.

Our non-humanlike acknowledging method using an LED-based artificial subtle expression (ASE) can apply to any interfaces on wearable / handheld devices, vehicles, whatever. It is, however, difficult to directly apply it to call-centers (*i.e.*, telephone interfaces), which occupy a big portion of the deployed SDSs pie. Yet, the underlying concept: *decelerated dialogues accompanied by an ASE* will be applicable even to telephone interfaces by using an auditory ASE, which is to be explored in future work.

The guideline is supported by findings in a rather hypothetical stage. More experiments are necessary to confirm these findings. In addition, the guideline is for the current transitory period in which intelligence technologies such as automatic recognition, language processing, reasoning etc. are below human level. In that sense, the contribution of this paper might be limited. However, this period will last until a decisive paradigm shift occurs in intelligence technologies. It may come after a year, a decade, or a century.

References

C. Bartneck, T. Kanda, H. Ishiguro, and N. Hagita. 2007. Is the uncanny valley an uncanny cliff? In *Proc. RO-MAN 2007*.

H. Clark. 1996. *Using Language*. Cambridge U. P.

R. Fernández, D. Schlangen, and T. Lucht. 2007. Push-to-talk ain't always bad! comparing different interactivity settings in task-oriented dialogue. In *Proc. DECALOG 2007*.

K. Funakoshi, K. Kobayashi, M. Nakano, T. Komatsu, and S. Yamada. 2010. Reducing speech collisions by using an artificial subtle expression in a decelerated spoken dialogue. In *Proc. 2nd Intl. Symp. New Frontiers in Human-Robot Interaction*.

J. Hirasawa, M. Nakano, T. Kawabata, and K. Aikawa. 1999. Effects of system barge-in responses on user impressions. In *Proc. EUROSPEECH'99*.

N. Kitaoka, M. Takeuchi, R. Nishimura, and S. Nakagawa. 2005. Response timing detection using prosodic and linguistic information for human-friendly spoken dialog systems. *Journal of The Japanese Society for AI*, 20(3).

T. Komatsu and S. Yamada. 2010. Effects of adaptation gap on user's variation of impressions of artificial agents. In *Proc. WMSCI 2010*.

T. Komatsu, S. Yamada, K. Kobayashi, K. Funakoshi, and M. Nakano. 2010. Artificial subtle expressions: Intuitive notification methodology of artifacts. In *Proc. CHI 2010*.

A. Raux and M. Eskenazi. 2009. A finite-state turn-taking model for spoken dialog systems. In *Proc. NAACL-HLT 2009*.

T. Shiwa, T. Kanda, M. Imai, H. Ishiguro, and N. Hagita. 2008. How quickly should communication robots respond? In *Proc. HRI 2008*.

Figure 5: One of the five task instructions used in study 1

S: Welcome to Hotel Wakamatsu-Kawada. May I help you?
U: I want to stay from March 10th to 11th.
S: What kind of room would you like?
U: One non-smoking twin room and one non-smoking double room.
S: Are your reservation details correctly shown on the screen?
U: Yes. No problem.
S: Your reservation has been accepted. Thank you for using us.

Figure 6: A successful dialogue observed with the task shown in Figure 5 (translated into English)

Table 1: Detailed comfortableness scores in study 2

Interaction pattern		PGG		QYQ		QNQ		PSQ		PLQ	
Condition		NL	BL	NL	BL	NL	BL	NL	BL	NL	BL
1 sec	mean	5.34	5.36	5.55	5.56	5.48	5.25	5.09	4.73	5.13	4.41
	s.d.	1.00	1.17	1.10	1.00	1.02	1.04	1.12	1.09	1.14	1.20
	p-value	0.93		0.96		0.23		0.09		0.001	
4 sec	mean	3.12	3.16	3.37	3.36	3.28	3.52	3.43	3.52	3.54	3.83
	s.d.	0.94	1.04	0.78	0.93	0.76	0.93	0.81	0.87	0.95	0.87
	p-value	0.83		0.98		0.14		0.59		0.08	

p-values were obtained by two-sided t-tests between NL and BL. Those are shown just for reference.

Enhanced Monitoring Tools and Online Dialogue Optimisation Merged into a New Spoken Dialogue System Design Experience

Ghislain Putois
Orange Labs
Lannion, France

Romain Laroche
Orange Labs
Issy-les-Moulineaux, France

Philippe Bretier
Orange Labs
Lannion, France

firstname.surname@orange-ftgroup.com

Abstract

Building an industrial spoken dialogue system (SDS) requires several iterations of design, deployment, test, and evaluation phases. Most industrial SDS developers use a graphical tool to design dialogue strategies. They are critical to get good system performances, but their evaluation is not part of the design phase.

We propose integrating dialogue logs into the design tool so that developers can jointly monitor call flows and their associated Key Performance Indicators (KPI). It drastically shortens the complete development cycle, and offers a new design experience.

Orange Dialogue Design Studio (ODDS), our design tool, allows developers to design several alternatives and compare their relative performances. It helps the SDS developers to understand and analyse the user behaviour, with the assistance of a reinforcement learning algorithm. The SDS developers can thus confront the different KPI and control the further SDS choices by updating the call flow alternatives.

Index Terms : Dialogue Design, Online Learning, Spoken Dialogue Systems, Monitoring Tools

1 Introduction

Recent research in spoken dialogue systems (SDS) has called for a "synergistic convergence" between research and industry (Pieraccini and Huerta, 2005). This call for convergence concerns architectures, abstractions and methods from both communities. Under this motivation, several research orientations have been proposed. This paper discusses three of them : dialogue design, dialogue management, and dialogue evaluation. Dialogue design and dialogue management reflect in this paper the respective paths that industry and research have followed for building their SDS. Dialogue evaluation is a concern for both communities, but remains hard to put into operational perspectives.

The second Section presents the context and related research. The third Section is devoted to the presentation of the tools : the historical design tool, its adaptation to provide monitoring functionalities and the insertion of design alternatives. It is eventually concluded with an attempt to reassessing the dialogue evaluation. The fourth Section describes the learning integration to the tool, the constraints we impose to the learning technique and the synergy between the tools and the embedded learning capabilities. Finally, the last Section concludes the paper.

2 Context

The spoken dialogue industry is structured around the architecture of the well known industrial standard VoiceXML [1]. The underlying dialogue model of VoiceXML is a mapping of the simplistic turn-based linguistic model on the browser-server based Web architecture (McTear, 2004). The browser controls the speech engines (recognition and text-to-speech) integrated into the voice platform according to the VoiceXML document served by an application server. A VoiceXML document contains a set of prompts to play and the list of the possible interactions the user is supposed to have at each point of the dialogue. The SDS developers [2], reusing Web standards and technologies (e.g. J2EE, JSP, XML...), are used to designing directed dialogues modelled by finite state automata. Such controlled and deterministic development process allows the spoken

1. http ://www.w3c.org/TR/voicexml20/
2. In this paper, the term "SDS developers" denotes without any distinction VUI designers, application developers, and any industry engineers acting in SDS building.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 185–192,
The University of Tokyo, September 24-25, 2010.

dialogue industry to reach a balance between usability and cost (Paek, 2007). This paper argues that tools are facilitators that improve both the usability vs. cost trade-off and the reliability of new technologies.

Spoken dialogue research has developed various models and abstractions for dialogue management : rational agency (Sadek et al., 1997), Information State Update (Bos et al., 2003), functional models (Pieraccini et al., 2001), planning problem solving (Ferguson and Allen, 1998). Only a very small number of these concepts have been transferred to industry. Since the late 90's, the research has tackled the ambitious problem of automating the dialogue design (Lemon and Pietquin, 2007), aiming at both reducing the development cost and optimising the dialogue efficiency and robustness. Recently, criticisms (Paek and Pieraccini, 2008) have been formulated and novel approaches (Williams, 2008) have been proposed, both aiming at bridging the gap between research –focused on Markov-Decision-Process (Bellman, 1957) based dialogue management– and industry –focused on dialogue design process, model, and tools. This paper contributes to extend this effort. It addresses all these convergence questions together as a way for research and industry to reach a technological breakthrough.

Regarding the dialogue evaluation topic, Paek (Paek, 2007) has pointed out that while research has exerted attention about "how best to evaluate a dialogue system ?", the industry has focused on "how best to design dialogue systems ?". This paper unifies those two approaches by merging system and design evaluation in a single graphical tool. To our knowledge, ODDS is the only industrial tool which handles the complete system lifecycle, from design to evaluation.

The tools and methods presented below have been tested and validated during the design and implementation of a large real-world commercial system : the 1013+ service is the Spoken Dialogue System for landline troubleshooting for France. It receives millions of calls a year and schedules around $8,000$ appointments a week. When the user calls the system, she is presented with an open question asking her for the reason of her call. If her landline is out of service, the Spoken Dialogue System then performs some automated tests on the line, and if the problem is confirmed, try and schedule an appointment with the user for a manual intervention. If the system and the user cannot agree on an appointment slot, the call is transferred to a human operator.

3 The tools

Industry follows the VUI-completeness principle (Pieraccini and Huerta, 2005) : *"the behaviour of an application needs to be completely specified with respect to every possible situation that may arise during the interaction. No unpredictable user input should ever lead to unforeseeable behaviour"*. The SDS developers consider reliable the technologies, tools, and methodologies that help them to reach the VUI-completeness and to control it.

3.1 The Dialogue Design Tool

The graphical abstraction proposed by our dialogue design tool conforms to the general graph representation of finite state automata, with the difference that global and local variables enable to factorise several dialogue states in a single node. Transitions relate to user inputs or to internal application events such as conditions based on internal information from the current dialogue state, from the back-end, or from the dialogue history. In that sense, dialogue design in the industry generally covers more than strict dialogue management, since its specification may indicate the type of spoken utterance expected from the user at each stage of the dialogue, up to the precise speech recognition model and parameter values to use, and the generation of the system utterance, from natural language generation to speech synthesis or audio recordings.

Our dialogue design tool offers to the SDS developers a graphical abstraction of the dialogue logic, sometimes also named the call flow. Thanks to a dynamic VoiceXML generation functionality, our dialogue design tool brings the SDS developers the guarantee that VUI-completeness at the design level automatically implies a similar completeness at the implementation level. During maintenance, If the SDS developers modify a specific part of the dialogue design, the tool guarantees that solely the corresponding code is impacted. This guarantee impacts positively VUI-completeness, reliability, and development cost.

Figure 1 presents the design of a typical VoiceXML page. This page is used when the system asks the user to accept an appointment time

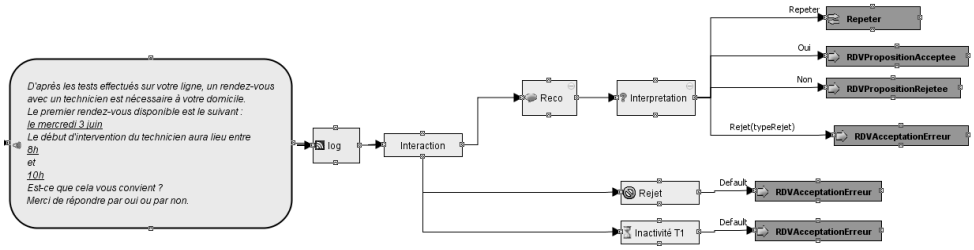


FIGURE 1 – 1013+ design excerpt : the system asks the user to confirm an appointment slot

slot. It first begins with a prompt box mixing static and dynamic prompts (the dynamic parts are underlined and realised by service-specific java code). A log box is then used some contextual session variables. Then, an interaction box is used to model the system reaction to the user behaviour : on the lower part of the Figure, we program the reaction to user inactivity or recognizer misunderstanding. In the upper part, we use a recognition box followed by a Natural Language Understanding (NLU), and we program the different output classes : repeat, yes, no and not understood. Each output is linked to a transition box, which indicates which VoiceXML page the service should call next.

3.2 Monitoring Functionalities inside the Design Tool

While researchers are focused on measuring the progress they incrementally reach, industry engineers have to deal with SDS tuning and upgrade. Their first dialogue evaluation KPI is task completion also called the automation rate because a SDS is deployed to automate specifically selected tasks. Most of the time, task completion is estimated thanks to the KPI. The KPI are difficult to exhaustively list and classify. Some are related to system measures, others are obtained thanks to dialogue annotations and the last ones are collected from users through questionnaires.

Some studies (Abella et al., 2004) investigated graphical monitoring tools. The corpus to visualise is a set of dialogue logs. The tool aims at revealing how the system transits between its possible states. As a dialogue system is too complex to enumerate all its possible states, the dialogue logs are regarded as a set of variables that evolve during time and the tool proposes to make a projection on a subset of these variables. This way, the generated graphs can either display the call flow, how the different steps are reached and where they lead, or display how different variables, as the number of errors evolve. This is mainly a tool for understanding how the users behave, because it has no direct connection with the way how the system was built. As consequence to this, it does not help to diagnose how to make it better. In other words, it does evaluate the system but does not meet one of our goal : the convergence between design and evaluation.

On the opposite, our graphical design tool provides an innovative functionality : local KPI projection into the original dialogue design thanks to an extensive logging. A large part of the KPI are automatically computed and displayed. As a consequence, it is possible to display percentage of which responses the system recognised, the users actually gave, and see how these numbers match the various KPI. It is one example among the numerous analysis views this graphical tool can provide.

3.3 Insertion of Alternatives

The 1013+ service has been used to test three kinds of design alternatives. The first kind is a strategy alternative : the service can choose between offering an appointment time slot to the client, or asking her for a time slot. This decision defines whether the next dialogue step will be system-initiative or user-initiative. The second kind is a speaking style alternative : the service can either be personified by using the “I” pronoun, adopt a corporate style by using the “We” pronoun, or speak in an impersonal style by using the passive mode. The third kind is a Text-To-Speech alternative : the service can use a different wording or prosody for a given sentence.

Figure 2 displays a monitoring view of an interaction implementation with alternatives. The recognition rate is the projected KPI on the graph at each branch. Other performance indicators are displayed at the bottom of the window : here, it

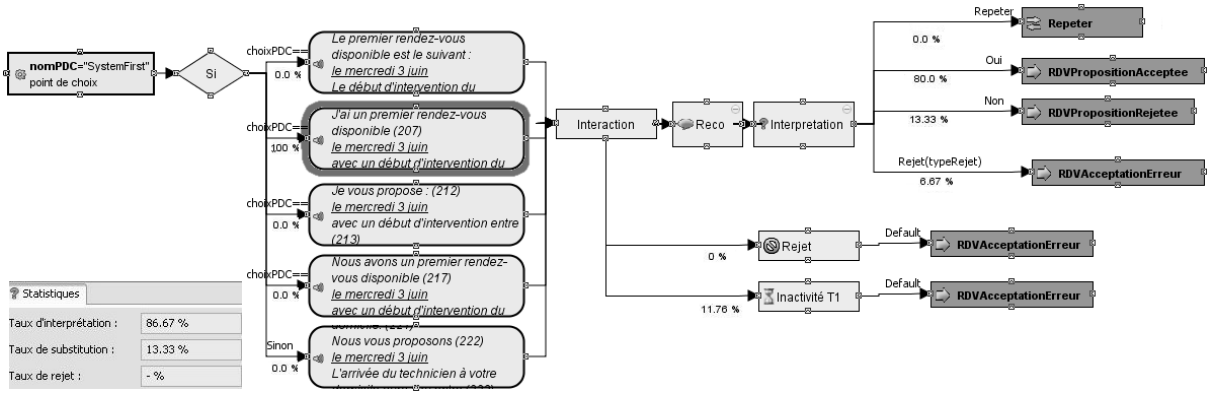


FIGURE 2 – Some user experience feedbacks related to a selected prompt alternative.

is the actual rate of correct semantic decoding, the semantic substitution rate, and the semantic rejection rate. The selection of the highlighted box conditions the displayed logs.

Our design tool also provides a multivariate testing functionality. This method consists in testing multiple alternatives and selecting the best one on a fixed set of predetermined criteria. Regarding the VUI-completeness, presenting the complete automaton to the SDS developers is acceptable, as long as they can inspect and control every branch of the design. In general, they even come up with several competing designs or points of choice, which can only be properly selected from in a statistical manner. The ability to compare all the dialogue design alternatives in the same test-field is a major factor to boost up SDS enhancement by drastically reducing the time needed. When we were developing the current 1013+ version, we have been able to develop the 5 main alternatives in less than a month, where it had taken a month and a half for a unique alternative in previous versions. It brings a statistical relevance in the causal link between the tested alternatives and the differences in performance measures, because it ensures a good random input space coverage.

The KPI graphical projection into the dialogue design covers the dialogue alternatives : KPI computation just needs to be conditioned by the alternatives. Figure 2 illustrates the merge of several system prompt alternatives inside a single design. It represents the prompt alternatives the system can choose when proposing an appointment time slot. An action block informs the Learning Manager about the current dialogue state and available dialogue alternatives. An “If” block then activates the prompt alternative corresponding to a local variable “choixPDC” filled by the Learning Manager. The rest of the design is identical to the design presented in Figure 1.

The displayed KPI are conditioned by the selected alternative (here, the second wording circled in bold grey). ODDS then indicates how the dialogue call flow is breakdown into the different alternatives. As we have here conditioned the displayed information by the second alternative, this alternative receives 100% of the calls displayed, when the other alternatives are not used. We can then see the different outcomes for the selected alternative : the customer answer have lead to a timeout of the recognition in 11.78% of the cases, and amongst the recognised sentences, 80% were an agreement, 13.33% were a reject, and 6.67% were not understood.

On the bottom-left part, one can display more specific KPI, such as good interpretation rate, substitution rate, and reject rate. These KPI are computed after the collected logs have been manually annotated, which remains an indispensable process to monitor and improve the recognition and NLU quality, and thus the overall service quality.

Conditioning on another alternative would have immediately led to different results, and someway, embedding the user experience feedback inside the dialogue design forms a new material to touch and feel : the SDS developers can now sculpt a unique reactive material which contains the design and the KPI measures distribution. By looking at the influence of each alternative on the KPI when graphically selecting the alternatives, the SDS developers are given a reliable means to understand how to improve the system.

3.4 Reassessing Dialogue Evaluation

The traditional approaches to dialogue evaluation attempt to measure how best the SDS is adapted to the users. We remind that each interaction between the user and the SDS appears to be a unique performance. First, each new dialogue is co-built in a unique way according to both the person-specific abilities of the user and the possibilities of the SDS. Second, the user adapts very quickly to new situations and accordingly changes her practices. The traditional approaches to dialogue evaluation are eventually based on the fragile reference frame of the user, not reliable enough for a scientific and an industrial approach of the spoken dialogue field, mostly because of the inability to get statistical call volumes for all the dialogue alternatives.

This suggests for a shift in the reference frame used for dialogue evaluation : instead of trying to measure the adequacy between the SDS and the user in the user's reference frame, one can measure the adequacy between the user and the SDS in the design reference frame composed by the dialogue logic, the KPI and their expected values. Taking the design as the reference allows reassessing the dialogue evaluation. The proposed basis for dialogue evaluation is reliable for the SDS developers because it is both stable and entirely under control. Deviations from the predicted situations are directly translated into anomalous values of measurable KPI that raise alerts. These automatically computable alerts warn the SDS developers about the presence of issues in their dialogue design.

4 Dialogue design learning

As presented in previous Section, the alternative insertion is an enabler for the dialogue system analysis tools. It provides the SDS developers with a novel call flow visualisation experience. The further step to this approach is to automate at least a part of those analyses and improvements with learning capabilities.

4.1 Constraints

The objective is to automatically choose online the best alternative among those proposed in the design tool, and to report this choice to the SDS developers via the monitoring functionalities that are integrated to the design tool. This approach differs from the classical reinforcement learning methods used in the dialogue literature, which make their decisions at the dialogue turn level.

We use a technique from a previous work (Laroche et al., 2009). It does not need to declare the reachable states : they are automatically created when reached. This is also a parameter-free algorithm, which is very important when we consider that most dialogue application developers are not familiar with reinforcement learning theory. We keep the developer focussed on its main task. The two additional tasks required for the reinforcement learning are to define the variable set on which the alternative choice should depend, and to implement a reward function based on the expected evaluation of the task completion, in order to get a fully automated optimisation with an online evaluation. The dialogue system automatic evaluation is a large problem that goes beyond the scope of this paper. However, sometimes, the dialogue application enables to have an explicit validation from the user. For instance, in an appointment scheduling application, the user is required to explicitly confirm the schedule he was proposed. This user performative act completes the task and provides a reliable automatic evaluation.

4.2 Learning and Monitoring Synergy in the Design Optimisation

The learning algorithm and the SDS developers are two actors on the same object : the dialogue system. But, they work at a different time space. The learning algorithm updates its policy after each dialogue while the SDS developers monitor the system behaviour more occasionally. The same kind of opposition can be made on the action space of those actors. The learning algorithm can only change its policy among a limited amount of alternatives, while the SDS developers can make deeper changes, such as implementing a new dialogue branch, adding new alternatives, new alternative points, removing alternatives, etc...

Last but not least, their sight ranges vary a lot too. The learning algorithm is concentrated on the alternative sets and automatic evaluation and ignores the rest, while the SDS developers can apprehend the dialogue application as a whole, as a system or as a service. They can also have access to additional evaluations through annotations, or user subjective evaluations.

These functionality differences make their respective roles complementary. The SDS develop-

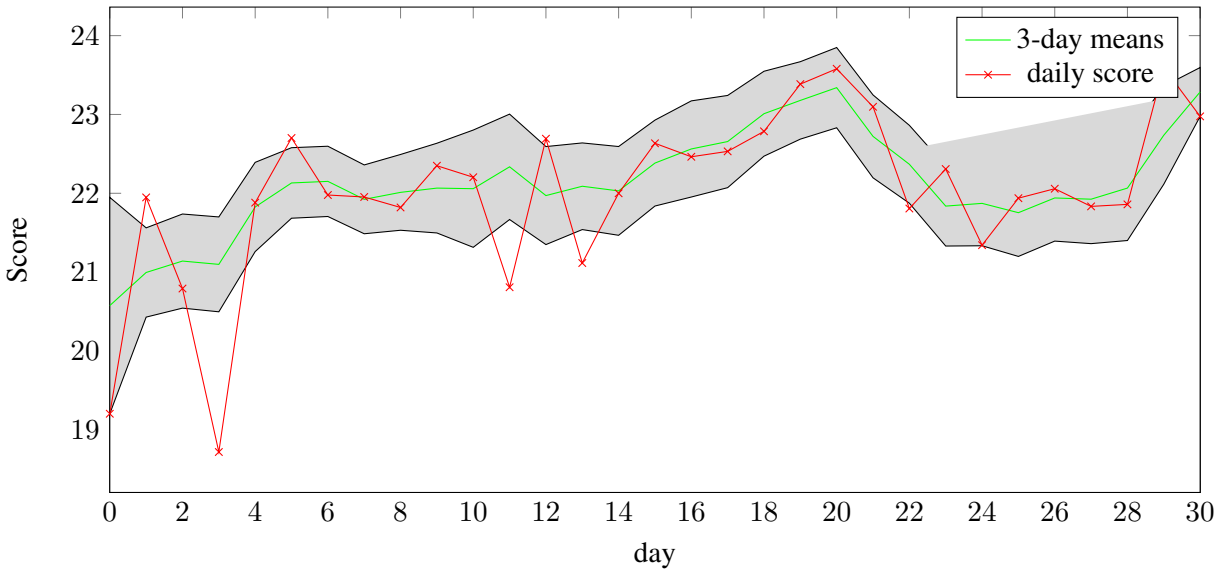


FIGURE 3 – Evolution of the system score

ers have the responsibility for the whole application and the macro-strategic changes while the learning manager holds the real-time optimisation.

4.3 Control vs. Automation : the Trusting Threshold

As argued by Pieraccini and Huerta (Pieraccini and Huerta, 2005), finite state machine applied to dialogue management does not restrict the dialogue model to strictly directed dialogues. Finite state machines are easily extensible to powerful and flexible dialogue models. Our dialogue design tool offers various extensions : dialogue modules, hierarchical design, arbitrary function invocation at any point of the design, conditional statements to split the flow in different paths. All those extensions allow designing any topology of the finite state machine required to handle complex dialogue models like mixed-initiative interaction. Dialogue model is not the point where research and industry fail to converge.

The divergence point concerns the control aspect of VUI-completeness versus the automation of the dialogue design. As pointed out by recent works (Paek and Pieraccini, 2008), MDP-based dialogue management aiming at automating the whole dialogue design is rejected by the SDS developers. Even more adaptive, it is seen as an uncontrollable black box sensitive to the tuning process. The SDS developers do not rely on systems that dynamically build their dialogue logic without a sufficient degree of monitoring and control.

Williams (Williams, 2008) has made a substantial effort to meet this industrial requirement. His system is a hybridisation of a conventional dialogue system following an industrial process, with a POMDP decision module, which is a MDP-based approach to dialogue management enhanced with dialogue state abstractions to model uncertainties. The responsibilities of each part of the system are shared as follows : the conventional system elects several candidate dialogue moves and the POMDP decision module selects the most competitive one. This is a great step towards industry because the dialogue move chosen by the POMDP module has been first controlled by the conventional system design. Nevertheless, the so-built hybrid system is still not fully compliant with the industrial constraints for the following reasons.

First, contrary to our approach, the SDS developer is called upon specific skills that cannot be demanded to a developer (modeling and tuning a (PO)MDP). This is a no-go for further integration in an industrial process.

Second, such a predictive module is not self-explanatory. Although the SDS developers have the control on the possible behaviour presented to the POMDP decision module, they are given no clue to understand how the choices are made. In fact, a learnt feature can never be exported to another context. At the opposite, our approach allows us to learn at the design level and consequently to report in the automaton the optimisation. The learning results are therefore understand-

able, analysable and replicable on a larger scale, in a way similar to classical ergonomics guidelines (but statistically proved).

4.4 Learning results on the 1013+ service

In the 1013+ service, our experiments have focused on the appointment scheduling domain. We have chosen to integrate the following rewards in the service : each time a user successfully manages to get an appointment, the system is given a $+30$ reward. If the system is unable to provide an appointment, but manages to transfer the user to a human operator, the system is given a $+10$ (a "resit"). Last, if the user hangs up, the system is not given any positive reward. Every time the system does not hear nor understand the user, it is given a penalty of 1.

In the beginning of the experiment, when the system is still using a random policy, the completion rate is as low as 51%, and the transfer rate is around 36%. When the system has learned its optimal policy, the completion rate raises up to 70%, with a transfer rate around 20%. In our experiment, the system has learned to favour an impersonal speaking style (passive mode) and it prefers proposing appointment time slots rather than asking the user to make a proposition (the later case leading to lot of "in private" user talks and hesitations, and worse recognition performance).

Figure 3 shows the evolution of the mean dialogue score during the first month. Each server have its own Learning Manager database, and optimises separately. This is a welcome feature, as each server can address a different part of the user population, which is a frequent operational requirement.

The dialogue score drawn on Figure 3 is computed by averaging the mean dialogue score per server. The crossed line represents the daily mean dialogue score. The normal line represents the 3-day smoothed dialogue mean score. The grayed area represents the 95% confidence interval. During this first month of commercial exploitation, one can notice two major trends : at first, the dialogue score is gradually increasing until day 20, then the performances noticeably drops, before rising up again. It turns out that new servers were introduced on day 20, which had to learn the optimal dialogue policy. Ultimately (on the second month), they converge to the same solution as the first servers.

5 Conclusion

5.1 A New Basis for Trusting Automatic Learning

This paper presents an original dialogue design tool that mixes dialogue design and dialogue evaluation in the same graphical interface. The design paradigm supported by the tool leads the SDS developers to predict value ranges of local KPI while designing the dialogue logic. It results a new evaluation paradigm using the system design as the reference and trying to measure deviations between the predicted and the measured values of the designed local KPI. The SDS developers rely on the tool to fulfil the VUI-completeness principle. Classically applied to dialogue design, the tool enables its application to the dialogue evaluation, leading to the comparison of dialogue design alternatives.

This places the SDS developers in a dialogue design improvement cycle close to the reinforcement learning decision process. Moreover, the inspector offered by the user experience feedback functionality allows the SDS developers to understand, analyse and generalize all the decisions among the dialogue design alternatives. Combining the learning framework and the design tool guarantees the SDS developers keep control of the system. It preserves VUI-completeness and opens the way to a reliable learning based dialogue management.

5.2 Implementation

This approach to learning led us to deploy in October 2009 the first commercial spoken dialogue system with online learning. The system's task is to schedule an appointment between the customer and a technician. This service receives approximately $8,000$ calls every month. At the time those lines are written, we are already in a virtuous circle of removing low-rated alternatives and replacing them with new ones, based on what the system learnt and what the designer understands from the data.

5.3 Future Work

On a social studies side, we are interested in collaborations to test advanced dialogue strategies and/or information presentation via generation. Indeed, we consider our system as a good opportunity for large scope experiments.

6 Acknowledgements

This research has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 216594 (CLASSIC project : www.classic-project.org).

References

A. Abella, J.H. Wright, and A.L. Gorin. 2004. Dialog trajectory analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 441–444, May.

R.E. Bellman. 1957. A markovian decision process. *Journal of Mathematics and Mechanics*, 6 :679–684.

J. Bos, E. Klein, O. Lemon, and T. Oka. 2003. Dipper : Description and formalisation of an information-state update dialogue system architecture.

George Ferguson and James F. Allen. 1998. Trips : An integrated intelligent problem-solving assistant. In *In Proc. 15th Nat. Conf. AI*, pages 567–572. AAAI Press.

R. Laroche, G. Putois, P. Bretier, and B. Bouchon-Meunier. 2009. Hybridisation of expertise and reinforcement learning in dialogue systems. In *Proceedings of Interspeech. Special Session : Machine Learning for Adaptivity in Spoken Dialogue*, Brighton (United Knigdom), September.

O. Lemon and O. Pietquin. 2007. Machine learning for spoken dialogue systems. In *Proceedings of the European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688, August.

M. F. McTear. 2004. *Spoken Dialogue Technology : Toward the Conversational User Interface*. Springer, August.

T. Paek and R. Pieraccini. 2008. Automating spoken dialogue management design using machine learning : An industry perspective. *Speech Communication*, 50 :716–729.

T. Paek. 2007. Toward evaluation that leads to best practices : Reconciling dialog evaluation in research and industry. In *Proceedings of the Workshop on Bridging the Gap : Academic and Industrial Research in Dialog Technologies*, pages 40–47, Rochester, NY, April. Association for Computational Linguistics.

R. Pieraccini and J. Huerta. 2005. Where do we go from here ? research and commercial spoken dialog systems. In Laila Dybkjaer and Wolfgang Minker, editors, *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 1–10.

R. Pieraccini, S. Caskey, K. Dayanidhi, B. Carpenter, and M. Phillips. 2001. Etude, a recursive dialog manager with embedded user interface patterns. In *Automatic Speech Recognition and Understanding, 2001 IEEE Workshop on*, pages 244–247.

M. D. Sadek, P. Bretier, and F. Panaget. 1997. Artimis : Natural dialogue meets rational agency. In *in Proceedings of IJCAI-97*, pages 1030–1035. Morgan Kaufmann.

J. D. Williams. 2008. The best of both worlds : Unifying conventional dialog systems and POMDPs. In *International Conference on Speech and Language Processing*.

Don't tell anyone! Two Experiments on Gossip Conversations

Jenny Brusk
School of informatics and humanities
University of Skövde
P.O. Box 408
541 28 Skövde, Sweden
jenny.brusk@his.se

Ron Artstein, David Traum
USC Institute for Creative Technologies
13274 Fiji Way
Marina del Rey, CA 90292
{artstein, traum}@ict.usc.edu

Abstract

The purpose of this study is to get a working definition that matches people's intuitive notion of gossip and is sufficiently precise for computational implementation. We conducted two experiments investigating what type of conversations people intuitively understand and interpret as gossip, and whether they could identify three proposed constituents of gossip conversations: third person focus, pejorative evaluation and substantiating behavior. The results show that (1) conversations are very likely to be considered gossip if all elements are present, no intimate relationships exist between the participants, and the person in focus is unambiguous. (2) Conversations that have at most one gossip element are not considered gossip. (3) Conversations that lack one or two elements or have an ambiguous element lead to inconsistent judgments.

1 Introduction

We are interested in creating believable characters, i.e. "characters that provide the illusion of life" (Bates, 1994). Since people engage extensively in gossip, such characters also need to be able to understand and engage in gossip in order to be believable in some situations. To enable characters to engage in gossip, we need a computational model of gossip that can be applied in the authoring of such characters and/or by the characters themselves. Unfortunately, such a model does not yet exist.

Moreover, there is not yet a clear consensus on how gossip should be defined, and most of the definitions are too vague or too general to be useful. Merriam-Webster online dictionary, for example, defines gossip as "rumor or report of an intimate nature" and "chatty talk", neither of which is specific enough. What we need is a working definition that (a) matches people's intuitive notion of gossip to the extent possible, given that the notion itself is somewhat vague, and (b) is sufficiently precise to provide a basis for computational implementation.

More recent definitions (e.g. Eder and Enke, 1991; Eggins and Slade, 1997; Hallett et al., 2009) have been derived from analyzing transcriptions of real gossip conversations. These definitions have only minor individual differences and can in essence be formulated as "evaluative talk about an absent third person". We have chosen to use this definition as a starting point since it currently is the most specific one and since it is based on the observed structure of naturally occurring gossip conversations.

This paper reports the results from two experiments on gossip conversations. The first experiment aimed at investigating what type of conversations people intuitively perceive as gossip. In the second study we also wanted to find out whether the subjects would accept a given definition and could apply it by identifying three specified gossip elements.

The paper is structured as follows. In section 2 we give a background to gossip with respect to both its social function as well as its conversational structure. Section 3 introduces the experimental method. In sections 4 and 5 we present the two experiments and discuss the results. In section 6, finally, we give some final remarks and suggestions for future work.

2 Background

Gossip has been described as a mechanism for social control (e.g. Gluckman, 1963; Fine and Rosnow, 1978; Bergmann, 1993; Eggins and

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 193–200,
The University of Tokyo, September 24-25, 2010.

Slade, 1997) that maintains "the unity, morals and values of social groups" (Gluckman, 1963). It has furthermore been suggested that gossip is a form of "information-management", primarily to improve one's self-image and "protect individual interests" (Paine, 1967), but also to influence others (Szwed, 1966; Fine and Rosnow, 1978). Gossip can furthermore be viewed as a form of entertainment (Abrahams, 1970) – "a satisfying diversion from the tedium of routine activities" (Fine and Rosnow, 1978:164).

Recent studies have used a sociological approach focusing on analyzing the structure of gossip conversations (e.g. Bergmann, 1993; Eder and Enke, 1991; Eggins and Slade, 1997; Hallett et al., 2009). Rather than observing and interviewing people in a certain community about their gossip behavior, they have analyzed transcripts of naturally occurring gossip conversations. Their studies show that gossipers collaborate in creating the gossip, making it a highly interactive genre. They also identified two key elements of gossip:

- **Third person focus** – the identification of an absent third person that is acquainted with, but emotionally disjoint from the other participants (Bergmann (1993) refers to this as being "virtually" absent, while Goodwin (1980) labels it "symbolically" absent).
- **An evaluation of the person in focus or of his or her behavior**. Eggins and Slade (1997) propose that the evaluation necessarily is pejorative to separate gossip from other types of chat.

Hallett et al. (2009) found that the gossipers often use implicit evaluations to conceal the critique, suggesting that the gossipers either speak in general terms about something that implicitly is understood to be about a certain person, or that the gossipers avoid evaluating the behavior under the assumption that the evaluation is implicit in the behavior itself. Instead of specifying the evaluation as being pejorative, they say it is "unsanctioned".

In addition to the two elements described above, Eggins and Slade (1997) propose a third obligatory element:

- **Substantiating behavior** – An elaboration of the deviant behavior that can either be used as a motivation for the negative evaluation, or as a way to introduce gossip in the conversation. Eder and Enke (1991) use a different model, but the substantiating behavior component corresponds roughly to their optional *Explanation* act.

There seems to be a consensus that gossip conversations have third person focus. The question is whether a gossip conversation necessarily has both a substantiating behavior component as well as a pejorative evaluation component, and if they do, can they be identified? In the experiments presented later in this paper, we hope to shed light on whether these components are necessary or not.

3 Method

During the fall 2009, we conducted two experiments about gossip conversations. The aim of the experiments was to verify to what extent the definition of gossip accords with intuitive recognition of gossip episodes, and secondly whether people could reliably identify constituent elements.

The data was collected using online questionnaires[1] that were distributed through different email-lists mainly targeting researchers and students within game design, language technology, and related fields, located primarily in North America and Europe. The questionnaires had the following structure: The first page consisted of an introduction, including instructions, and each page thereafter had a dialogue excerpt retrieved from a screenplay followed by the question and/or task.

3.1 Hypotheses

Based on the previous studies presented earlier (in particular Bergmann, 1993; Eder and Enke, 1991; and Eggins and Slade, 1997) we had the following hypotheses:

- The more gossip elements present in the text, the more likely the conversation will be considered gossip.
- Third person focus is a necessary (but not sufficient) element of gossip.
- Conversations in which the participants (including the target) are intimately related will be rated lower than those in which all participants are emotionally separated.

[1] Created using http://www.surveygizmo.com/

4 Experiment I: Identifying gossip text

The aim of the first experiment was to investigate how people intuitively understand and interpret gossip conversations.

4.1 Material and procedure

The questionnaire contained 16 different dialogue excerpts retrieved from transcripts of the famous sitcoms Desperate Housewives [2] and Seinfeld[3]. The excerpts were selected to cover different combinations of the elements presented in the previous section (third person focus, an evaluation, and a motivation for the evaluation), as in the following dialogue[4]:

B: Tisha. Tisha. Oh, I can tell by that look on your face you've got something good. Now, come on, don't be selfish.

T: Well, first off, you're not friends with Maisy Gibbons, are you?

B: No.

T: Thank god, because this is too good. Maisy was arrested. While Harold was at work, she was having sex with men in her house for money. Can you imagine?

B: No, I can't.

T: And that's not even the best part. Word is, she had a little black book with all her clients' names.

R: So, uh … you think that'll get out?

T: Of course. These things always do. Nancy, wait up. I can't wait to tell you this. Wait, wait.

A preliminary analysis to determine whether the elements were present or not, was made by the first author. The instructions contained no information about the elements and no definition was given. To each excerpt we provided some contextual information, such as the interpersonal relationship between the speakers and other people mentioned in the dialogue, e.g.:

The married couple, Bree (B) and Rex (R) Van de Kamp, is having lunch at the club. Some women laughing at the next table cause the two of them to turn and look. One of their acquaintances, Tisha (T), walks away from that table and heads to another one. Maisy Gibbons is another woman in their neighborhood, known to be very dominant and judgmental towards the other women.

[2] Touchstone Television (season 1 & 2)
[3] Castle Rock Entertainment
[4] From Desperate Housewives, Touchstone Television.

The subjects were asked to read and rank the excerpts using the following scale:

- Absolutely not gossip
- Could be considered gossip in some contexts
- Would be considered gossip in most contexts
- Absolutely gossip

For the purpose of analysis we converted the above responses to integers from 0 to 3.

4.2 Results

A total of 52 participants completed the experiment. The following table shows the distribution of ratings for each of the 16 excerpts (the table is sorted by the mean rating).

ID[5]	Rating distribution				Mean
	0	1	2	3	rating
11	50	1	1	0	0.058
6	46	5	0	1	0.154
15	33	15	4	0	0.442
2	28	20	4	0	0.538
5	30	15	6	1	0.577
10	17	24	10	1	0.904
9	10	26	13	3	1.173
16	11	17	16	8	1.404
4	8	18	18	8	1.500
14	11	13	11	17	1.654
3	6	20	11	15	1.673
1	1	17	25	9	1.808
13	3	18	17	14	1.808
12	5	9	15	23	2.077
8	3	0	11	38	2.615
7	1	2	4	45	2.788

Table 1: Gossip ratings of all 16 questions sorted by their mean value.

It is apparent from the table that a few excerpts are clearly gossip or clearly not gossip, but there is much disagreement on other excerpts. Inter-rater reliability is $\alpha = 0.437$: well above chance, but not particularly high[6]. Only 7 of the 16 excerpts (ID #2, 5, 6, 7, 8, 11, 15) were clearly rated as gossip or not gossip by more than half of the subjects, and only 5 of those have a mean rating below 0.5 or above 2.5.

[5] Presentation was ordered by ID, same for all subjects.
[6] The reported value is Krippendorff's α with the *interval distance metric* (Krippendorff 1980). Interval α is defined as $1 - D_o/D_e$, where D_o (observed disagreement) is twice the mean variance of the individual item ratings, and D_e (expected disagreement) is twice the variance of all the ratings. For the above table, $D_o = 1.327$ and $D_e = 2.585$.

Despite the apparently low agreement, the results correspond fairly well with our expectations. The 3 excerpts with a mean value below 0.5 had no gossip elements at all and the other two excerpts with a median value of 0 had only one gossip element. Similarly, the two excerpts rated highest clearly had all gossip elements. The rest of the excerpts, however, either lacked one element or had one element that was unclear in some regard (see discussion, below). Conversations between family members or partners also caused higher disagreements, which seem to support Bergmann's (1993) remark: "[…] we can ask whether we should call gossip the conversations between spouses […] alone. This surely is a borderline case for which there is no single answer" (p. 68).

4.3 Discussion

Among the nine excerpts with a mean value approximately between 1 and 2 (ID #1, 3, 4, 9, 10, 12, 13, 14, and 16), we made the following observations: 3 excerpts lacked one element; in 2 of them, the gossipers were family members or partners; 3 excerpts had an ambiguous focus, among which one also possibly was perceived as a warning.

By "ambiguous focus" we mean that it is unclear whether the person in focus is the speaker, the addressee or the absent third person. Instead, the absent third person seems to play a sub-ordinate role rather than focused role, for instance as part of a self-disclosure or a confrontation. If the conversation is the least bit confrontational, the addressee tends to go into defense rather than choosing a more typical gossip response, such as support, expansion, or challenge (Eder and Enke, 1991) in order to protect the face. Hence, no "gossip fuel" is added to the conversation.

The result of the remaining excerpt[7] is however more difficult to explain. One possible explanation is that the initiator was unacquainted with the target, but perhaps more likely is that some of the subjects interpreted the conversation as mocking rather than gossip:

E: Who's that?

D: That's Sam, the new girl in accounting.

W: What's with her arms? They just hang like salamis.

D: She walks like orangutan.

E: Better call the zoo.

[7] ID #14. From Seinfeld, Castle Rock Entertainment.

5 Experiment II: Identifying gossip elements in a text

The aim of the second experiment was to investigate whether the subjects could accept and apply a given definition by identifying the three obligatory elements of gossip according to Eggins and Slade (1997) (see section 2); *third person focus, pejorative evaluation, and substantiating behavior.* In addition to the elements, we provided the more general definition presented in section 1 (*"evaluative talk about an absent third person"*).

The results from the first experiment indicated that conversations that seemingly had all the elements but in which the person in focus was ambiguous, received a lower gossip rating than those having an unambiguous third person focus. So an additional goal was to investigate whether changing the relationship between the participants would affect the gossip rating.

5.1 Material

We used excerpts from Seinfeld[8], Desperate Housewives[9], Legally blonde[10], and Mean girls[11]. In total we selected 21 excerpts, of which 8 also occurred in the first experiment. Two of the recurring excerpts were used both in their original versions as well as in modified versions, in which we had removed the emotional connections between the participants. The purpose of this was to find out whether changing the interpersonal relationship would change the gossip rating.

5.2 Procedure

The subjects were instructed to read the excerpts and then identify the gossip elements according to the following description:

- The person being talked about (third person focus) – the "target", e.g. "Maisy Gibbons was arrested"
- Pejorative evaluation. A judgment of the target him-/herself or of the target's behavior. This evaluation is in most cases negative, e.g. "She's a slut", "He's weird"

[8] Touchstone Television.
[9] Castle Rock Entertainment.
[10] Directed by Robert Luketic. Metro Goldwyn Mayer (2001).
[11] Directed by Mark Waters. Paramount Pictures (2004).

- The deviant behavior that motivates the gossip and provides evidence for the judgment (also called the substantiating behavior stage), e.g. "Maisy Gibbons was arrested"

For each element they found, they were asked to specify the corresponding line reference as given in the text. They were also instructed to say whether they considered the conversation to be *gossip* or *not gossip*. If their rating disagreed with the definition, i.e. if they had found all the elements but still rated the conversation as not gossip, or if one or more elements were lacking but the conversation was considered gossip anyway, they were asked to specify why.

5.3 Results

We analyzed the results from the 19 subjects who completed ratings for all 21 excerpts. This gave a total of 399 yes/no judgments on 4 attributes. Inter-coder reliability[12] is shown in Table 2. The easiest attribute to interpret is third person focus. All but three of the subjects marked either 4 or 5 excerpts as not having third person focus, with the remaining subjects not deviating by much (marking 3, 6, and 7 excerpts). Moreover, the subjects agree on which excerpts have third person focus: only one excerpt gets a substantial number of conflicting ratings (see the analysis given below in section 5.4), while the remaining 20 excerpts get consistent ratings from all subjects with only occasional deviation by one or two of the deviant subjects. This accounts for the high observed agreement on this feature (94.9%). Expected agreement is high because the corpus is not balanced (16 of 21 excerpts display third person focus), but even so, chance-corrected agreement is high (85.1%), showing that third person focus is an attribute that participants can readily and reliably identify.

The remaining attributes, including gossip, are less clear. Agreement on all of them is clearly above chance, but is not particularly high, showing that these notions are either not fully defined, or that the excerpts are ambiguous. Gossip itself is identified somewhat more reliably than either substantiating behavior or pejorative evaluation; this casts doubt about the ability to use the latter two as defining features of gossip, given that they are more difficult to identify.

	Alpha	Observed agreement	Expected agreement
Gossip	0.466	0.744	0.520
Third person focus	0.851	0.949	0.661
Substantiating behavior	0.376	0.709	0.533
Pejorative evaluation	0.384	0.733	0.567

Table 2: Inter-coder reliability.

To test the relationship between the various features, we looked for co-occurrences among the individual judgments. We have a total of 399 ratings (21 excerpts times 19 judges), each with 4 attributes; these are distributed as shown in Table 3[13]. We can see that third person focus is an almost necessary condition for classifying a screenplay conversation as gossip, though it is by no means sufficient. Tables 4–6 show the co-occurrences of individual features to gossip; the association is strongest between gossip and third person focus and weakest between gossip and pejorative evaluation.

		3rd person		~~3rd person~~	
		Subst	~~Subst~~	Subst	~~Subst~~
Gossip	Pejor	168	24		2
	~~Pejor~~	33	14		
~~Gossip~~	Pejor	25	20	17	17
	~~Pejor~~	6	23	3	47

Table 3: Relationship between the different elements and gossip.

	3rd person	~~3rd person~~
Gossip	239	2
~~Gossip~~	74	84

Table 4: Gossip – third person focus.

	Substantiating behavior	~~Substantiating behavior~~
Gossip	201	40
~~Gossip~~	51	107

Table 5: Gossip – substantiating behavior

[12] We used Krippendorff's alpha with the *nominal distance metric*. Observed agreement is defined as $A_o = 1 - D_o$, while expected agreement is: $A_e = 1 - D_e$.

[13] Strike-through marks the absence of a feature.

	Pejorative	~~Pejorative~~
Gossip	194	47
~~Gossip~~	79	79

Table 6: Gossip – pejorative evaluation

In addition to the co-occurrences of features on the individual judgments, we can look at these co-occurrences grouped by screenplay. Table 7 shows, for each of the 21 excerpts, how many subjects identified each of the four features (the table is sorted by the gossip score). It is apparent from the table that all the features are correlated to some extent.

ID[14]	Gossip	Third person	Subst. behavior	Pejorative evaluation
2	0	0	1	3
11	0	0	9	9
19	0	1	6	8
14	1	0	2	12
5	7	19	5	1
15	7	19	18	17
21	8	17	6	16
12	9	17	10	14
20	13	13	10	10
16	14	18	14	7
8	14	19	7	19
7	14	19	9	9
17	14	19	17	18
18	15	19	19	19
4	17	19	12	9
10	17	19	16	19
6	17	19	19	19
9	18	19	17	8
1	18	19	19	19
3	19	19	18	18
13	19	19	18	19

Table 7: Co-occurrences grouped by excerpts.

Table 8 shows the correlation between gossip and each of the other three features. The first column calculates correlation based on the individual judgments (399 items, each score is either 0 or 1); the second column calculates correlation based on the rated excerpts (21 items, each score is an integer between 0 and 19, as in table 7); and the third column groups the judgments by subject (19 items, each score is an integer between 0 and 21, indicating the number of dialogues in which the subject identified the particular feature; the full data are not shown).

[14] Presentation was ordered by ID, same for all subjects.

Correlation with gossip	Pearson's r Individual	Excerpt	Subject
Third person	0.622***	0.849***	0.503*
Substantiating	0.518***	0.765***	0.625**
Pejorative	0.321***	0.518*	0.459*

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table 8: Correlation between gossip and each of the three features.

All the correlations are significantly different from 0 at the $p \leq 0.05$ level or greater. The differences between the columns are not significant, except for the difference between the third person correlation by individuals and that by excerpt, which is significant at $p \leq 0.05$. The correlations between the features on the individual judgments show that subjects tend to identify the different features together; this may be partly a reflection of awareness on their part that the features are expected to go together, given the task definition. The correlations between the excerpt scores show that the excerpts themselves differ along the four dimensions, and these differences go hand in hand. Finally, we see that the subjects themselves differ in how often they identify the different features, though the correlations are likely to be just a reflection of the first tendency identified above, to mark the features together.

5.4 Discussion

We wanted to find out whether the subjects would accept, understand and be able to apply a given definition. The results from the experiment showed that the subjects accepted the given definition to some extent and managed to apply it. When the subjects disagreed they were asked to say why. One of the subjects, for example, explicitly disagreed with the definition given in the introduction and provided a counter definition: "Gossip is idle talk or rumor, especially about the personal or private affairs of others". Yet another subject was uncertain about which definition to use: "Depends what you mean by gossip. It can either mean malicious, behind the back talk of other people or idle chat. If you mean 'idle chat' with gossip then this is also gossip". A possible explanation could be that the subjects refer to different forms of gossip (see e.g. Gilmore, 1978) and therefore apply different definitions (such as the lexical definition presented earlier) than the one that was given in the experiment.

Several subjects stated that they judged the conversation as gossip even if they did not identify any pejorative evaluation, and they also questioned whether the evaluation had to be negative or even present at all, or as one of the subjects put it: "Although there is no pejorative evaluation (at least not clearly) I believe this is gossip". These subjects thus explicitly reject Eggins and Slade's (1997) requirement that the evaluation has to be pejorative.

The examples above show that people have variable intuitions of gossip and consequently the concept of gossip is somewhat vague. Even so, the experiment also showed that people to a large degree are in agreement when the examples according to the given definition clearly are gossip or not gossip. Meaning that even though the definition does not capture all types of (potential) gossip conversations, it captures those episodes that most people agree to be gossip, which for our purpose is sufficient.

5.5 Effect of interpersonal relations

In some particular cases, the subjects did not choose gossip even if all elements had been found. The results from the first experiment indicated that this deviation either was related to the interpersonal relationship between the gossip participants or that the focus was ambiguous. In order to test whether changing the inter-personal relationship between the participants would change the gossip rating, we compared the results from the conversations we had modified with their original counterparts. In one of the original excerpts, the addressee was romantically involved with the man that the speaker was talking about. The speaker formulated the negative assessment and deviant behavior in a way that for most people would be interpreted as a warning, which probably explains why only 7 of the 19 subjects rated the original conversation as gossip. The modified version on the other hand, was rated as gossip by all subjects.

In the second dialogue, the speaker questions the addressee's choice of person to date, and does this by both evaluating the person negatively as well as providing evidence for the evaluation. It turns out, however, that the addressee thinks she is going out for a date with someone else, so a large part of the conversation deals with trying to identify the target. 15 of 19 subjects rated the original conversation as gossip, while all subjects rated it as gossip in the modified version. These comparisons indicate that the status of the relationship between the gossipers and the gossip target affects whether the dialogue is considered gossip or not. In the original version of both these examples, the focus was ambiguous, i.e. the focus was as much on the addressee as on the absent third person.

We have shown that third person focus is a key element of gossip. The correlation was furthermore confirmed by the subjects themselves in their comments, where the lack of third person often was listed as a reason for not choosing gossip. In one example, the respondent regarded the conversation as gossip even if it really was an insult directed towards the addressee, but explained it as its "…almost like he's forgotten he's talking to the person he's giving this opinion/gossip about".

The highest disagreement concerning third person focus was found in the following excerpt[15]:

Karen: Okay, what is it?

Gretchen: Regina says everyone hates you because you're such a slut.

Karen: She said that?

Gretchen: You didn't hear it from me.

The dialogue contains an ambiguous focus in that it both includes a quote as well as a confrontational insult. By using the third person reference, Gretchen avoids taking responsibility for the insult. In some sense both Karen and Regina are in focus, where Karen is the target of the pejorative evaluation and Regina can be interpreted as being the focus of the substantiating behavior component. How Regina's role is interpreted is determined by the respondents' personal attitude towards gossiping in general (i.e. whether they interpret Gretchen's utterance as containing an implicit evaluation of Regina's behavior or not), and how they perceive the interpersonal relationship between Karen and Gretchen. Gossip has an inherent contradiction in that it both has a function of negotiating the accepted way to behave while it at the same time often is considered an inappropriate activity that can have serious negative consequences for both the gossipers as well as the gossip target (see e.g. Gilmore, 1978; Bergmann, 1993; Eggins and Slade, 1997; Hallett et al., 2009).

[15] From Mean Girls, Paramount Pictures, 2004.

6 Final remarks and future work

The aim of these studies has been to get a workable definition of gossip that people can agree upon and that is sufficiently precise to provide a basis for computational implementation.

We conducted two experiments to investigate people's intuitive notion of gossip and the results show that (1) conversations in which all elements are present, where no intimate relationships exist between the participants, and in which the person in focus is unambiguous, are very likely to be considered gossip. (2) Conversations that have at most one gossip element are not considered gossip. (3) Inconsistencies are mainly found in conversations that lack one or two elements or have at least one element that is ambiguous, or are taking place between gossipers that have an intimate relationship.

We have suggested that third person focus is a necessary, but not sufficient, element of gossip, but the other elements are less clear even if their co-occurrence in a conversation clearly affects the gossip score. In the second experiment this might be due to the instructions, but it does not explain the unbiased results from the first experiment. So on the one hand we can clearly see that all three elements are important for the understanding of gossip, but on the other hand, the subjects' had trouble in identifying them. This suggests that we need to further investigate these elements to see how they can be specified more clearly.

We have taken a first step toward a computational account of gossip, by empirically verifying the extent to which the given definition can be applied and the components recognized by people. Some of our next steps to further this program include authoring content for believable characters that follow this definition, as well as attempting to automatically recognize these elements.

Among the possible applications of gossip we can think of game characters and virtual humans that are capable of engaging in gossip conversations to share information and create social bonds with a human user or its avatar. This involves being able to both generate gossip on basis of the interpersonal relationship and selecting content that could be regarded as gossip, as well as to automatically detect gossip occurring in a conversation. The latter use could also be used for characters that actively want to avoid taking part in gossip conversations.

References

Roger D. Abrahams. 1970. A Performance-Centred Approach to Gossip. *Man,* New Series, Vol. 5, No. 2 (Jun.), pp. 290-301.

Joseph Bates. (1994. The Role of Emotion in Believable agents. *Communications of the ACM*, Vol. 37, No. 7 (Jul.), pp. 122-125.

Jörg R. Bergmann. 1993. *Discreet Indiscretions: The Social Organization of Gossip*. New York: Aldine. Suzanne Eggins and Diana Slade (1997) *Analysing Casual Conversation*. Equinox Publishing Ltd.

Donna Eder and Janet Lynne Enke (1991) The Structure of Gossip: Opportunities and Constraints on Collective Expression among Adolescents. *American sociological Review*, Vol. 56, No. 4 (Aug.), pp. 494–508.

Gary Alan Fine and Ralph L. Rosnow. 1978. Gossip, Gossipers, Gossiping. *Personality and Social Psychology Bulletin*, Vol. 4, No. 1, pp 161-168.

David Gilmore. 1978. Varieties of Gossip in a Spanish Rural Community. *Ethnology*, Vol. 17, No. 1 (Jan.), pp. 89-99.

Max Gluckman. 1963. Papers in Honor of Melville J. Herskovits: Gossip and Scandal. *Current Anthropology*, Vol. 4, No. 3 (Jun), pp. 307–316.

Marjorie Harness Goodwin. 1980. He-Said-She-Said: Formal Cultural Procedures for the Construction of a Gossip Dispute Activity. *American Ethnologist*, Vol. 7, No. 4 (Nov.), pp. 674-695.

Tim Hallett., Brent Harget and Donna Eder (2009) Gossip at Work: Unsanctioned Evaluative Talk in Formal School Meetings. *Journal of Contemporary Ethnography*, Vol. 38, No. 5, pp 584–618.

Klaus Krippendorff. 1980. *Content Analysis: An Introduction to Its Methodology*, chapter 12. Sage Beverly Hills, CA.

Robert Paine.1967. What is gossip about? An alternative hypothesis. *Man*, New Series, Vol. 2, No. 2 (Jun.), pp. 278-285.

John F. Szwed. 1966. Gossip, Drinking, and Social Control: Consensus and Communication in a Newfoundland Parish. *Ethnology*, Vol. 5, No. 4 (Oct.), pp. 434-441.

Gaussian Processes for Fast Policy Optimisation of POMDP-based Dialogue Managers

M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu and S. Young
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK
{mg436, fj228, sk561, farm2, brmt2, ky219, sjy}@eng.cam.ac.uk

Abstract

Modelling dialogue as a Partially Observable Markov Decision Process (POMDP) enables a dialogue policy robust to speech understanding errors to be learnt. However, a major challenge in POMDP policy learning is to maintain tractability, so the use of approximation is inevitable. We propose applying Gaussian Processes in Reinforcement learning of optimal POMDP dialogue policies, in order (1) to make the learning process faster and (2) to obtain an estimate of the uncertainty of the approximation. We first demonstrate the idea on a simple voice mail dialogue task and then apply this method to a real-world tourist information dialogue task.

1 Introduction

One of the main challenges in dialogue management is effective handling of speech understanding errors. Instead of hand-crafting the error handler for each dialogue step, statistical approaches allow the optimal dialogue manager behaviour to be learnt automatically. Reinforcement learning (RL), in particular, enables the notion of planning to be embedded in the dialogue management criteria. The objective of the dialogue manager is for each dialogue state to choose such an action that leads to the highest expected long-term reward, which is defined in this framework by the Q-function. This is in contrast to Supervised learning, which estimates a dialogue strategy in such a way as to make it resemble the behaviour from a given corpus, but without directly optimising overall dialogue success.

Modelling dialogue as a Partially Observable Markov Decision Process (POMDP) allows action selection to be based on the differing levels of uncertainty in each dialogue state as well as the overall reward. This approach requires that a distribution of states (*belief state*) is maintained at each turn. This explicit representation of uncertainty in the POMDP gives it the potential to produce more robust dialogue policies (Young et al., 2010).

The main challenge in the POMDP approach is the tractability of the learning process. A discrete state space POMDP can be perceived as a continuous space MDP where the state space consists of the belief states of the original POMDP. A grid-based approach to policy optimisation assumes discretisation of this space, allowing for discrete space MDP algorithms to be used for learning (Brafman, 1997) and thus approximating the optimal Q-function. Such an approach takes the order of $100,000$ dialogues to train a real-world dialogue manager. Therefore, the training normally takes place in interaction with a simulated user, rather than real users. This raises questions regarding the quality of the approximation as well as the potential discrepancy between simulated and real user behaviour.

Gaussian Processes have been successfully used in Reinforcement learning for continuous space MDPs, for both model-free approaches (Engel et al., 2005) and model-based approaches (Deisenroth et al., 2009). We propose using GP Reinforcement learning in a POMDP dialogue manager to, firstly, speed up the learning process and, secondly, obtain the uncertainty of the approximation. We opt for the model-free approach since it has the potential to allow the policy obtained in interaction with the simulated user to be further refined in interaction with real users.

In the next section, the core idea of the method is explained on a toy dialogue problem where different aspects of GP learning are examined. Following that, in Section 3, it is demonstrated how this methodology can be effectively applied to a real world dialogue. We conclude with Section 4.

2 Gaussian Process RL on a Toy Problem

2.1 Gaussian Process RL

A Gaussian Process is a generative model of Bayesian inference that can be used for function regression (Rasmussen and Williams, 2005). A Gaussian Process is fully defined by a mean and a kernel function. The kernel function defines prior function correlations, which is crucial for obtaining good posterior estimates with just a few observations. GP-Sarsa is an on-line reinforcement learning algorithm for both continuous and discrete MDPs that incorporates GP regression (En-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 201–204,
The University of Tokyo, September 24-25, 2010.

gel et al., 2005). Given the observation of rewards, it estimates the Q-function utilising its correlations in different parts of the state and the action space defined by the kernel function. It also gives a variance of the estimate, thus modelling the uncertainty of the approximation.

2.2 Voice Mail Dialogue Task

In order to demonstrate how this methodology can be applied to a dialogue system, we first explain the idea on the voice mail dialogue problem (Williams, 2006).

The state space of this task consists of three states: the user asked for the message either to be saved or deleted, or the dialogue ended. The system can take three actions: ask the user what to do, save or delete the message. The observation of what the user wants is corrupted with noise, therefore we model this as a three-state POMDP. This POMDP can be viewed as a continuous MDP, where the MDP state is the POMDP belief state, a 3-dimensional vector of probabilities. For both learning and evaluation, a simulated user is used which makes an error with probability 0.3 and terminates the dialogue after at most 10 turns. In the final state, it gives a positive reward of 10 or a penalty of -100 depending on whether the system performed a correct action or not. Each intermediate state receives the penalty of -1. In order to keep the problem simple, a model defining transition and observation probabilities is assumed so that the belief can be easily updated, but the policy optimisation is performed in an on-line fashion.

2.3 Kernel Choice for GP-Sarsa

The choice of kernel function is very important since it defines the prior knowledge about the Q-function correlations. They have to be defined on both states and actions. In the voice mail dialogue problem the action space is discrete, so we opt for a simple δ kernel over actions:

$$k(a, a') = 1 - \delta_a(a'), \qquad (1)$$

where δ_a is the Kronecker delta function. The state space is a 3-dimensional continuous space and the kernel functions over the state space that we explore are given in Table 1. Each kernel function defines a different correlation. The polynomial kernel views elements of the state vector as features, the dot-product of which defines the correlation. They can be given different relevance r_i in the parametrised version. The Gaussian kernel accounts for smoothness, *i.e.*, if two states are close to each other the Q-function in these states is correlated. The scaled norm kernel defines positive correlations in the points that are close to each other and a negative correlation otherwise. This is particularly useful for the voice mail problem, where, if two belief states are very different, taking the same action in these states generates a negatively correlated reward.

kernel function	expression
polynomial	$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$
parametrised poly.	$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{D} \frac{x_i x_i'}{r_i^2}$
Gaussian	$k(\mathbf{x}, \mathbf{x}') = p^2 \exp \frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma_k^2}$
scaled norm	$k(\mathbf{x}, \mathbf{x}') = 1 - \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\|\mathbf{x}\|^2 \|\mathbf{x}'\|^2}$

Table 1: Kernel functions

2.4 Optimisation of Kernel Parameters

Some kernel functions are in a parametrised form, such as Gaussian or parametrised polynomial kernel. These parameters, also called *the hyper-parameters*, are estimated by maximising the marginal likelihood[1] on a given corpus (Rasmussen and Williams, 2005). We adapted the available code (Rasmussen and Williams, 2005) for the Reinforcement learning framework to obtain the optimal hyper-parameters using a dialogue corpus labelled with states, actions and rewards.

2.5 Grid-based RL Algorithms

To assess the performance of GP-Sarsa, it was compared with a standard grid-based algorithm used in (Young et al., 2010). The grid-based approach discretises the continuous space into regions with their representative points. This then allows discrete MDP algorithms to be used for policy optimisation, in this case the Monte Carlo Control (MCC) algorithm (Sutton and Barto, 1998).

2.6 Optimal POMDP Policy

The optimal POMDP policy was obtained using the POMDP solver toolkit (Cassandra, 2005), which implements the Point Based Value Iteration algorithm to solve the POMDP off-line using the underlying transition and observation probabilities. We used 300 sample dialogues between the dialogue manager governed by this policy and the simulated user as data for optimisation of the kernel hyper-parameters (see Section 2.4).

2.7 Training set-up and Evaluation

The dialogue manager was trained in interaction with the simulated user and the performance was compared between the grid-based MCC algorithm and GP-Sarsa across different kernel functions from Table 1.

The intention was, not only to test which algorithm yields the best policy performance, but also to examine the speed of convergence to the optimal policy. All the algorithms use an ϵ-greedy approach where the exploration rate ϵ was fixed at 0.1. The learning process greatly depends on

[1] Also called *evidence maximisation* in the literature.

the actions that are taken during exploration. If early on during the training, the systems discovers a path that generates high rewards due to a lucky choice of actions, then the convergence is faster. To alleviate this, we adopted the following procedure. For every training set-up, exactly the same training iterations were performed using 1000 different random generator seedings. After every 20 dialogues the resulting 1000 partially optimised policies were evaluated. Each of them was tested on 1000 dialogues. The average reward of these 1000 dialogues provides just one point in Fig. 1.

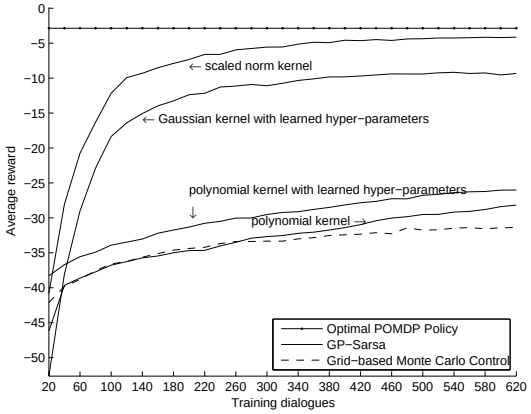


Figure 1: Evaluation results on Voice Mail task

The grid-based MCC algorithm used a Euclidian distance to generate the grid by adding every point that was further than 0.01 from other points as a representative of a new region. As can be seen from Fig 1, the grid-Based MCC algorithm has a relatively slow convergence rate. GP-Sarsa with the polynomial kernel exhibited a learning rate similar to MCC in the first 300 training dialogues, continuing with a more upward learning trend. The parametrised polynomial kernel performs slightly better. The Gaussian kernel, however, achieves a much faster learning rate. The scaled norm kernel achieved close to optimal performance in 400 dialogues, with a much higher convergence rate then the other methods.

3 Gaussian Process RL on a Real-world Task

3.1 HIS Dialogue Manager on CamInfo Domain

We investigate the use of GP-Sarsa in a real-world task by extending the Hidden Information State (HIS) dialogue manager (Young et al., 2010). The application domain is tourist information for Cambridge, whereby the user can ask for information about a restaurant, hotel, museum or another tourist attraction in the local area. The database consists of more than 400 entities each of which has up to 10 attributes that the user can query.
The HIS dialogue manager is a POMDP-based dialogue manager that can tractably maintain belief states for large domains. The key feature of this approach is the grouping of possible user goals into *partitions*, using relationships between different attributes from possible user goals. Partitions are combined with possible user dialogue actions from the N-best user input as well as with the dialogue history. This combination forms the state space – the set of *hypotheses*, the probability distribution over which is maintained during the dialogue. Since the number of states for any real-world problem is too large, for tractable policy learning, both the state and the action space are mapped into smaller scale summary spaces. Once an adequate summary action is found in the summary space, it is mapped back to form an action in the original *master space*.

3.2 Kernel Choice for GP-Sarsa

The summary state in the HIS system is a four-dimensional space consisting of two elements that are continuous (the probability of the top two hypotheses) and two discrete elements (one relating the portion of the database entries that matches the top partition and the other relating to the last user action type). The summary action space is discrete and consists of eleven elements.
In order to apply the GP-Sarsa algorithm, a kernel function needs to be specified for both the summary state space and the summary action space. The nature of this space is quite different from the one described in the toy problem. Therefore, applying a kernel that has negative correlations, such as the scaled norm kernel (Table 1) might give unexpected results. More specifically, for a given summary action, the mapping procedure finds the most appropriate action to perform if such an action exists. This can lead to a lower reward if the summary action is not adequate but would rarely lead to negatively correlated rewards. Also, parametrised kernels could not be used for this task, since there was no corpus available for hyper-parameter optimisation. The polynomial kernel (Table 1) assumes that the elements of the space are features. Due to the way the probability is maintained over this very large state space, the continuous variables potentially encode more information than in the simple toy problem. Therefore, we used the polynomial kernel for the continuous elements. For discrete elements, we utilise the δ-kernel (Eq. 2.3).

3.3 Active Learning GP-Sarsa

The GP RL framework enables modelling the uncertainty of the approximation. The uncertainty estimate can be used to decide which actions to take during the exploration (Deisenroth et al.,

2009). In detail, instead of a random action, the action in which the Q-function for the current state has the highest variance is taken.

3.4 Training Set-up and Evaluation

Policy optimisation is performed by interacting with a simulated user on the dialogue act level. The simulated user gives a reward at the final state of the dialogue, and that is 20 if the dialogue was successful, 0 otherwise, less the number of turns taken to fulfil the user goal. The simulated user takes a maximum of 100 turns in each dialogue, terminating it when all the necessary information has been obtained or if it looses patience.
A grid-based MCC algorithm provides the baseline method. The distance metric used ensures that the number of regions in the grid is small enough for the learning to be tractable (Young et al., 2010).
In order to measure how fast each algorithm learns, a similar training set-up to the one presented in Section 2.7 was adopted and the averaged results are plotted on the graph, Fig. 2.

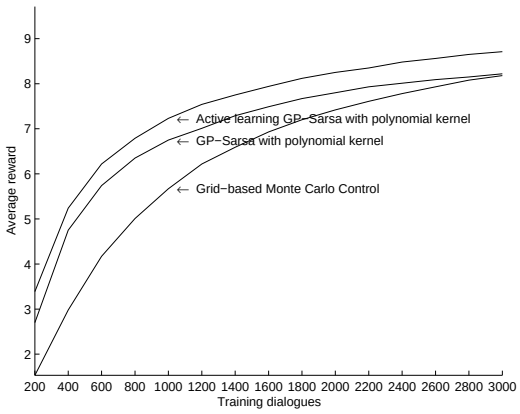


Figure 2: Evaluation results on CamInfo task

The results show that in the very early stage of learning, *i.e.*, during the first 400 dialogues, the GP-based method learns faster. Also, the learning process can be accelerated by adopting the active learning framework where the actions are selected based on the estimated uncertainty.
After performing many iterations in an incremental noise learning set-up (Young et al., 2010) both the GP-Sarsa and the grid-based MCC algorithms converge to the same performance.

4 Conclusions

This paper has described how Gaussian Processes in Reinforcement learning can be successfully applied to dialogue management. We implemented a GP-Sarsa algorithm on a toy dialogue problem, showing that with an appropriate kernel function faster convergence can be achieved. We also demonstrated how kernel parameters can be learnt from a dialogue corpus, thus creating a bridge between Supervised and Reinforcement learning methods in dialogue management. We applied GP-Sarsa to a real-world dialogue task showing that, on average, this method can learn faster than a grid-based algorithm. We also showed that the variance that GP is estimating can be used in an Active learning setting to further accelerate policy optimisation.
Further research is needed in the area of kernel function selection. The results here suggest that the GP framework can facilitate faster learning, which potentially allows the use of larger summary spaces. In addition, being able to learn efficiently from a small number of dialogues offers the potential for learning from direct interaction with real users.

Acknowledgements

The authors would like to thank Carl Rasmussen for valuable discussions. This research was partly funded by the UK EPSRC under grant agreement EP/F013930/1 and by the EU FP7 Programme under grant agreement 216594 (CLASSiC project).

References

RI Brafman. 1997. A Heuristic Variable Grid Solution Method for POMDPs. In *AAAI*, Cambridge, MA.

AR Cassandra. 2005. POMDP solver. `http://www.cassandra.org/pomdp/code/index.shtml`.

MP Deisenroth, CE Rasmussen, and J Peters. 2009. Gaussian Process Dynamic Programming. *Neurocomput.*, 72(7-9):1508–1524.

Y Engel, S Mannor, and R Meir. 2005. Reinforcement learning with Gaussian processes. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 201–208, New York, NY.

CE Rasmussen and CKI Williams. 2005. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA.

RS Sutton and AG Barto. 1998. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA.

JD Williams. 2006. *Partially Observable Markov Decision Processes for Spoken Dialogue Management*. Ph.D. thesis, University of Cambridge.

SJ Young, M Gašić, S Keizer, F Mairesse, J Schatzmann, B Thomson, and K Yu. 2010. The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.

Coherent Back-Channel Feedback Tagging of In-Car Spoken Dialogue Corpus

Yuki Kamiya
Graduate School of Information Science, Nagoya University, Japan
kamiya@el.itc.nagoya-u.ac.jp

Tomohiro Ohno
Graduate School of International Development, Nagoya University, Japan
ohno@nagoya-u.jp

Shigeki Matsubara
Graduate School of Information Science, Nagoya University, Japan
matubara@nagoya-u.jp

Abstract

This paper describes the design of a back-channel feedback corpus and its evaluation, aiming at realizing in-car spoken dialogue systems with high responsiveness. We constructed our corpus by annotating the existing in-car spoken dialogue data with back-channel feedback timing information in an off-line environment. Our corpus can be practically used in developing dialogue systems which can provide verbal back-channel feedbacks. As the results of our evaluation, we confirmed that our proposed design enabled the construction of back-channel feedback corpora with high coherency and naturalness.

1 Introduction

In-car spoken dialogue processing is one of the most prevailing applications of speech technology. Until now, to realize the system which can surely achieve such tasks navigation and information retrieval, the development of speech recognition, speech understanding, dialogue control and so on has been promoted. Now, it becomes important to increase responsiveness of the system not only for the efficient achievement of the task but for increasing drivers' comfortableness in a dialogue.

One way to increase responsiveness of a system is to timely disclose system's state of understanding, by making the system show some kind of reaction during user's utterances. In human dialogues, such disclosure is performed by actions such as nods, facial expressions, gestures and back-channel feedbacks. However, since drivers do not look towards a spoken dialogue system while driving, the system has to inevitably use voice responses, that is, back-channel feedbacks. Furthermore, in the response strategy for realizing in-car dialogues in which drivers feel comfortable, it is necessary for the system to provide back-channel feedbacks during driver's utterances aggressively as well as timely.

This paper describes the design of a back-channel feedback corpus having coherency (tagging is performed by different annotators equally) and naturalness, and its evaluation, aiming at realizing in-car spoken dialogue systems with high responsiveness. Although there have been several researches on back-channel feedback timings (Cathcart et al., 2003; Maynard, 1989; Takeuchi et al., 2004; Ward and Tsukahara, 2000), in many of them, back-channel feedback timings in human dialogues were observed and analyzed by using a general spoken dialogue corpus. On the other hand, we constructed our corpus by annotating the existing in-car spoken dialogue data with back-channel feedback timing information in an off-line environment. Our corpus can be practically used in developing dialogue systems which can provide back-channel feedbacks.

In our research, the driver utterances (11,181 turns) in the CIAIR in-car spoken dialogue corpus (Kawaguchi et al., 2005) were used as the existing data. We created the Web interface for the annotation of back-channel feedbacks and constructed the corpus including 5,416 back-channel feedbacks. Experiments have shown that our proposed corpus design enabled the construction of back-channel feedback corpora with high coherency and naturalness.

2 Corpus Design

A back-channel feedback is a sign to inform a speaker that the listener received the speaker's utterances. Thus, in an in-car dialogue between a driver and a system, it is preferable that the system provides as many back-channel feedbacks as possible. However, if back-channel feedbacks are unnecessarily provided, they can not play the primary role because the driver wonders if the system really comprehends the speech.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 205–208,
The University of Tokyo, September 24-25, 2010.

For this reason, the timings at which the system provides back-channel feedbacks become important. Several researches investigated back-channel feedback timings in human-human dialogues (Cathcart et al., 2003; Maynard, 1989; Takeuchi et al., 2004; Ward and Tsukahara, 2000). They reported back-channel feedbacks had the following tendencies: "within or after a pause," "after a conjunction or sentence-final particle," and "after a clause wherein the final pitch descends."

However, it is difficult to systematize the appropriate timings of back-channel feedbacks since their detection is intertwined in a complex way with various acoustic and linguistic factors. Although machine learning using large-scale data would be a solution to the problem, existing spoken dialogue corpora are not suitable for direct use as data, because the timings of the back-channel feedbacks lack coherency due to the influence of factors such as the psychological state of a speaker, the environment and so on.

In our research, to create more pragmatic data in which the above-mentioned problem is solved, we constructed the back-channel feedback corpus with coherency. To this end, we established the following policies for annotation:

- **Comprehensive tagging:** Back-channel feedback tags are provided for all timings which are not unnatural. In human-human dialogues, there are some cases that even if a timing is suited for providing a back-channel feedback, no back-channel feedback is not provided (Ward and Tsukahara, 2000). On the other hand, in our corpus, comprehensive tagging enables coherent tagging.
- **Off-line tagging:** Annotators tag all timings at which back-channel feedbacks can be provided after listening to the target speech one or more times. Compared with providing back-channel feedbacks in on-line environment, the off-line annotation decreases the chances of tagging wrong positions or failing in tagging back-channel feedbacks, realizing coherent tagging.
- **Discretization of tagging points:** Tagging is performed for each segment into which driver's utterances are divided. In a normal dialogue, the listener can provide back-channel feedbacks whenever he/she wants to, but the inconsistency in the timings to give such feedbacks becomes larger in exchange for smaller restrictions. The discretization of tagging points enables not only coherent tagging but also the reduction of tagging cost.
- **Elaboration using synthesized sound:** An annotator checks the validity of the annotation by listening to the sounds. In other words, an annotator elaborates the annotation by revising it many times by listening to the automatically created dialogue sound which includes not only driver's voices but also sounds of back-channel feedbacks generated according to the provided timings. The back-channel feedbacks had been synthesized by using a speech synthesizer because our corpus aims to be used for implementing the system which can provide back-channel feedbacks.

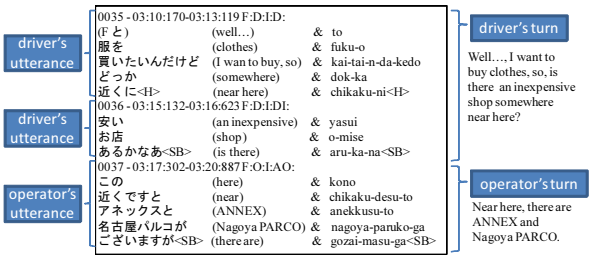


Figure 1: Sample of transcribed text

3 Corpus Construction

We constructed the back-channel feedback corpus by annotating an in-car speech dialogue corpus.

3.1 CIAIR in-car spoken dialogue corpus

We used the CIAIR in-car spoken dialogue corpus (Kawaguchi et al., 2005) as the target of annotation. The corpus consists of the speech and transcription data of dialogues between a driver and an operator about shopping guides, driving directions, and so on. Figure 1 shows an example of the transcription. We used only the utterances of drivers in the corpus. We divided the utterances into morphemes by using the morphological analyzer Chasen[1]. In addition, each morpheme was provided start and end times estimated by using the continuous speech recognition system Julius[2].

3.2 Tagging of spoken dialogue corpus

We constructed the corpus by providing the back-channel feedback tags at the proper timings for the driver's utterances, according to the design described in Section 2.

[1]http://chasen-legacy.sourceforge.jp
[2]http://julius.sourceforge.jp

	content	start time	end time
sp	[short pause]	0.000	0.030
(Fと)	**(Well...)**	0.030	0.090
服	**(clothes)**	0.090	0.340
を	(no translation)	0.340	0.520
sp	[short pause]	0.520	0.610
買い	**(buy)**	0.610	0.850
たい	**(want to)**	0.850	1.080
ん	(no translation)	1.080	1.150
だ	(no translation)	1.150	1.240
けど	**(so)**	1.240	1.420
どっ	**(somewhere)**	1.420	1.670
か	(no translation)	1.670	1.850
近く	**(near hear)**	1.850	2.190
に	(no translation)	2.190	2.880
sp	[short pause]	2.880	3.080
pause	[pause]	3.080	4.992
安い	**(inexpensive)**	4.992	5.362
お	(no translation)	5.362	5.422
店	**(shop)**	5.422	5.652
ある	**(is there)**	5.652	5.832
か	(no translation)	5.832	5.982
なあ	(no translation)	5.982	6.272

Figure 2: Sample of division of a dialogue turn into basic segments

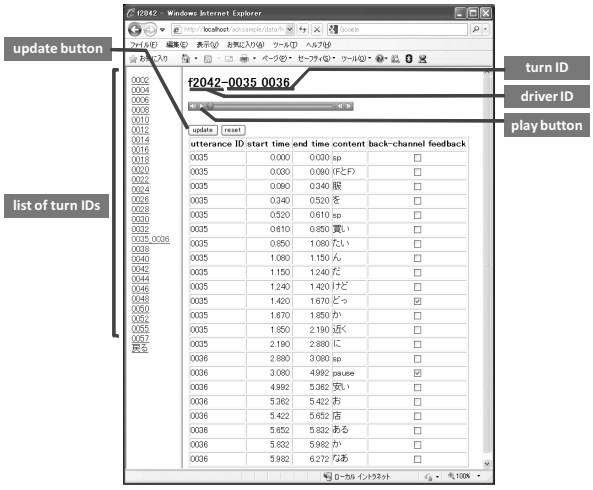


Figure 3: Web interface for tagging

Table 1: Size of back-channel feedback corpus

drivers	346
dialogue turns	11,181
clauses	16,896
bunsetsus[4]	12,689
morpheme segments	94,030
pause segments	19,142
back-channel feedbacks	5,416

For "comprehensive tagging," an annotator listens to each dialogue turn[3] from the start and tags a position where a back-channel feedback can be provided when the timing is found. Here, the timing of the last back-channel feedback is also used for judging whether or not the timing is unnatural.

For "off-line tagging," an annotator tags the transcribed text of each dialogue turn of drivers.

To perform "discretization of tagging points," a dialogue turn is assumed to be a sequence of morphemes or pauses (hereafter, we call them **basic segments**), which are continuously arranged on the time axis, and it is judged whether or not a back-channel feedback should be provided at each basic segment. Here, in consideration of the unequal pause durations, if the length of a pause is over 200ms, the pause is divided into the initial 200ms pause and the subsequent pause, each of which is considered as a basic segment. Figure 2 shows an example of a dialogue turn divided into basic segments.

Furthermore, for "elaboration using synthesized sound," we prepared the annotation environment where the dialogue sound including not only driver's voice but also back-channel feedbacks generated according to the provided timings is automatically created in real time for annotators to listen to. There are several types of back-channel feedbacks and in normal conversations, we choose and use appropriate back-channel feedbacks from among them according to the scene. In our study, we used the most general form "はい *hai* (yes)" for the synthesized speech since our focus was on the timing of back-channel feedbacks. The back-channel feedbacks had been created by using Hitachi's speech synthesizer "HitVoice," and one feedback was placed 50 milli-seconds after the start time of a tagged basic segment.

We developed a Web interface for tagging back-channel feedbacks. Figure 3 shows the Web interface. The interface displays a sequence of basic segments in a dialogue turn in table format. Annotators perform tagging by checking basic segments where a back-channel feedback can be provided.

3.3 Size of back-channel feedback corpus

Table 1 shows the size of our corpus constructed by two trained annotators. The corpus includes 5,416 back-channel feedbacks. This means that a back-channel feedback is generated at intervals of about 21 basic segments.

4 Corpus Evaluation

We conducted experiments for evaluating the tagging in the constructed corpus.

[3] A dialogue turn is defined as the interval between the time at which the driver starts to utter just after the operator finishes uttering and the time at which the driver finishes uttering just before the operator starts to utter.

[4] *Bunsetsu* is a linguistic unit in Japanese that roughly corresponds to a basic phrase in English. A bunsetsu consists of one independent word and zero or more ancillary words.

Table 2: Kappa values of the existing corpus

	a,c	a,d	a,b	c,d	b,c	b,d
κ	0.536	0.438	0.322	0.311	0.310	0.167

4.1 Coherency of corpus tagging

We conducted an evaluation experiment to confirm that the tagging is coherently performed in the corpus. In the experiment, two different annotators performed tagging on the same data, and then we measured the degree of the agreement between them. As the indicator, we used Cohen's kappa value (Cohen, 1960), calculated as follows:

$$\kappa = \frac{P(O) - P(E)}{1 - P(E)}$$

where $P(O)$ is the observed agreement between annotators, and $P(E)$ is the hypothetical probability of chance agreement. A subject who has a certain level of knowledge annotated 673 dialogue turns. The kappa value was 0.731 ($P(O) = 0.975, P(E) = 0.907$), and thus we can see the substantial agreement between annotators.

As the target for comparison, we used the kappa value in the existing back-channel feedback corpus (Kamiya et al., 2010). The corpus had been constructed by the way that the recorded driver's voice was replayed and 4 subjects independently produced back-channel feedbacks for the same sound. This means that the policies for tagging the existing corpus differ from those of our corpus, and are "on-line tagging," "tagging on the time axis" and "tagging without elaborating." In the exisiting corpus, 297 dialogue turns were used as driver's sound. Table 2 shows the kappa value between two among the 4 subjects. The kappa value of our corpus was higher than that between any subjects of the existing corpus, substantiating the high coherency of our corpus.

4.2 Validity of corpus tagging

In our corpus, we discretized the tagging points to enhance the coherency of tagging. However, such constraint restricts the points available for tagging and may make annotators provide tags at the unnatural timings. Therefore, we conducted a subjective experiment to evaluate the naturalness of the back-channel feedback timings. In the experiment, one subject listened to the replay of our back-channel feedback corpus and subjectively judged the naturalness of each timing. The back-channel feedback sound was generated in the same way described in Section 3.2.

In the experiment, we used 345 dialogue turns including 131 back-channel feedbacks. 98.47% of all the back-channel feedbacks were judged to be natural. Only 2 back-channel feedbacks were judged to be unnatural because the intervals between them and the back-channel feedbacks provided immediately before them were felt too short. This showed the validity of our discretization of tagging points.

5 Conclusion

This paper described the design, construction and evaluation of the back-channel feedback corpus which had the coherency of tagged back-channel feedback timings. We constructed the spoken dialogue corpus including 5,416 back-channel feedbacks in 11,181 dialogue turns. The results of our evaluation confirmed high coherency and enough naturalness of our corpus.

In the future, we will use our corpus to see to what extent the timings of back-channel feedbacks that have been annotated correlate with the cues provided by earlier researchers. Then we will develop a system which can detect back-channel feedback timings comprehensively.

Acknowledgments: This research was supported in part by the Grant-in-Aid for Challenging Exploratory Research (No.21650028) of JSPS.

References

N. Cathcart, J. Carletta, and E. Klein. 2003. A shallow model of backchannel continuers in spoken dialogue. In *Proc. of 10th EACL*, pages 51–58.

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46.

Y. Kamiya, T. Ohno, S. Matsubara, and H. Kashioka. 2010. Construction of back-channel utterance corpus for responsive spoken dialogue system development. In *Proc. of 7th LREC*.

N. Kawaguchi, S. Matsubara, K. Takeda, and F. Itakura. 2005. CIAIR in-car speech corpus – influence of driving status–. *IEICE Trans. on Info. and Sys.*, E88-D(3):578–582.

S. K. Maynard. 1989. *Japanese conversation : self-contextualization through structure and interactional management*. Ablex.

M. Takeuchi, N. Kitaoka, and S. Nakagawa. 2004. Timing detection for realtime dialog systems using prosodic and linguistic information. In *Proc. of Speech Prosody 2004*, pages 529–532.

N. Ward and W. Tsukahara. 2000. Prosodic features which cue back-channel responses in English and Japanese. *Journal of Pragmatics*, 32:1177–1207.

Representing Uncertainty about Complex User Goals in Statistical Dialogue Systems

Paul A. Crook
Interaction Lab
Heriot-Watt University
Edinburgh, United Kingdom
p.a.crook@hw.ac.uk

Oliver Lemon
Interaction Lab
Heriot-Watt University
Edinburgh, United Kingdom
o.lemon@hw.ac.uk

Abstract

We point out several problems in scaling-up statistical approaches to spoken dialogue systems to enable them to deal with complex but natural user goals, such as disjunctive and negated goals and preferences. In particular, we explore restrictions imposed by current independence assumptions in POMDP dialogue models. This position paper proposes the use of Automatic Belief Compression methods to remedy these problems.

1 Introduction

One of the main problems for a spoken dialogue system is to determine the user's goal (*e.g.* plan suitable meeting times or find a good Indian restaurant nearby) under uncertainty, and thereby to compute the optimal next system dialogue action (*e.g.* offer a restaurant, ask for clarification). Recent research in statistical spoken dialogue systems (SSDS) has successfully addressed aspects of these problems through the application of Partially Observable Markov Decision Process (POMDP) approaches (Thomson and Young, 2010; Young et al., 2010). However POMDP SSDS are currently limited by an impoverished representation of user goals adopted to enable tractable learning.

Current POMDP SSDS state approximations make it impossible to represent some plausible user goals, *e.g.* someone who wants to know about nearby cheap restaurants *and* high-quality ones further away, or wants to schedule a meeting anytime this week except monday afternoon (also see Examples in Tables 1–3). This renders dialogue management sub-optimal and makes it impossible to deal adequately with the following types of user utterance: "I'm looking for French or Italian food", or "Not Italian, unless it's expensive". User utterances with negations and disjunctions of various sorts are very natural, and exploit the full power of natural language input. Moreover, work in dialogue system evaluation, *e.g.* (Walker et al., 2004; Lemon et al., 2006), shows that real user goals are generally *sets of items* with different features, rather than a single item. People like to explore possible trade offs between features of items.

A central challenge for the field of spoken dialogue systems is therefore to: develop realistic *large-scale* statistical approaches with an accurate, extended representation of user goals.

In this paper we propose that the independence assumptions that have guided POMDP SSDS design to date should be relaxed, user goal sets should be introduced and that the subsequent explosion in the size of the state space should be dealt with by employing *Automatic Belief Compression (ABC)* techniques.

2 POMDP SSDS

Partially Observable Markov Decision Processes (POMDPs) are Markov Decision Processes where the system's state is only *partially observable*, *i.e.* there is *uncertainty* as to what the true state is. The ability to account for uncertainty is crucial for spoken dialogue systems because their knowledge about the state is uncertain due to speech recognition errors and the fact that the user's goals are not directly observable. In POMDP models of spoken dialogue (Williams and Young, 2005; Thomson and Young, 2010; Young et al., 2010) the dialogue policy (what the system should say next) is based not on a single view of the current state of the conversation, but on a probability distribution over all possible states of the conversation. The optimal POMDP SSDS dialogue act thus automatically takes account of the uncertainty about the user's utterances and goals.

All work to date on POMDP SSDS has assumed *(i) that a user has a singular, fully constrained, fixed goal* and *(ii) that there is one source of un-*

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 209–212,
The University of Tokyo, September 24-25, 2010.

User goal: good Italian but not Thai food	User Goal: good Italian but not Thai food
POMDP SSDS (Example 1)	ABC-SSDS (Example 2)
U1: What restaurants are nearby? S1: There are 3 nearby. What type of food do you want? U2: *Not Italian, unless it's really good* S2: There are 2 Thai restaurants. ⟨*The compressed state (i.e. marginal likelihoods in Figure 1) indicates that Italian food is dis-preferred.*⟩ U3: No Thanks. Bye.	U'1: What restaurants are nearby? S'1: There are 3 nearby. What type of food do you want? U'2: *Not Italian, unless it's really good* S'2: Ok. Al Capone's is highly rated. There are also 2 Thai restaurants nearby. U'3: Great. I'll try Al Capone's.

Table 1: State-of-the-art POMDP SSDS vs ABC-SSDS: Example dialogues 1 and 2. U=User, S=System

certainty; errors in speech recognition. The traditional role of the Dialogue Manager (DM) is therefore one of reducing uncertainty until its belief is strongly focused on a single goal which strictly fits the user's stated constraints. Both the policies that DMs currently follow and the design decisions implemented to make SDSS tasks learnable use these assumptions (Thomson and Young, 2010; Young et al., 2010). The resulting systems force users into selecting a single domain object as their goal in order to progress the dialogue. The example POMDP SSDS dialogues, Examples 1 and 3 (Tables 1 and 2), show the problems inherent in this approach.

User goal: good Italian or good French food
S1: How can I help you? U1: What restaurants are nearby? S2: There are 6, what type of food do you want? U2: *Err... Italian or French* S3: Do you want Italian or French food? U3: Uh. either. S4: Do you want Italian or French food? U4: Ok , how about French then. S5: There are 2 good French restaurants nearby. What price range do you want? U5: What about Italian? S6: Did you say Italian? U6: QUIT

Table 2: POMDP SSDS: Example 3

3 Independence Assumptions

Even considering limited domains, POMDP state spaces grow very quickly. For example, consider finding a user's restaurant preference, which involves getting 4 pieces of information, *i.e.* food-type, city-area, price range, quality rating. Given 8 food types, 8 city areas, 3 price ranges and 3 quality ratings, coupled with 7 user actions and a 3^4 dialogue progress indicator[1] then the dialogue state space contains $8 \times 8 \times 3 \times 3 \times 7 \times 3^4 = 326,592$ states. A POMDP belief space is a probability distribution over all these dialogue states, *i.e.* a $326,592$ *dimensional* real valued ($\mathbb{R}$) space.

In order to render such large belief spaces tractable, the current state of the art in POMDP SSDS uses a variety of *handcrafted* compression techniques, such as making several types of independence assumption. For example, by assuming that users are only ever interested in one type of food or one location, and that their interests in food type, price range, quality, *etc.* are independent, the $326,592$ real valued state space can be reduced to a much smaller "summary space" (Williams and Young, 2005) consisting of, say, $4 \times \mathbb{R}$ values[2]. See Figure 1 for a graphical depiction of such assumptions[3].

As illustrated by Figure 1 the information lost due to the independence assumptions mean that these approaches are unable to support conversations such as that shown in Example 2 (Table 1).

4 Sets of User Goals

Getting rid of independence assumptions allows the DM to reason and ask questions about the user's requirements in a more rational way. It can, for example distinguish between the user wanting "excellent Italian" or "any Thai" versus only "excellent" restaurants – see Figure 1. However, the resulting high dimensional real valued state space can still only represent uncertainly over *singular* user goals (limited to *single* points in the feature space, *e.g.* an excellent Italian restaurant).

[1] Whether each piece of information is obtained, confirmed or unknown.

[2] By considering only the maximum marginal likelihood for each of the features.

[3] These apply after utterance U2/U'2 of Example 1.

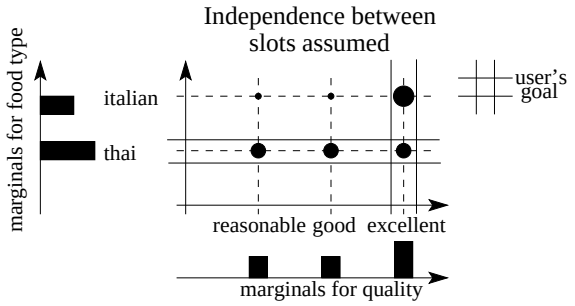


Figure 1: Assuming independence of features is equivalent to marginalising across features. Here, marginalisation incorrectly suppresses belief in Italian. Thai retains a uniform belief (which exists across all restaurant types not yet mentioned).

To achieve a substantial gain in the flexibility of SSDS we need to allow user's goals that are *sets* of points. Maintaining beliefs over "sets of goals" allows a POMDP DM to refine its belief in the user's requirements (managing speech recognition errors) without forcing the user to specify a singular tightly constrained goal. The disadvantage of this approach is a further expansion of the state space.

5 Automatic Belief Compression

To allow for expansion of the state space, whilst keeping its size tractable for policy learning, we suggest replacing handcraft approaches with *Automatic Belief Compression (ABC)* techniques.

We propose to use proven, principled statistical learning methods for automatically reducing the dimensionality of belief spaces, but which preserve the useful distributions within the full space.

Two complementary methods that we are currently investigating are **VDC** (Poupart, 2005) and **E-PCA** (Roy and Gordon, 2002; Roy et al., 2005). These methods have been applied successfully in a real-time daily living assistant with over 10^6 states (St-Aubin et al., 2000; Hoey and Poupart, 2005; Poupart et al., 2006) and to robotic navigation by (Roy and Gordon, 2002; Roy et al., 2005). They:

- reduce the dimensionality of state spaces that were previously intractable for POMDP solution methods, and
- automatically compress the representation of belief space distributions to take advantage of sparsity between likely distributions.

The tight coupling between some dialogue states and actions (*e.g.* a user's goal state `travel-from-London` and system act `confirm-from-London`) has led some researchers to conclude that compression techniques, such as state aggregation, are not useful in the dialogue domain (Williams and Young, 2007). However, such tight coupling may not exist for all states, indeed VDC has already been applied to a small spoken dialogue system problem (Poupart, 2005) where it was shown that compressions could be found *without* losing any information[4]. Further, for POMDP approaches the *state* is not the dialogue state but the belief distribution over dialogue states. Incompressibility at the dialogue state level does not rule out compressibility of belief distributions. Finally, our introduction of sets for user goals should provide additional possibilities for compression.

Our aim in applying ABC methods is to allow POMDP SSDS to handle the much larger state spaces that are required to achieve the expressiveness which we believe will be a real benefit to users. We plan to do this for real world tasks, *e.g.* a city search over 1000s of entities with an uncompressed belief space of the order of $10^8 \times \mathbb{R}$.

6 Target Dialogues

In general, when a user starts a dialogue they rarely have a singular goal in mind (Walker et al., 2004; Lemon et al., 2006). Their goal is not a fixed point in the domain but instead can be thought of as a (possibly disconnected) set of points, for example either a nearby cheap restaurant or high-quality one further away. The set represents trade offs that the particular user is interested in. People rarely communicate their goals in terms of such distributions or trade offs, preferring to provide information in a piecemeal manner and thus incrementally explore the domain.

In Examples 1–4 (Tables 1–3) we contrast the operation of a current state-of-the-art POMDP SSDS with our proposed *ABC-SSDS* system. The user's goal in Examples 3 and 4 (Tables 2 and 3) is to explore what restaurants are nearby, with a preference for French or Italian. Current POMDP SSDS approaches assume that any spread of probability mass in the belief space represents uncertainty which needs to be resolved. This generates problems for the POMDP SSDS in Example 3 since the user is forced into specifying one food type at a time, resulting in an unwieldy confirma-

[4]Compressing a test problem of 433 states to 31 basis functions, *i.e.* a summary space of 31 states.

User goal: good Italian or good French food
S'1: How can I help you?
U'1: What restaurants are nearby?
S'2: There are 6, what type of food do you want?
U2': *Err... Italian or French*
S'3: Ok, there are 2 good French restaurants nearby, and one good Italian.
U'4: OK. Which is best quality?
S'3: Mamma Mia's has the best rating.
U'5: Great. I'll go there!

Table 3: Proposed *ABC-SSDS*: Example 4

tion step (S6 of Example 3) where the user is assumed to have *changed their mind*. In contrast, the proposed *ABC-SSDS* system can believe that the user has requested information on the combined set of French and Italian restaurants.

In Examples 1 and 2 (both shown in Table 1) the user's goal is to explore restaurants nearby, including only well-rated Italians. Here the standard POMDP SSDS is forced by its "summary space" (see marginals in Figure 1) to incorrectly represent the user's goal after U2 "Not Italian, unless it's really good" by ruling out all Italian restaurants[5]. The *ABC-SSDS* user is able to find the restaurant of their choice, whereas the POMDP SSDS user's choice is artificially restricted, and they quit having failed to find a suitable item.

The *ABC-SSDS* style of dialogue is clearly more efficient than that of current POMDP SSDS. It seems likely that users of such a system may also find the style of the conversation more natural, and may be more confident that their eventual choices really meet their goals (Walker et al., 2004).

All of these hypotheses remain to be explored in our future empirical work.

7 Conclusion

We present several problems for current POMDP approaches to spoken dialogue systems, concerning the representation of complex, but natural, user goals. We propose the development of principled *automatic* methods for dimensionality reduction, in place of the ad-hoc assumptions and *handcrafted* compressions currently used.

In parallel we are also exploring: (i) what approaches are required for *updating* beliefs over *sets* in real time – in principle a method similar to *user goal state partitioning* (Young et al., 2010) would appear to be sufficient, (ii) what exploitable bounds exist on the sets of goals that are communicable and (iii) to what extent the complexity of user goal sets can be traded off against the overall user experience.

[5]There are several ways to try to remedy this, but all have problems.

Acknowledgments

Thanks to Dr. Jesse Hoey, the SIGdial reviewers and the Engineering and Physical Sciences Research Council (EPSRC) project EP/G069840/1.

References

J. Hoey and P. Poupart. 2005. Solving POMDPs with Continuous or Large Discrete Observation Spaces. In *IJCAI*.

O. Lemon, K. Georgila, and J. Henderson. 2006. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Evaluation. In *IEEE/ACL Spoken Language Technology*.

P. Poupart, N. Vlassis, and J. Hoey. 2006. An Analytic Solution to Discrete Bayesian Reinforcement Learning. In *ICML*.

P. Poupart. 2005. *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. Ph.D. thesis, Dept. Computer Science, University of Toronto.

N. Roy and G. Gordon. 2002. Exponential Family PCA for Belief Compression in POMDPs. In *NIPS*.

N. Roy, G. Gordon, and S. Thrun. 2005. Finding Approximate POMDP Solutions Through Belief Compression. *Artificial Intelligence Research*, 22(1-40).

R. St-Aubin, J. Hoey, and C. Boutilier. 2000. Approximate policy construction using decision diagrams. In *NIPS*.

B. Thomson and S. Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588.

M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. User tailored generation in the match multimodal dialogue system. *Cognitive Science*, 28:811–840.

J. Williams and S. Young. 2005. Scaling Up POMDPs for Dialog Management: The "Summary POMDP" Method. In *Proc. ASRU*.

J. Williams and S. Young. 2007. Scaling POMDPs for spoken dialog management. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2116 –2129, Sept.

S. Young, M. Gašić, S. Keizer, F. Mairesse, B. Thomson, and K. Yu. 2010. The Hidden Information State model: a practical framework for POMDP based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174.

Investigating Clarification Strategies in a Hybrid POMDP Dialog Manager

Sebastian Varges and **Silvia Quarteroni** and **Giuseppe Riccardi** and **Alexei V. Ivanov**
Department of Information Engineering and Computer Science
University of Trento, 38050 Povo di Trento, Italy
{varges|silviaq|riccardi|ivanov}@disi.unitn.it

Abstract

We investigate the clarification strategies exhibited by a hybrid POMDP dialog manager based on data obtained from a phone-based user study. The dialog manager combines task structures with a number of POMDP policies each optimized for obtaining an individual concept. We investigate the relationship between dialog length and task completion. In order to measure the effectiveness of the clarification strategies, we compute concept precisions for two different mentions of the concept in the dialog: first mentions and final values after clarifications and similar strategies, and compare this to a rule-based system on the same task. We observe an improvement in concept precision of 12.1% for the hybrid POMDP compared to 5.2% for the rule-based system.

1 Introduction

In recent years, probabilistic models of dialog have been introduced into dialog management, the part of the spoken dialog system that takes the action decision. A major motivation is to improve robustness in the face of uncertainty, in particular due to speech recognition errors. The interaction is characterized as a dynamic system that manipulates its environment by performing dialog actions and perceives feedback from the environment through its sensors. The original sensory information is obtained from the speech recognition (ASR) results which are typically processed by a spoken language understanding module (SLU) before being passed on to the dialog manager (DM).

The seminal work of (Levin et al., 2000) modeled dialog management as a Markov Decision Process (MDP). Using reinforcement learning as the general learning paradigm, an MDP-based dialog manager incrementally acquires a policy by obtaining rewards about actions it performed in specific dialog states. As we found in earlier experiments, an MDP can learn to gradually drop the use of clarification questions if there is no noise. This is due to the fact that clarifications do not improve the outcome of the dialog, i.e. the reward. However, with extremely high levels of noise, the learner prefers to end the dialog immediately (Varges et al., 2009). In contrast to deliberate decision making in the pragmatist tradition of dialog processing, reinforcement learning can be regarded as low-level decision making.

MDPs do not account for the observational uncertainty of the speech recognition results, a key challenge in spoken dialog systems. Partially Observable Markov Decision Process (POMDPs) address this issue by explicitly modeling how the distribution of observations is governed by states and actions.

In this work, we describe the evaluation of a divide-and-conquer approach to dialog management with POMDPs that optimizes policies for acquiring individual concepts separately. This makes optimization much easier and allows us to model the confusability of concrete concept values explicitly. This also means that different clarification strategies are learned for individual concepts and even individual concept values. The use of the POMDP policies is orchestrated by an explicit task structure, resulting in a hybrid approach to dialog management. The evaluation involved a user study of 20 subjects in a tourist information domain. The system is compared against a rule-based baseline system in the same domain that was also evaluated with 20 subjects.

2 Hybrid POMDP dialog management

In this section we introduce the hybrid POMDP dialog manager that was used in the data collection.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 213–216,
The University of Tokyo, September 24-25, 2010.

2.1 Concept-level POMDPs

The domain is a tourist information system that uses 5 different policies that can be used in 8 different task roles (see below). For each concept we optimized an individual policy. The number of states of the POMDP can be limited to the concept values, for example a location name such as `trento`. The set of actions consists of a question to obtain the concept (e.g. `question-location`), a set of clarification actions (e.g. `verify-trento`) and a set of submit actions (e.g. `submit-trento`). POMDP modeling including a heuristically set reward structure follows the (simpler) 'tiger problem' that is well-known in the AI community (Kaelbling et al., 1998): the system has a number of actions to obtain further information which it can try and repeat in any order until it is ready to commit to a concept value. For optimization we used the `APPL` solver (Kurniawati et al., 2008).

2.2 Task structure and dialog management

The use of individual policies is orchestrated by an explicit task structure that activates and deactivates them. The task structure is essentially a directed `AND-OR` graph with a common root node. The dialog manager maintains a separate belief distribution for each concept. Figure 1 shows the general system architecture with a schematic view of the task structure, and additionally a more detailed view of an active location node. In the example, the root node has already finished and the system is currently obtaining the location for a lodging task. The term 'role' refers to a concept's part in the task, for example a month may be the check-in or check-out month for accommodation booking.

At the beginning of a dialog, the task structure is initialized by activating the root node. A top level function activates nodes of the task structure and passes control to that node. Each node maintains a belief b_c for a concept c, which is used to rank the available actions by computing the inner product of policy vectors and belief. The top-ranked action a_m is selected by the system, i.e. it is exploiting the policy, and passed to the natural language generator (NLG). Next, the top-ranked SLU results for the active node and concept are used as observation $z_{u,c}$ to update the belief to b'_c, which

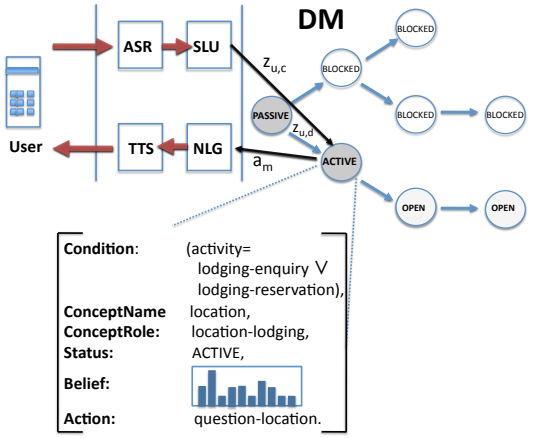


Figure 1: System architecture with Task Structure (task node example in detailed view)

follows the standard method for POMDPs:

$$b'_c(s') = \sum_{s \in S} b_c(s)\, T(s, a_m, s')\, O(a, s', z_{u,c}) / p_{z_{u,c}} \quad (1)$$

where probability $b'_c(s')$ is the updated belief of being in state s', which is computed as the sum of the probabilities of transitioning from all previous belief points s to s' by taking machine action a_m with probability $T(s, a_m, s')$ and observing $z_{u,c}$ with (smoothed) probability $O(a_m, s', z_{u,c})$. Normalization to obtain a valid probability distribution is performed by dividing by the probability of the observation $p_{z_{u,c}}$.

A concept remains active until a submit action is selected. At that point, the next active node is retrieved from the task structure and immediately used for action selection with an initially uniform belief. Submit actions are not communicated to the user but collected and used for the database query at the end of the dialog.

Overanswering, i.e. the user providing more information than directly asked for, is handled by *delayed belief updating*: the SLU results are stored until the first concept of a matching type becomes active. This is a heuristic rule designed to ensure that a concept is interpreted in its correct role. Operationally, unused SLU results $z_{u,d}$ (where concept $d \neq c$) are passed on to the next activated task node (see also figure 1).

3 Experiments and data analysis

We conducted user studies with two systems involving 20 subjects and 8 tasks in each study. The systems use a Voice XML platform to drive ASR and TTS components. Speech recognition is

	Lodging Task		Event Enquiry	
	TCR	#turns	TCR	#turns
Rule-based DM	75.5% (40/53)	13.7 (σ=4.8)	66.7% (28/42)	8.7 (σ=3.3)
POMDP-DM	78.1% (50/64)	23.0 (σ=8.8)	84.3% (27/32)	14.4 (σ=4.5)

Table 1: Task completion and length metrics

based on statistical language models for the opening prompt, and is grammar-based otherwise. One system used the hybrid POMDP-DM, the other is a rule-based dialog manager that uses explicit, heuristically set confidence thresholds to trigger the use of clarification questions (Varges et al., 2008).

Dialog length and task completion Table 1 shows task completion rates ('TCR') and durations ('#turns') for the POMDP and rule-based systems. Task completion in this metric is defined as the number of tasks of a certain type that were successfully concluded. Duration is measured in the number of turn pairs consisting of a system action followed by a user action. We combine the counts for two closely related lodging tasks. The number of tasks is shown in brackets. Table 1 shows that the POMDP-DM successfully concludes more and longer lodging tasks and almost as many event tasks. In general, the POMDP policies can be described as more cautious although obviously the dialog length of the rule system depends on the chosen thresholds.

Concept precision at the value level In order to measure the effect of the clarification strategies in both systems, we computed concept precisions for two different mentions of a concept in a dialog (table 2): first mentions and final values after clarifications and similar strategies. The rationale for this metric is that the last mentioned concept value is the value that the system ultimately obtains from the user, which is used in the database query:

- if the system decides not to use clarifications, the only mentioned value is the accepted one,
- if the system verifies and obtains a positive answer, the last mentioned value is the accepted one,
- if the system verifies and obtains a negative answer, the user will mention a new value (which may or may not be accepted).

Thus, this metric is a uniform way of capturing the obtained values from systems that internally use very different dialog managers and representations. The actual precision of a concept C is calculated by comparing SLU results to annotations and counting true positives (matches M) and false positives (separated into mismatches N and entirely un-annotated concepts U): $Prec(C) = \frac{M}{M+N+U}$. Unrecognized concepts, on the other hand, are recall related and not counted since they cannot be part of any system belief.

	Rule-based DM			POMDP-DM		
	first	final	Δ%	first	final	Δ%
a) activity	0.78	0.74	-4.1	0.83	0.88	5.0
b) location	0.64	0.74	15.8	0.69	0.73	6.3
c) starrating	0.67	0.70	3.4	0.90	0.97	7.7
d) month	0.85	0.89	4.3	0.76	0.86	12.7
e) day	0.70	0.76	8.3	0.61	0.76	25.3
ALL (a-e)	0.74	0.78	**5.2**	0.74	0.83	**12.1**
Clarifications	0.84	0.85	1.5	0.96	0.87	-8.8

Table 2: Concept precision of first vs final value

As table 2 clearly shows, the use of clarification strategies has a positive effect on concept precision in both systems. The exception is the precision of concept activity in the rule-based system for which the system reprompted rather than verified.[1] In table 2, row 'All' refers to the average weighted precision of the five concepts. Both systems start from a similar level of overall precision. The relative improvement of the POMDP-DM for all concepts is 12.1%, compared to 5.2% of the rule-based DM.

We conducted a statistical significance test by computing the delta in the form of three values for individual data points, i.e. dialogs, and assigned `+1` for all changes from non-match to match, `-1` for a change in the opposite direction and `0` for everything else (e.g. from mismatch to mismatch). We found that, although there is a tendency for the POMDP-DM to perform better, the difference is not statistically significant at p=0.05 (a possible explanation is the data size since we are using human subjects).

We furthermore measured the precision of recognizing 'yes/no' answers to clarification questions. In contrast to actual concepts, there is no belief distribution for these in the DM since clarification actions are part of the concept POMDP models. We are thus dealing with individual one-off recognition results that should be entirely independent of each other. However, as table 2 (bottom)

[1]The second value obtained may be incorrect but above the confidence threshold; note that the rule system does not maintain a belief distribution over values.

shows, the precision of verifications decreases for the hybrid POMDP system. A plausible explanation for this is the increasing impatience of the users due to the longer dialog duration.

Characterization of dialog strategies For some concepts, the best policy is to ask the concept question once and then verify once before committing to the value (assuming the answer is positive). Other policies verify the same value twice. Another learned strategy is to ask the original concept question twice and then only verify the value once (assuming that the understood value was the same in both concept questions). In other words, the individual concept policies show different types of strategies regarding uncertainty handling. This is in marked contrast to the manually programmed DM that always asks the concept question once and verifies it if needed (concept activity being the exception).

HCI and language generation The domain is sufficiently simple to use template-based generation techniques to produce the surface forms of the responses. However, the experiments with the POMDP-DM highlight some new challenges regarding HCI aspects of spoken dialog systems: the choice of actions may not be 'natural' from the user's perspective, for example if the system asks for a concept twice. However, it should be possible to better communicate the (change in the) belief to the user.

4 Related work

The pragmatist tradition of dialog processing uses explicit representations of dialog structure to take decisions about clarification actions. These models are more fine-grained and often deal with written text, e.g. (Purver, 2006), whereas in spoken dialog systems a major challenge is managing the uncertainty of the recognition. Reinforcement learning approaches to dialog management learn decisions from (often simulated) dialog data in a less deliberative way. For example, the Hidden Information State model (Young et al., 2010) uses a reduced summary space that abstracts away many of the details of observations and dialog state, and mainly looks at the confidence scores of the hypotheses. This seems to imply that clarification strategies are not tailored toward individual concepts and their values. (Bui et al., 2009) uses factored POMDP representations that seem closest to our approach. However, the effect of clarifications does not seem to have been investigated.

5 Conclusions

We presented evaluation results for a hybrid POMDP system and compared it to a rule-based one. The POMDP system achieves higher concept precision albeit at the cost of longer dialogs, i.e. there is an empirically measurable trade-off between concept precision and dialog length.

Acknowledgments

This work was partially supported by the European Commission Marie Curie Excellence Grant for the ADAMACH project (contract No. 022593).

References

T.H. Bui, M. Poel, A. Nijholt, and J. Zwiers. 2009. A tractable hybrid DDN-POMDP approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering*, 15(2):273–307.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134.

H. Kurniawati, D. Hsu, and W.S. Lee. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Proc. Robotics: Science and Systems*.

E. Levin, R. Pieraccini, and W. Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1).

Matthew Purver. 2006. CLARIE: Handling clarification requests in a dialogue system. *Research on Language and Computation*, 4(2-3):259–288, October.

Sebastian Varges, Giuseppe Riccardi, and Silvia Quarteroni. 2008. Persistent information state in a data-centric architecture. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, Columbus, Ohio.

Sebastian Varges, Giuseppe Riccardi, Silvia Quarteroni, and Alexei V. Ivanov. 2009. The exploration/exploitation trade-off in reinforcement learning for dialogue management. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

S. Young, M. Gasic, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24:150–174.

Cooperative User Models in Statistical Dialog Simulators

Meritxell González[1,2], **Silvia Quarteroni**[1], **Giuseppe Riccardi**[1], **Sebastian Varges**[1]
[1] DISI - University of Trento, Povo (Trento), Italy
[2] TALP Center - Technical University of Catalonia, Barcelona, Spain
mgonzalez@lsi.upc.edu, name.lastname@disi.unitn.it

Abstract

Statistical user simulation is a promising methodology to train and evaluate the performance of (spoken) dialog systems. We work with a modular architecture for data-driven simulation where the "intentional" component of user simulation includes a User Model representing user-specific features. We train a dialog simulator that combines traits of human behavior such as cooperativeness and context with domain-related aspects via the Expectation-Maximization algorithm. We show that cooperativeness provides a finer representation of the dialog context which directly affects task completion rate.

1 Introduction

Data-driven techniques are a promising approach to the development of robust (spoken) dialog systems, particularly when training statistical dialog managers (Varges et al., 2009). User simulators have been introduced to cope with the scarcity of real user conversations and optimize a number of SDS components (Schatzmann et al., 2006).

In this work, we investigate the combination of aspects of human behavior with contextual aspects of conversation in a joint yet modular data-driven simulation model. For this, we integrate conversational context representation, centered on a Dialog Act and a Concept Model, with a User Model representing persistent individual features. Our aim is to evaluate different simulation regimes against real dialogs to identify any impact of user-specific features on dialog performance.

In this paper, Section 2 presents our simulator architecture and Section 3 focuses on our model of cooperativeness. Our experiments are illustrated in Section 4 and conclusions are summarized in Section 5.

Work partly funded by EU project ADAMACH (022593) and Spanish project OPENMT-2 (TIN2009-14675-C03).

2 Simulator Architecture

Data-driven simulation takes place within the rule-based version of the ADASearch system (Varges et al., 2009), which uses a taxonomy of 16 dialog acts and a dozen concepts to deal with three tasks related to tourism in Trentino (Italy): Lodging Enquiry, Lodging Reservation and Event Enquiry.

Simulation in our framework occurs at the *intention level*, where the simulator and the Dialog Manager (DM) exchange *actions*, i.e. ordered sequences of *dialog acts* and a number of *concept-value* pairs. In other words, we represent the DM action as $a_s = \{da_0, .., da_n\}$, (s is for "System") where da_j is short for a dialog act defined over zero or more concept-value pairs, $da_j(c_0(v_0), .., c_m(v_m))$.

In response to the DM action a_s, the different modules that compose the User Simulator generate an N-best list of simulated actions $A_u(a_s) = \{a_u^0, .., a_u^N\}$. The probability of each possible action being generated after the DM action a_s is estimated based on the conversation context. Such a context is represented by a User Model, a Dialog Act Model, a Concept Model and an Error Model (Quarteroni et al., 2010). The User Model simulates the behavior of an individual user in terms of goals and other behavioral features such as cooperativeness and tendency to hang up. The Dialog Act Model generates a distribution of M actions $A_u = \{a_u^0, .., a_u^M\}$. Then, one action $\hat{a}_u$ is chosen out of A_u. In order to vary the simulation behavior, the choice of the user action $\hat{a}_u$ is a random sampling according to the distribution of probabilities therein; making the simulation more realistic. Finally, the Concept Model generates concept values for $\hat{a}_u$; and the Error Model simulates the noisy ASR-SLU channel by "distorting" $\hat{a}_u$.

These models are derived from the ADASearch

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 217–220,
The University of Tokyo, September 24-25, 2010.

dataset, containing 74 spoken human-computer conversations.

2.1 User Model

The User Model represents user-specific features, both transient and persistent. The transient feature we focus on in this work is the user's goal in the dialog (UG), represented as a task name and the list of concepts and values required to fulfill it: an example of UG is {Activity(EventEnquiry), Time_day(2), Time_month(may), Event_type(fair), Location_name(Povo)}.

Persistent features included in our model so far are: patience, silence (no input) and cooperativeness. Patience pat is defined as the tendency to abandon the conversation (hang up event), i.e. $pat = P(HangUp|a_s)$. Similarly, NoInput probability noi is used to account for user behavior in noisy environments: $noi = P(NoInput|a_s)$. Finally, cooperativeness *coop* is a real value representing the ratio of concepts mentioned in a_s that also appear in $\hat{a}_u$ (see Section 3).

2.2 Dialog Act Model

We define three Dialog Act (DA) Models: Obedient (OB), Bigram (BI) and Task-based (TB).

In the Obedient model, total patience and cooperativeness are assumed of the user, who will always respond to each query requiring values for a set of concepts with an answer concerning exactly such concepts. Formally, the model responds to a DM action a_s with a single user action $\hat{a}_u$ obtained by consulting a rule table, having probability 1. In case a request for clarification is issued by the DM, this model returns a clarifying answer. Any offer from the DM to continue the conversation will be either readily met with a new task request or denied at a fixed probability: $A_u(a_s) = \{(\hat{a}_u, 1)\}$.

In the Bigram model, first defined in (Eckert et al., 1997), a transition matrix records the frequencies of transition from DM actions to user actions, including hang up and no input/no match. Given a DM action a_s, the model responds with a list of M user actions and their probabilities estimated according to action distribution in the real data: $A_u(a_s) = \{(a_u^0, P(a_u^0|a_s)), .., (a_u^M, P(a_u^M|a_s))\}$.

The Task-based model, similarly to the "goal" model in (Pietquin, 2004), produces an action distribution containing only the actions observed in the dataset of dialogs in the context of a specific task T_k. The TB model divides the dataset into one partition for each T_k, then creates a task-specific bigram model, by computing $\forall\ k$: $A_u(a_s) = \{(a_u^0, P(a_u^0|a_s, T_k)), .., (a_u^M, P(a_u^M|a_s, T_k))\}$. As the partition of the dataset reduces the number of observations, the TB model includes a mechanism to back off to the simpler bigram and unigram models.

2.3 Concept & Error Model

The Concept Model takes the action $\hat{a}_u$ selected by the DA Model and attaches values and sampled interpretation confidences to its concepts. In this work, we adopt a Concept Model which assigns the corresponding User Goal values for the required concepts, which makes the user simulated responses consistent with the user goal.

The Error Model is responsible of simulating the noisy communication channel between user and system; as we simulate the error at SLU level, errors consist of incorrect concept values. We experiment with a data-driven model where the precision Pr_c obtained by a concept c in the reference dataset is used to estimate the frequency with which an error in the true value $\tilde{v}$ of c will be introduced: $P(c(v)|c(\tilde{v})) = 1 - Pr_c$ (Quarteroni et al., 2010).

3 Modelling Cooperativeness

As in e.g. (Jung et al., 2009), we define cooperativeness at the turn level ($coop_t$) as a function of the number of dialog acts in the DM action a_s sharing concepts with the dialog acts in the user action a_u; at the dialog level, *coop* is the average of turn-level cooperativeness.

We discretize *coop* into a binary variable reflecting high vs low cooperativeness based on whether or not the dialog cooperativeness exceeds the median value of *coop* found in a reference corpus; in our ADASearch dataset, the median value found for *coop* is 0.28; hence, we annotate dialogs as cooperative if they exceed such a threshold, and as uncooperative otherwise. Using a corpus threshold allows domain- and population-driven tuning of cooperativeness rather than a "hard" definition (as in (Jung et al., 2009)).

We then model cooperativeness as two bigram models, reflecting the high vs low value of *coop*. In practice, given a DM action a_s and the *coop* value ($\kappa = high/low$) we obtain a list of user actions and their probabilities:
$A_u(a_s, \kappa) = \{(a_u^0, P(a_u^0|a_s, \kappa)), .., (a_u^M, P(a_u^M|a_s, \kappa))\}$.

3.1 Combining cooperativeness and context

At this point, the distribution $A_u(a_s, \kappa)$ is linearly interpolated with the distribution of actions $A_u(a_s, \psi)$ obtained using the DA model ψ (in the Task-based DA model; ψ can have three values, one for each task as explained in Section 2.2):

$$A_u(a_s) = \lambda_\kappa \cdot A_u(a_s, \kappa) + \lambda_\psi \cdot A_u(a_s, \psi),$$

where λ_κ and λ_ψ are the weights of each feature/model and $\lambda_\psi + \lambda_\kappa = 1$.

For each user action a_u^i, λ_κ and λ_ψ are estimated using the Baum-Welch Expectation-Maximization algorithm as proposed by (Jelinek and Mercer, 1980). We use the distributions of actions obtained from our dataset and we align the set of actions of the two models. Since we only have two models, we only need to calculate expectation for one of the distributions:

$$P(\kappa|a_s, a_u^i) = \frac{P(a_u^i|a_s, \kappa)}{P(a_u^i|a_s, \kappa) + P(a_u^i|a_s, \psi)} \; \forall_{i=0}^{M} a_u^i$$

where M is the number of actions. Then, the weights λ_κ and λ_ψ that maximize the data likelihood are calculated as follows:

$$\lambda_\kappa = \frac{\sum_{j=0}^{M} P(\kappa|a_s, a_u^j)}{M}; \lambda_\psi = 1 - \lambda_\kappa.$$

The resulting combined distribution $A_u(a_s)$ is obtained by factoring the probabilities of each action with the weight estimated for the particular distribution:

$$A_u(a_s) = \{(a_u^0, \lambda_\kappa \cdot P(a_u^0|a_s, \kappa)), .., (a_u^M, \lambda_\kappa \cdot P(a_u^M|a_s, \kappa)),$$
$$(a_u^0, \lambda_\psi \cdot P(a_u^0|a_s, \psi)), .., (a_u^M, \lambda_\psi \cdot P(a_u^M|a_s, \psi))\}$$

3.2 Effects of cooperativeness

To assess the effect of the cooperativeness feature in the final distribution of actions, we set a 5-fold cross-validation experiment with the ADASearch dataset where we average the λ_κ estimated at each turn of the dialog. We investigated in which context cooperativeness provides more contribution by comparing the λ_κ weights attributed by high vs. low *coop* models to user action distributions in response to Dialog Manager actions.

Figure 1 shows the values achieved by λ_κ for several DM actions for high vs low *coop* regimes. We can see that λ_κ achieves high values in case of uncooperative users in response to DM dialog acts as [ClarificationRequest] and [Info-request]. In contrast, forward-looking actions, such as the ones including [Offer], seem to discard the contribution of the low *coop* model, but to favor the contribution provided by high *coop*.

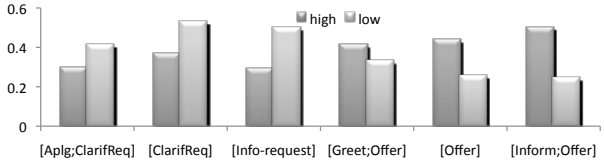


Figure 1: Estimated λ_κ weights in response to selected DM actions in case of high/low *coop*

4 Experiments

We evaluate our simulator models using two methods: first, "offline" statistics are used to assess how realistic the action estimations by DA Models are with respect to a dataset of real conversations (Sec. 4.1); then, "online" statistics (Sec. 4.2) evaluate end-to-end simulator performance in terms of dialog act distributions, error robustness and task duration and completion rates by comparing real dialogs with fresh simulated dialogs using action sampling in the different simulation models.

4.1 "Offline" statistics

In order to compare simulated and real user actions, we evaluate dialog act Precision (P_{DA}) and Recall (R_{DA}) following the methodology in (Schatzmann et al., 2005).

For each DM action a_s the simulator picks a user action $\hat{a}_u$ from $A_u(a_s)$ and we compare it with the real user choice $\tilde{a}_u$. A simulated dialog act is correct when it appears in the real action $\tilde{a}_u$. The measurements were obtained using 5-fold cross-validation on the ADASearch dataset.

Table 1: Dialog Act Precision and Recall

DA Model	Simulation (a_u^*)		Most frequent (a_u^*)	
	P_{DA}	R_{DA}	P_{DA}	R_{DA}
OB	33.8	33.4	33.9	33.5
BI ($+coop$)	35.6 (**35.7**)	35.5 (**35.8**)	49.3 (47.9)	48.8 (47.4)
TB ($+coop$)	38.2 (**39.7**)	38.1 (**39.4**)	51.1 (50.6)	50.6 (50.2)

Table 1 shows P_{DA}/R_{DA} obtained for the OB, BI and TB models alone and with cooperativeness models ($+coop$). First, we see that TB is much better than BI and OB at reproducing real action selection. This is also visible in both P_{DA} and R_{DA} obtained by selecting a_u^*, the most frequent user action from the A_s generated by each model. By definition, a_u^* maximizes the expected P_{DA} and R_{DA}, providing an upper bound for our models; however, to reproduce *any possible* user behavior, we need to sample a_u rather than choosing it by frequency. By now inspecting ($+coop$)

values in Table 1, we see that explicit cooperativeness models match real dialogs more closely. It points out that partitioning the reference dataset in high vs low *coop* sets allows better data representation. There is however no improvement in the a_u^* case: we explain this by the fact that by "slicing" the reference dataset, the cooperative model augments data sparsity, affecting robustness.

4.2 "Online" statistics

We now discuss online deployment of our simulation models with different user behaviors and "fresh" user goals and data. To align with the ADASearch dataset, we ran 60 simulated dialogs between the ADASearch DM and each combination of the Task-based and Bigram models and high and low values of *coop*. For each set of simulated dialogs, we measured task duration, defined as the average number of turns needed to complete each task, and task completion rate, defined as: $TCR = \frac{number\ of\ times\ a\ task\ has\ been\ completed}{total\ number\ of\ task\ requests}$.

Table 2 reports such figures in comparison to the ones obtained for real dialogs from the ADASearch dataset. In general, we see that task duration is closer to real dialogs in the Bigram and Task-based models when compared to the Obedient model. Moreover, it can easily be observed in both BI and TB models that under *high-coop* regime (in boldface), the number of turns taken to complete tasks is lower than under *low-coop*. Furthermore, in both TB and BI models, TCR is higher when cooperativeness is higher, indicating that cooperative users make dialogs not only shorter but also more efficient.

Table 2: Task duration and TCR in simulated dialogs with different regimes vs real dialogs.

	Lodging Enquiry		Lodging Reserv		Event Enquiry		All
Model	#turns	TCR	#turns	TCR	#turns	TCR	TCR
OB	9.2±0.0	78.1	9.7±1.4	82.4	8.1±2.9	66.7	76.6
BI+low	15.1±4.1	71.4	14.2±3.9	69.4	9.3±1.8	52.2	66.7
BI+high	**12.1±2.5**	**74.6**	**12.9±3.1**	**82.9**	**7.8±1.8**	**75.0**	**77.4**
TB+low	13.6±4.1	75.8	13.4±3.7	83.3	8.4±3.3	64.7	77.2
TB+high	**11.6±2.8**	**80.0**	**12.6±3.6**	**83.7**	**6.5±1.9**	**57.1**	**78.4**
Real dialogs	11.1±3.0	71.4	12.7±4.7	69.6	9.3±4.0	85.0	73.4

5 Conclusion

In this work, we address data-driven dialog simulation for the training of statistical dialog managers. Our simulator supports a modular combination of user-specific features with different models of dialog act and concept-value estimation, in addition to ASR/SLU error simulation.

We investigate the effect of joining a model of user intentions (Dialog Act Model) with a model of individual user traits (User Model). In particular, we represent the user's cooperativeness as a real-valued feature of the User Model and create two separate simulator behaviors, reproducing high and low cooperativeness. We explore the impact of combining our cooperativeness model with the Dialog Act model in terms of dialog act accuracy and task success.

We find that 1) an explicit modelling of user cooperativeness contributes to an improved accuracy of dialog act estimation when compared to real conversations; 2) simulated dialogs with high cooperativeness result in higher task completion rates than low-cooperativeness dialogs. In future work, we will study yet more fine-grained and realistic User Model features.

References

W. Eckert, E. Levin, and R. Pieraccini. 1997. User modeling for spoken dialogue system evaluation. In *Proc. IEEE ASRU*.

F. Jelinek and R. L. Mercer. 1980. Interpolated estimation of Markov source parameters from sparse data. In *Workshop on Pattern Recognition in Practice*.

S. Jung, C. Lee, K. Kim, and G. G. Lee. 2009. Hybrid approach to user intention modeling for dialog simulation. In *Proc. ACL-IJCNLP*.

O. Pietquin. 2004. *A Framework for Unsupervised Learning of Dialogue Strategies*. Ph.D. thesis, Faculté Polytechnique de Mons, TCTS Lab (Belgique).

S. Quarteroni, M. González, G. Riccardi, and S. Varges. 2010. Combining user intention and error modeling for statistical dialog simulators. In *Proc. INTERSPEECH*.

J. Schatzmann, K. Georgila, and S. Young. 2005. Quantitative evaluation of user simulation techniques for spoken dialogue systems. In *Proc. SIGDIAL*.

J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *Knowl. Eng. Rev.*, 21(2):97–126.

S. Varges, S. Quarteroni, G. Riccardi, A. V. Ivanov, and P. Roberti. 2009. Leveraging POMDPs trained with user simulations and rule-based dialogue management in a spoken dialogue system. In *Proc. SIGDIAL*.

Modeling Spoken Decision Making Dialogue and Optimization of its Dialogue Strategy

Teruhisa Misu, Komei Sugiura, Kiyonori Ohtake,
Chiori Hori, Hideki Kashioka, Hisashi Kawai and Satoshi Nakamura
MASTAR Project, NICT
Kyoto, Japan.
teruhisa.misu@nict.go.jp

Abstract

This paper presents a spoken dialogue framework that helps users in making decisions. Users often do not have a definite goal or criteria for selecting from a list of alternatives. Thus the system has to bridge this knowledge gap and also provide the users with an appropriate alternative together with the reason for this recommendation through dialogue. We present a dialogue state model for such decision making dialogue. To evaluate this model, we implement a trial sightseeing guidance system and collect dialogue data. Then, we optimize the dialogue strategy based on the state model through reinforcement learning with a natural policy gradient approach using a user simulator trained on the collected dialogue corpus.

1 Introduction

In many situations where spoken dialogue interfaces are used, information access by the user is not a goal in itself, but a means for decision making (Polifroni and Walker, 2008). For example, in a restaurant retrieval system, the user's goal may not be the extraction of price information but to make a decision on candidate restaurants based on the retrieved information.

This work focuses on how to assist a user who is using the system for his/her decision making, when he/she does not have enough knowledge about the target domain. In such a situation, users are often unaware of not only what kind of information the system can provide but also their own preference or factors that they should emphasize. The system, too, has little knowledge about the user, or where his/her interests lie. Thus, the system has to bridge such gaps by sensing (potential) preferences of the user and recommend information that the user would be interested in, considering a trade-off with the length of the dialogue.

We propose a model of dialogue state that considers the user's preferences as well as his/her knowledge about the domain changing through a decision making dialogue. A user simulator is trained on data collected with a trial sightseeing system. Next, we optimize the dialogue strategy of the system via reinforcement learning (RL) with a natural policy gradient approach.

2 Spoken decision making dialogue

We assume a situation where a user selects from a given set of alternatives. This is highly likely in real world situations; for example, the situation wherein a user selects one restaurant from a list of candidates presented by a car navigation system. In this work, we deal with a sightseeing planning task where the user determines the sightseeing spot to visit, with little prior knowledge about the target domain. The study of (Ohtake et al., 2009), which investigated human-human dialogue in such a task, reported that such consulting usually consists of a sequence of information requests from the user, presentation and elaboration of information about certain spots by the guide followed by the user's evaluation. We thus focus on these interactions.

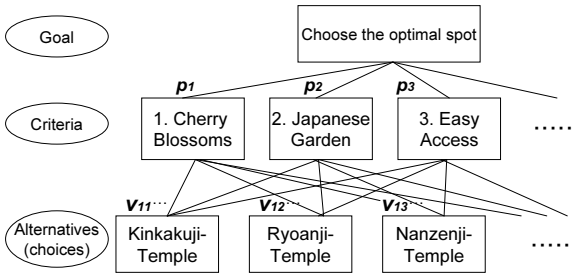


Figure 1: Hierarchy structure for sightseeing guidance dialogue

Several studies have featured decision support systems in the operations research field, and the typical method that has been employed is the Analytic Hierarchy Process (Saaty, 1980) (AHP). In AHP, the problem is modeled as a hierarchy that consists of the decision goal, the alternatives for achieving it, and the criteria for evaluating these alternatives. An example hierarchy using these criteria is shown in Figure 1.

For the user, the problem of making an optimal decision can be solved by fixing a weight vector $\mathbf{P}_{user} = (p_1, p_2, \ldots, p_M)$ for criteria and local weight matrix $\mathbf{V}_{user} = (v_{11}, v_{12}, \ldots, v_{1M}, \ldots, v_{NM})$ for alternatives in terms of the criteria. The optimal alternative is then identified by selecting the spot k with the maximum priority of $\sum_{m=1}^{M} p_m v_{km}$. In typical AHP methods, the procedure of fixing these weights is often conducted through pairwise comparisons for all the possible combinations of criteria and spots in terms of the criteria, followed by weight tuning based on the results of these comparisons (Saaty, 1980). However, this methodology cannot be directly applied to spoken dialogue systems. The information about the spot in terms of the criteria is not known to the users, but is obtained only via navigating through the system's information. In addition, spoken dialogue systems usually handle several candidates and criteria, making pairwise comparison a costly affair.

We thus consider a spoken dialogue framework that estimates the weights for the user's preference (potential preferences) as well as the user's knowledge

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 221–224,
The University of Tokyo, September 24-25, 2010.

about the domain through interactions of information retrieval and navigation.

3 Decision support system with spoken dialogue interface

The dialogue system we built has two functions: answering users' information requests and recommending information to them. When the system is requested to explain about the spots or their determinants, it explains the sightseeing spots in terms of the requested determinant. After satisfying the user's request, the system then provides information that would be helpful in making a decision (e.g., instructing what the system can explain, recommending detailed information of the current topic that the user might be interested in, etc.). Note that the latter is optimized via RL (see Section 4).

3.1 Knowledge base

Our back-end DB consists of 15 sightseeing spots as alternatives and 10 determinants described for each spot. We select determinants that frequently appear in the dialogue corpus of (Ohtake et al., 2009) (e.g. cherry blossoms, fall foliage). The spots are annotated in terms of these determinants if they apply to them. The value of the evaluation e_{nm} is "1" when the spot n applies to the determinant m and "0" when it does not.

3.2 System initiative recommendation

The content of the recommendation is determined based on one of the following six methods:

1. **Recommendation of determinants based on the currently focused spot (Method 1)**
   This method is structured on the basis of the user's current focus on a particular spot. Specifically, the system selects several determinants related to the current spot whose evaluation is "1" and presents them to the user.
2. **Recommendation of spots based on the currently focused determinant (Method 2)**
   This method functions on the basis of the focus on a certain specific determinant.
3. **Open prompt (Method 3)**
   The system does not make a recommendation, and presents an open prompt.
4. **Listing of determinants 1 (Method 4)**
   This method lists several determinants to the user in ascending order from the low level user knowledge $\mathbf{K}_{sys}$ (that the system estimates). ($\mathbf{K}_{sys}$, $\mathbf{P}_{sys}$, p_m and $Pr(p_m = 1)$ are defined and explained in Section 4.2.)
5. **Listing of determinants 2 (Method 5)**
   This method also lists the determinants, but the order is based on the user's high preference $\mathbf{P}_{sys}$ (that the system estimates).
6. **Recommendation of user's possibly preferred spot (Method 6)**
   The system recommends a spot as well as the determinants that the users would be interested in based on the estimated preference $\mathbf{P}_{sys}$. The system selects one spot k with a maximum value of $\sum_{m=1}^{M} Pr(p_m = 1) \cdot e_{k,m}$. This idea is based on collaborative filtering which is often used for recommender systems (Breese et al., 1998). This method will be helpful to users if the system successfully estimates the user's preference; however, it will be irrelevant if the system does not.

We will represent these recommendations through a dialogue act expression, ($ca_{sys}\{sc_{sys}\}$), which consists of a communicative act ca_{sys} and the semantic content sc_{sys}. (For example $Method1\{(Spot_5), (Det_3, Det_4, Det5)\}$, $Method3\{NULL, NULL\}$, etc.)

4 Optimization of dialogue strategy

4.1 Models for simulating a user

We introduce a user model that consists of a tuple of knowledge vector $\mathbf{K}_{user}$, preference vector $\mathbf{P}_{user}$, and local weight matrix $\mathbf{V}_{user}$. In this paper, for simplicity, a user's preference vector or weight for determinants $\mathbf{P}_{user} = (p_1, p_2, \ldots, p_M)$ is assumed to consist of binary parameters. That is, if the user is interested in (or potentially interested in) the determinant m and emphasizes it when making a decision, the preference p_m is set to "1". Otherwise, it is set to "0". In order to represent a state that the user has potential preference, we introduce a knowledge parameter $\mathbf{K}_{user} = (k_1, k_2, \ldots, k_M)$ that shows if the user has the perception that the system is able to handle or he/she is interested in the determinants. k_m is set to "1" if the user knows (or is listed by system's recommendations) that the system can handle determinant m and "0" when he/she does not. For example, the state that the determinant m is the potential preference of a user (but he/she is unaware of that) is represented by ($k_m = 0$, $p_m = 1$). This idea is in contrast to previous research which assumes some fixed goal observable by the user from the beginning of the dialogue (Schatzmann et al., 2007). A user's local weight v_{nm} for spot n in terms of determinant m is set to "1", when the system lets the user know that the evaluation of spots is "1" through recommendation Methods 1, 2 and 6.

We constructed a user simulator that is based on the statistics calculated through an experiment with the trial system (Misu et al., 2010) as well as the knowledge and preference of the user. That is, the user's communicative act ca^t_{user} and the semantic content sc^t_{user} for the system's recommendation a^t_{sys} are generated based on the following equation:

$$\begin{aligned} &Pr(ca^t_{user}, sc^t_{user} | ca^t_{sys}, sc^t_{sys}, \mathbf{K}_{user}, \mathbf{P}_{user}) \\ &\quad = Pr(ca^t_{user} | ca^t_{sys}) \\ &\qquad \cdot Pr(sc^t_{user} | \mathbf{K}_{user}, \mathbf{P}_{user}, ca^t_{user}, ca^t_{sys}, sc^t_{sys}) \end{aligned}$$

This means that the user's communicative act ca_{user} is sampled based on the conditional probability of $Pr(ca^t_{user}|ca^t_{sys})$ in (Misu et al., 2010). The semantic content sc_{user} is selected based on the user's preference $\mathbf{P}_{user}$ under current knowledge about the determinants $\mathbf{K}_{user}$. That is, the sc is sampled from the determinants within the user's knowledge ($k_m = 1$) based on the probability that the user requests the determinant of his/her preference/non-preference, which is also calculated from the dialogue data of the trial system.

4.2 Dialogue state expression

We defined the state expression of the user in the previous section. However the problem is that for the system, the state ($\mathbf{P}_{user}, \mathbf{K}_{user}, \mathbf{V}_{user}$) is not observable, but is only estimated from the interactions with the user. Thus, this model is a partially observable Markov decision process (POMDP) problem. In order to estimate unobservable properties of a POMDP

Priors of the estimated state:
- Knowledge: $\mathbf{K}_{sys} = (0.22, 0.01, 0.02, 0.18, \dots)$
- Preference: $\mathbf{P}_{sys} = (0.37, 0.19, 0.48, 0.38, \dots)$

Interactions (observation):
- System recommendation:
  $a_{sys} = Method1\{(Spot_5), (Det_1, Det_3, Det4)\}$
- User query:
  $a_{user} = Accept\{(Spot_5), (Det_3)\}$

Posterior of the estimated state:
- Knowledge: $\mathbf{K}_{sys} = (1.00, 0.01, 1.00, 1.00, \dots)$
- Preference: $\mathbf{P}_{sys} = (0.26, 0.19, 0.65, 0.22, \dots)$

User's knowledge acquisition:
- Knowledge: $\mathbf{K}_{user} \leftarrow \{k_1 = 1, k_3 = 1, k_4 = 1\}$
- Local weight: $\mathbf{V}_{user} \leftarrow \{v_{51} = 1, v_{53} = 1, v_{54} = 1\}$

Figure 2: Example of state update

and handle the problem as an MDP, we introduce the system's inferential user knowledge vector $\mathbf{K}_{sys}$ or probability distribution (estimate value) $\mathbf{K}_{sys} = (Pr(k_1 = 1), Pr(k_2 = 1), \dots, Pr(k_M = 1))$ and that of preference $\mathbf{P}_{sys} = (Pr(p_1 = 1), Pr(p_2 = 1), \dots, Pr(p_M = 1))$.

The dialogue state DS^{t+1} or estimated user's dialogue state of the step $t+1$ is assumed to be dependent only on the previous state DS^t, as well as the interactions $I^t = (a^t_{sys}, a^t_{user})$.

The estimated user's state is represented as a probability distribution and is updated by each interaction. This corresponds to representing the user types as a probability distribution, whereas the work of (Komatani et al., 2005) classifies users to several discrete user types. The estimated user's preference $\mathbf{P}_{sys}$ is updated when the system observes the interaction I^t. The update is conducted based on the following Bayes' theorem using the previous state DS^t as a prior.

$$Pr(p_m = 1|I^t) = \frac{Pr(I^t|p_m=1)Pr(p_m=1)}{Pr(I^t|p_m=1)Pr(p_m=1)+Pr(I^t|(p_m=0))Pr(1-Pr(p_m=1))}$$

Here, $Pr(I^t|p_m = 1), Pr(I^t|(p_m = 0)$ to the right side was obtained from the dialogue corpus of (Misu et al., 2010). This posterior is then used as a prior in the next state update using interaction I^{t+1}. An example of this update is illustrated in Figure 2.

4.3 Reward function

The reward function that we use is based on the number of agreed attributes between the user preference and the decided spot. Users are assumed to determine the spot based on their preference $\mathbf{P}_{user}$ under their knowledge $\mathbf{K}_{user}$ (and local weight for spots $\mathbf{V}_{user}$) at that time, and select the spot k with the maximum priority of $\sum_m k_k \cdot p_k \cdot v_{km}$. The reward $\mathbf{R}$ is then calculated based on the improvement in the number of agreed attributes between the user's actual (potential) preferences and the decided spot k over the expected agreement by random spot selection.

$$R = \sum_{m=1}^{M} p_m \cdot e_{k,m} - \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} p_m \cdot e_{n,m}$$

For example, if the decided spot satisfies three preferences and the average agreement of the agreement by random selection is 1.3, then the reward is 1.7.

4.4 Optimization by reinforcement learning

The problem of system recommendation generation is optimized through RL. The MDP **(S, A, R)** is defined as follows. The state parameter $\mathbf{S} = (s_1, s_2, \dots, s_I)$ is generated by extracting the features of the current dialogue state DS^t. We use the following 29 features [1]. 1. Parameters that indicate the # of interactions from the beginning of the dialogue. This is approximated by five parameters using triangular functions. 2. User's previous communicative act (1 if $a^{t-1}_{user} = x_i$, otherwise 0). 3. System's previous communicative act (1 if $a^{t-1}_{sys} = y_j$, otherwise 0). 4. Sum of the estimated user knowledge about determinants ($\sum_{n=1}^{N} Pr(k_n = 1)$). 5. Number of presented spot information. 6. Expectation of the probability that the user emphasizes the determinant in the current state ($Pr(k_n = 1) \times Pr(p_n = 1)$) (10 parameters). The action set $\mathbf{A}$ consists of the six recommendation methods shown in subsection 3.2. Reward $\mathbf{R}$ is given by the reward function of subsection 4.3.

A system action a_{sys} (ca_{sys}) is sampled based on the following soft-max (Boltzmann) policy.

$$\pi(a_{sys} = k|\mathbf{S}) = Pr(a_{sys} = k|\mathbf{S}, \mathbf{\Theta}) = \frac{\exp(\sum_{i=1}^{I} s_i \cdot \theta_{ki})}{\sum_{j=1}^{J} \exp(\sum_{i=1}^{I} s_i \cdot \theta_{ji})}$$

Here, $\mathbf{\Theta} = (\theta_{11}, \theta_{12}, \dots \theta_{1I}, \dots, \theta_{JI})$ consists of J (# actions) $\times$ I (# features) parameters. The parameter θ_{ji} works as a weight for the i-th feature of the action j and determines the likelihood that the action j is selected. This $\mathbf{\Theta}$ is the target of optimization by RL. We adopt the Natural Actor Critic (NAC) (Peters and Schaal, 2008), which adopts a natural policy gradient method as the policy optimization method.

4.5 Experiment by dialogue simulation

For each simulated dialogue session, a simulated user ($\mathbf{P}_{user}, \mathbf{K}_{user}, \mathbf{V}_{user}$) is sampled. A preference vector $\mathbf{P}_{user}$ of the user is generated so that he/she has four preferences. As a result, four parameters in $\mathbf{P}_{user}$ are "1" and the others are "0". This vector is fixed throughout the dialogue episode. This sampling is conducted based on the rate proportional to the percentage of users who emphasize it for making decisions (Misu et al., 2010). The user's knowledge $\mathbf{K}_{user}$ is also set based on the statistics of the "percentage of users who stated the determinants before system recommendation". For each determinant, we sample a random valuable r that ranges from "0" to "1", and k_m is set to "1" if r is smaller than the percentage. All the parameters of local weights $\mathbf{V}_{user}$ are initialized to "0", assuming that users have no prior knowledge about the candidate spots. As for system parameters, the estimated user's preference $\mathbf{P}_{sys}$ and knowledge $\mathbf{K}_{sys}$ are initialized based on the statistics of our trial system (Misu et al., 2010).

We assumed that the system does not misunderstand the user's action. Users are assumed to continue a dialogue session for 20 turns[2], and episodes are sampled using the policy π at that time and the user simulator

[1] Note that about half of them are continuous variables and that the value function cannot be denoted by a lookup table.

[2] In practice, users may make a decision at any point once they are satisfied collecting information. And this is the reason why we list the rewards in the early dialogue stage in

Table 1: Comparison of reward with baseline methods

	Reward ($\pm std$)			
Policy	T = 5	T = 10	T = 15	T = 20
NAC	0.96 (0.53)	1.04 (0.51)	1.12 (0.50)	1.19 (0.48)
B1	0.02 (0.42)	0.13 (0.54)	0.29 (0.59)	0.34 (0.59)
B2	0.46 (0.67)	0.68 (0.65)	0.80 (0.61)	0.92 (0.56)

Table 2: Comparison of reward with discrete dialogue state expression

	Reward ($\pm std$)			
State	T = 5	T = 10	T = 15	T = 20
PDs	0.96 (0.53)	1.04 (0.51)	1.12 (0.50)	1.19 (0.48)
Discrete	0.89 (0.60)	0.97 (0.56)	1.03 (0.54)	1.10 (0.52)

Table 3: Effect of estimated preference and knowledge

	Reward ($\pm std$)			
Policy	T = 5	T = 10	T = 15	T = 20
Pref+Know	0.96 (0.53)	1.04 (0.51)	1.12 (0.50)	1.19 (0.48)
Pref only	0.94 (0.57)	0.96 (0.55)	1.02 (0.55)	1.09 (0.53)
Know only	0.96 (0.59)	1.00 (0.56)	1.08 (0.53)	1.15 (0.51)
No Pref or Know	0.93 (0.57)	0.96 (0.55)	1.02 (0.53)	1.08 (0.52)

of subsection 4.1. In each turn, the system is rewarded using the reward function of subsection 4.3. The policy (parameter $\mathbf{\Theta}$) is updated using NAC in every 2,000 dialogues.

4.6 Experimental result

The policy was fixed at about 30,000 dialogue episodes. We analyzed the learned dialogue policy by examining the value of weight parameter $\mathbf{\Theta}$. We compared the parameters of the trained policy between actions[3]. The weight of the parameters that represent the early stage of the dialogue was large in Methods 4 and 5. On the other hand, the weight of the parameters that represent the latter stage of the dialogue was large in Methods 2 and 6. This suggests that in the trained policy, the system first bridges the knowledge gap between the user, estimates the user's preference, and then, recommends specific information that would be useful to the user.

Next, we compared the trained policy with the following baseline methods.

1. **No recommendation (B1)**
   The system only provides the requested information and does not generate any recommendations.
2. **Random recommendation (B2)**
   The system randomly chooses a recommendation from six methods.

The comparison of the average reward between the baseline methods is listed in Table 1. Note that the oracle average reward that can be obtained only when the user knows all knowledge about the knowledge base (it requires at least 50 turns) was 1.45. The reward by the strategy optimized by NAC was significantly better than that of baseline methods ($n = 500, p < .01$).

We then compared the proposed method with the case where estimated user's knowledge and preference are represented as discrete binary parameters instead of probability distributions (PDs). That is, the estimated user's preference p_m of determinant m is set to "1" when the user requested the determinant, otherwise it is "0". The estimated user's knowledge k_m is set to "1" when the system lets the user know the determinant, otherwise it is "0". Another dialogue strategy was trained using this dialogue state expression. This result is shown in Table 2. The proposed method that represents the dialogue state as a probability distribution outperformed ($p < .01$ (T=15,20)) the method using a discrete state expression.

We also compared the proposed method with the case where either one of estimated preference or knowledge was used as a feature for dialogue state in order to carefully investigate the effect of these factors. In the proposed method, expectation of the probability that the user emphasizes the determinant ($Pr(k_n = 1) \times Pr(p_n = 1)$) was used as a feature of dialogue state. We evaluated the performance of the cases where the estimated knowledge $Pr(k_n = 1)$ or estimated preference $Pr(p_n = 1)$ was used instead of the expectation of the probability that the user emphasizes the determinant. We also compared with the case where no preference/knowledge feature was used. This result is shown in Table 3. We confirmed that significant improvement ($p < .01$ (T=15,20)) was obtained by taking into account the estimated knowledge of the user.

5 Conclusion

In this paper, we presented a spoken dialogue framework that helps users select an alternative from a list of alternatives. We proposed a model of dialogue state for spoken decision making dialogue that considers knowledge as well as preference of the user and the system, and its dialogue strategy was trained by RL. We confirmed that the learned policy achieved a better recommendation strategy over several baseline methods.

Although we dealt with a simple recommendation strategy with a fixed number of recommendation components, there are many possible extensions to this model. The system is expected to handle a more complex planning of natural language generation. We also need to consider errors in speech recognition and understanding when simulating dialogue.

References

J. Breese, D. Heckerman, and C. Kadie. 1998. "empirical analysis of predictive algorithms for collaborative filtering". In *"Proc. the 14th Annual Conference on Uncertainty in Artificial Intelligence"*, pages 43–52.

K. Komatani, S. Ueno, T. Kawahara, and H. Okuno. 2005. User Modeling in Spoken Dialogue Systems to Generate Flexible Guidance. *User Modeling and User-Adapted Interaction*, 15(1):169–183.

T. Misu, K. Ohtake, C. Hori, H. Kashioka, H. Kawai, and S. Nakamura. 2010. Construction and Experiment of a Spoken Consulting Dialogue System. In *Proc. IWSDS*.

K. Ohtake, T. Misu, C. Hori, H. Kashioka, and S. Nakamura. 2009. Annotating Dialogue Acts to Construct Dialogue Systems for Consulting. In *Proc. The 7th Workshop on Asian Language Resources*, pages 32–39.

J. Peters and S. Schaal. 2008. Natural Actor-Critic. *Neurocomputing*, 71(7-9):1180–1190.

J. Polifroni and M. Walker. 2008. Intensional Summaries as Cooperative Responses in Dialogue: Automation and Evaluation. In *Proc. ACL/HLT*, pages 479–487.

T. Saaty. 1980. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. Mcgraw-Hill.

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. 2007. Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc. HLT/NAACL*.

the following subsections. In our trial system, the dialogue length was 16.3 turns with a standard deviation of 7.0 turns.

[3]The parameters can be interpreted as the size of the contribution for selecting the action.

The vocal intensity of turn-initial cue phrases in dialogue

Anna Hjalmarsson
Department of Speech Music and Hearing, KTH
Stockholm, Sweden
annah@speech.kth.se

Abstract

The present study explores the vocal intensity of turn-initial cue phrases in a corpus of dialogues in Swedish. Cue phrases convey relatively little propositional content, but have several important pragmatic functions. The majority of these entities are frequently occurring monosyllabic words such as "eh", "mm", "ja". Prosodic analysis shows that these words are produced with higher intensity than other turn-initial words are. In light of these results, it is suggested that speakers produce these expressions with high intensity in order to claim the floor. It is further shown that the difference in intensity can be measured as a dynamic inter-speaker relation over the course of a dialogue using the end of the interlocutor's previous turn as a reference point.

1 Introduction

In dialogue, interlocutors produce speech incrementally and on-line as the dialogue progresses. Articulation can be initiated before the speaker has a complete plan of what to say (Pechmann, 1989). When speaking, processes at all levels (e.g. semantic, syntactic, phonologic and articulatory) work in parallel to render the utterance. This processing strategy is efficient, since the speaker may employ the time devoted to articulating an early part of an utterance to plan the rest.

Speakers often initiate new turns with *cue phrases* – standardized lexical or non-lexical expressions such as "ehm" "okay", "yeah", and "but" (c.f. Gravano, 2009). Cue phrases (or *discourse markers*) are linguistic devices used to signal relations between different segments of speech (for an overview see Fraser, 1996). These devices convey relatively little propositional content, but have several important pragmatic functions. For example, these words provide feedback and signal how the upcoming utterance relates to previous context. Another important function is to claim the conversational floor (c.f. Levinson, 1983).

With these fundamental properties of language production in mind, it is proposed that turn-initial cue phrases can be used in spoken dialogue systems to initiate new turns, allowing the system additional time to generate a complete response. This approach was recently explored in a user study with a dialogue system that generates turn-initial cue phrases incrementally (Skantze & Hjalmarsson, in press). Results from this experiment show that an incremental version that used turn-initial cue phrases had shorter response times and was rated as more efficient, more polite and better at indicating when to speak than a non-incremental implementation of the same system. The present study carries on this research, investigating acoustic parameters of turn-initial cue phrases in order to build a dialogue system that sounds convincing intonation wise.

Another aim of this study was to explore if the vocal intensity of the other speaker's immediately preceding speech can be used as a reference point in order to measure intensity as an inter-speaker relation over the course of a dialogue. Thus, in addition to measuring overall differences in intensity, the relative difference between the first token of a new turn and the last token of the immediately preceding turn was measured. This dynamic approach, if proven feasible, allows spoken dialogue system designers to adjust the system's vocal intensity on-line in order to accommodate variations in the surrounding acoustic environment.

2 Related work

There are a few examples of research that have manipulated intensity to signal pragmatic functions. For example, Ström & Seneff (2000) increases intensity in order to signal that user

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 225–228,
The University of Tokyo, September 24-25, 2010.

barge-ins are disallowed in particular dialogue states. Theoretical support for such manipulations is provided by an early line of research on interruptions in dialogue (Meltzer et al., 1971). Meltzer et al. (1971) propose that the outcome of speech overlaps is affected by prosodic characteristics and show that the greater the increase in amplitude, the greater the likelihood of "interruption success". Moreover, it is show that the success of interruptions, that is who retains the floor, is based on how much higher the intensity of the interruption is compared to the previous speaker's intensity or compared to the speaker's own intensity at the end of that speaker's previous speaker turn.

Measuring inter-speaker relative intensity is further motivated by research that suggests that speakers adjust their vocal intensity online over the course of a dialogue in order to accommodate the surrounding acoustic context. For example, speakers tend to raise their voice unintentionally when background noise increases to enhance their audibility; this is the so-called Lombard effect (Pick et al., 1989). Moreover, speakers adjust intensity based on their conversational partners (Natale, 1975) and the distance to their listeners (Healey et al., 1997).

3 Method

3.1 Data: The DEAL corpus

DEAL is a dialogue system that is currently being developed at the department of Speech, Music and Hearing, KTH (Wik & Hjalmarsson, 2009). The aim of the DEAL dialogue system is to provide conversation training for second language learners of Swedish. The scene of DEAL is set at a flea market where a talking animated persona is the owner of a shop selling used goods.

The dialogue data used as a basis for the data analyzes presented in this paper were human-human dialogues, collected in a recording environment set up to mimic the interaction in the DEAL domain. The dialogue collected were informal, human-human, face-to-face conversation in Swedish. The recordings were made with close talk microphones with six subjects (four male and two female). In total, eight dialogues were collected. Each dialogue was about 15 minutes, making for about two hours of speech in total in the corpus. The dialogues were transcribed orthographically and annotated for entities such as laughter, lip-smacks, breathing and hemming. The transcripts from the dialogues were time-aligned with the speech signal. This was done using forced alignment with subsequent manual verification of the timings. The dialogues were also segmented into *speaker turns*. A speaker turn here is a segment of speech of arbitrary length surrounded by another speaker's vocalization. All together, the dialogues contained 2036 speaker turns.

The corpus was also annotated for cue phrases using 11 functional categories. The definition of cue phrases used for annotation of the DEAL corpus was broad and all types of vocalizations that the speakers use to hold the dialogue together at different communicative levels were included. Cue phrase annotation was designed as a two-fold task: (i) to decide if a word was a cue phrase or not – a binary task, and (ii) to select its functional class according to the annotation scheme. The annotators could see the transcriptions and listen to the recordings while labelling. The kappa coefficient for task (i) was 0.87 ($p<.05$). The kappa coefficient for (ii) was 0.82 ($p<.05$). For a detailed description of the cue phrase categories and their annotation, see (Hjalmarsson, 2008).

3.2 Data analysis

The first word in each turn was extracted and analyzed. Here, a word is all annotated tokens in the corpus except breathing, lip-smacks, and laughter, which are all relevant, but outside the scope of this study. 1137 (57%) words were annotated as some type of cue phrase, and 903 (43 %) were other words. The turn-initial cue phrases were annotated with different cue phrase categories. 587 (28%) turn-initial words were annotated as either RESPONSIVE, RESPONSIVE DISPREFERENCE or RESPONSIVE NEW INFORMATION. The annotation of these was based on the interpretation of the speakers' attitudes, expressing either neutral feedback (RESPONSIVE), non-agreement (RESPONSIVE DISPREFERENCE) or surprise (RESPONSIVE NEW INFORMATION). The RESPONSIVES were most frequently realized as either "ja", "a", and "mm" (Eng: "yeah", "mm").

Furthermore, 189 (9%) of all turn-initial words were annotated as CONNECTIVES. The connective cue phrase categories indicate how the new utterance relates to previous context. For example, these signal whether the upcoming speaker turn is *additive*, *contrastive* or *alternative* to previous context. Examples of these categories are "och" (Eng: "and"), "men" (Eng: "but") and "eller" (Eng: "or"), respectively.

A third category of cue phrases in a turn-initial position was filled pauses (57, 3%). Whereas filled pause may not typically be considered as cue phrases, these elements have similar characteristics. For example, fillers provide important pragmatic information that listeners attend and adjust their behaviour according to. For example, a corpus study of Dutch fillers showed that these tokens highlight discourse structure (Swertz, 1998). Frequently occurring filler words in the corpus were "eh" and "ehm".

The majority of the turn-initial cue phrases were high frequency monosyllabic words, which are typically not associated with stress, although on listening, they give the impression of being louder than other turn-initial vocalizations. To verify this observation, the intensity in decibel of the first word of each turn was extracted using Snack (www.speech.kth.se/snack). In order to explore the vocal intensity as an inter-speaker relation over the course of the dialogue, the average intensity of the last word of all turns was extracted. The motivation of this approach is to use the previous speaker's voice intensity as a reference point. Thus, in order to avoid the need for global analysis over speakers and dialogues, only the (un-normalized) difference in intensity between the last word of the immediately preceding turn and the first word of a new turn was calculated.

All turns following a one word only turn from the other speaker were excluded as an approximation to avoid speech following backchannel responses. 300 (33%) of the speaker changes contained overlapping speech. These overlaps were excluded from the data analysis since the recordings were not completely channel-separated and crosstalk could conceivably interfere with the results.

Since the distance between the lips and the microphone was not controlled for during the recordings, the values were first normalized per speaker and dialogue (each value was shifted by the mean value per speaker and dialogue).

4 Results

Figure 1 presents the average normalized intensity for turns initiated with cue phrases and other words.

An independent samples t-test was conducted between the intensity of turns initiated with cue phrases and other turn-initial words. There was a significant difference in intensity between turns initiated with cue phrases (M=3.20 dB, SD=6.99) and turns initiated with other words (M=-4.20 dB, SD=9.98), t(597)=10.55, p<.000. This shows that, on average, turns initiated with cue phrases were significantly louder (on average 6 dB) than turns initiated with other words.

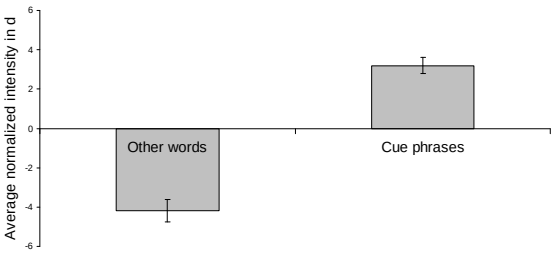


Figure 1 : Average normalized vocal intensity in dB for turn-initial words. Error bars represents the standard error.

In order to explore the vocal intensity as an inter-speaker relation the difference in voice intensity between a new turn and the end of the immediately preceding turn was extracted. The inter-speaker differences in intensity for turn-initial cue phrases and other words are presented in Figure 2.

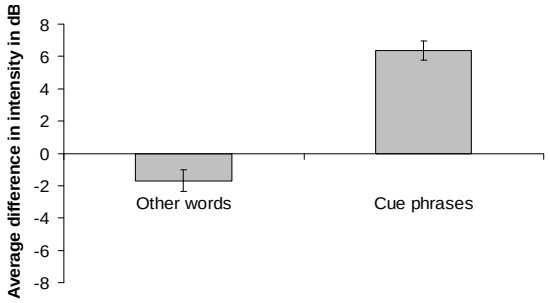


Figure 2 Average difference in intensity (in dB) for turn-initial words. Error bars represents the standard error.

An independent samples t-test was conducted to explore the difference in voice intensity as an inter-speaker relation. There was a significant difference in intensity between turns initiated with cue phrases (M=6.14 dB, SD=11.86) and turns initiated with other words (M=-1.52 dB, SD=13.07); t(595)=7.48, p<.000. This shows that the increase in intensity was significantly larger for turns initiated with cue phrases (about 7 dB) than for turns initiated with other words.

5 Discussion

This paper presents analyses of the intensity of turn-initial words. It shown that turns are frequently initiated with cue phrases (about 55% of the turns in the DEAL corpus). The majority of

these consist of high frequency monosyllabic words such as “yes”, “mm” and “okay”. The most frequent turn-initial words that were not annotated as cue phrases were “den” (Eng: “it”), “vad” (Eng: “what”), and “jag” (Eng: “I”). Thus, similar to turn-initial cue phrases, this category contains high-frequency monosyllabic words, items that are not typically associated with prosodic stress. Yet, the results show that turn-initial cue phrases are produced with higher intensity than other turn-initial words are. In the light of previous research, which suggests that increased intensity have turn-claiming functions, one can speculate that speakers produce talkspurt-initial cue phrases with increased intensity in order to claim the floor convincingly before having formulated a complete utterance.

It is further argued that turn-initial cue phrases can be used in dialogue systems capable of incremental speech production. Such words can be used to initiate turns once the user has stopped speaking, allowing the system more time to process input without response delays.

Finally, it is suggested that intensity may be better modelled relative to the intensity of the immediately preceding speech rather than in absolute of speaker-normalized terms. Speakers adjust their intensity to the current acoustical environment, and such a dynamic inter-speaker relative model may accommodate the current acoustic context over the course of a dialogue. In support of this approach, the present study shows that the increase in intensity can be calculated dynamically over the dialogue using the end of the previous speaker’s turn as a reference point. Inter-speaker relative measures are also motivated practically. Extracting objective measures of intensity is problematic since contextual factors such as the distance between the microphone and the lips are difficult to control between dialogues and speakers, but the effects are mitigated by dynamic and relative measures. This is not to say that measuring intensity over the course of a single dialogue is trivial. Variation due to for example unforeseen alterations of the distance between the lips and the microphone during the dialogue are still problematic, but it is less of a problem within a session than between different sessions.

Acknowledgments

This research was carried out at Centre for Speech Technology, KTH. Funding was provided by Riksbankens Jubileumsfond (RJ) project P09-0064:1-E Prosody in conversation and the Graduate School for Language Technology (GSLT). Many thanks to Rolf Carlson, Jens Edlund and Joakim Gustafson for valuable comments.

References

Fraser, B. (1996). Pragmatic markers. *Pragmatics, 6*(2), 167-190.

Gravano, A. (2009). *Turn-Taking and Affirmative Cue Words in Task-Oriented Dialogue*. Doctoral dissertation, Columbia University.

Healey, C., Jones, R., & Berky, R. (1997). Effects of perceived listeners on speakers'vocal intensity. *Journal of Voice, 11*(1), 67-73.

Hjalmarsson, A. (2008). Speaking without knowing what to say... or when to end. In *Proceedings of SIGDial 2008*. Columbus, Ohio, USA.

Levinson, S. C. (1983). *Pragmatics.* Cambridge: Cambridge University press.

Meltzer, L., Hayes, D., & Morris, M. (1971). Interruption Outcomes and Vocal Amplitude: Explorations in Social Psychophysics. *Journal of Personality and Social Psychology, 18*(3), 392-402.

Natale, M. (1975). Convergence of mean vocal intensity in dyadic communication as a function of social desirability. *Personality and Social Psychology, 32*(5), 790-804.

Pechmann, T. (1989). Incremental speech production and referential overspecification. *Linguistics, 27*, 89-110.

Pick, H. L. J., Siegel, G. M., Fox, P. W., Garber, S. R., & Kearney, J. K. (1989). Inhibiting the Lombard effect. *JASA, 85*(2), 894-900.

Skantze, G., & Hjalmarsson, A. (in press). Towards Incremental Speech Generation in Dialogue Systems. To be published in *Proceedings of SigDial*. Tokyo, Japan.

Ström, N., & Seneff, S. (2000). Intelligent barge-in in conversational systems. In *Procedings of ICSLP-00*.

Swertz, M. (1998). Filled pauses as markers of discourse structure. *Journal of Pragmatics, 30*(4), 485-496.

Wik, P., & Hjalmarsson, A. (2009). Embodied conversational agents in computer assisted language learning. *Speech communication, 51*(10), 1024-1037.

Pamini: A framework for assembling mixed-initiative human-robot interaction from generic interaction patterns

Julia Peltason and Britta Wrede
Applied Informatics, Faculty of Technology
Bielefeld University, Germany
jpeltaso, bwrede@techfak.uni-bielefeld.de

Abstract

Dialog modeling in robotics suffers from lack of generalizability, due to the fact that the dialog is heavily influenced by the tasks the robot is able to perform. We introduce interleaving interaction patterns together with a general protocol for task communication which enables us to systematically specify the relationship between dialog structure and task structure. We argue that this approach meets the requirements of advanced dialog modeling on robots and at the same time exhibits a better scalability than existing concepts.

1 Introduction

The need for interaction modeling on robots is widely acknowledged, not only for instructing them but also for enabling them to learn from humans within interaction. Yet, today's robotic systems often do not have a dedicated dialog system but employ simple command-matching techniques (e.g. (Böhme et al., 2003)). Other systems rely on finite-state based dialog managers (e.g. (Bauer et al., 2009)) that couple dialog and task management which hampers maintainability and reuse and does not scale well for more complex interaction scenarios.

On the other hand, concepts for reusable dialog frameworks have been established within the spoken dialog community for traditional information-seeking domains where the system first collects the required parameters, then presents the desired information to the user, e.g. an accommodation or travel information (e.g. (Bohus and Rudnicky, 2009)). However, these concepts are not directly transferable to robotics, due to the situated nature of human-robot interaction. A robot perceives the world and acts on it – and it is supposed to communicate about that. For instance, a robot manipulating objects needs to be able to verbalize its intended actions and give feedback about its internal state, but it also needs to react on feedback comments or corrections uttered by the human tutor. In order to equip a robot with such fine-grained communication skills, tight integration of action execution into interaction is required. In simple terms, while the dialog frameworks mentioned can be referred to as information-oriented, we need rather action-oriented approaches for robotics.

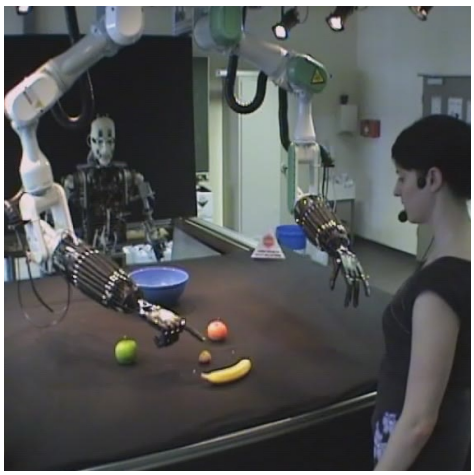


Figure 1: The Curious Robot set-up.

R: What is that?	⟨points⟩
H: This is a lemon.	
R: A melon. OK.	
H: No, I said 'lemon'!	
R: A lemon. OK.	
R: How should I grasp the lemon?	
H: That way.	⟨demonstrates grip⟩
R: Alright, with the power grasp.	
R: I am going to grasp it	⟨starts grasping⟩
R: I start grasping now.	
H: Robot, what objects do you know?	
R: Lemon.	
R: OK, I finished grasping.	⟨stops grasping⟩

Table 1: A typical interaction with the Curious Robot.

2 The Pamini framework

This paper presents the Pamini (Pattern-based mixed-initiative) human-robot interaction framework. Pamini proposes a new approach for dialog

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 229–232,
The University of Tokyo, September 24-25, 2010.

modeling on robots that includes (1) a task-state protocol providing a fine-grained interface to the complex domain processing of the robotic system and (2) the concept of generic interaction patterns that support rapid prototyping of human-robot interactions and can be combined in a flexible way.

Previous versions of the Pamini framework have been applied in several mixed-initiative learning scenarios. For example, in the Home-Tour scenario a mobile robot builds up a spatial model of its environment and gradually improves its model by attempting to obtain information from the human (Peltason et al., 2009). In the Curious Robot scenario shown in figure 1, an anthropomorphic robot learns to label and grasp objects, applying a proactive dialog strategy that provides guidance for untrained users (Lütkebohle et al., 2009). A dialog excerpt is shown in table 1.

2.1 The task state protocol

A dialog system for robotics needs to coordinate with a number of components, e.g. for perceptual analysis, motor control or components generating nonverbal feedback. To realize this, we use the concept of *tasks* that can be performed by components. Tasks are described by an execution state and a task specification containing the information required for execution. A protocol specifies task states relevant for coordination and possible transitions between them as shown in figure 2. Task updates, i.e. updates of the task state and possibly the task specification, cause event notifications which are delivered to the participating components whereupon they take an appropriate action.

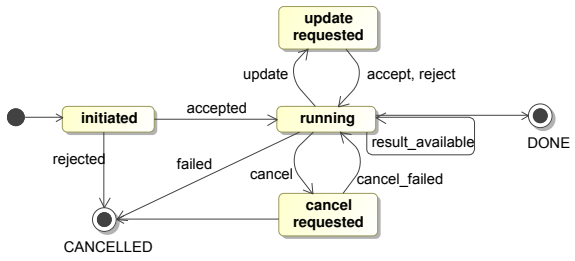


Figure 2: The task life-cycle. A task gets initiated, accepted, may be cancelled or updated, may deliver intermediate results and finally is completed. Alternatively, it can be rejected by the handling component or execution may fail.

Tight integration with action execution A robot performing e.g. a grasp action supervised by the human requires multi-step communication between the dialog system and the arm control as illustrated in figure 3. Generally, with the *accepted* state, the proposed protocol enables the dialog system to provide feedback during slow-going actions indicating the internal system state. Further, with the *update* and *result_available* states, it supports the modification of the task specification during execution and thus gives the robot the ability to react to comments, corrections and commands on-the-fly.

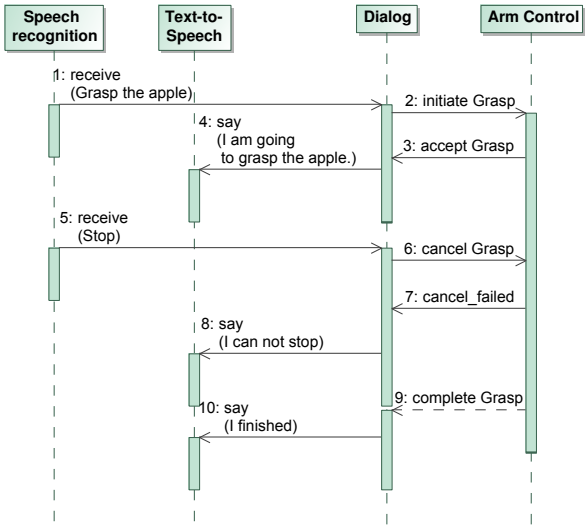


Figure 3: The component communication for a grasp action requested by the human. As the dialog manager (DLG) receives the grasp command, it *initiates* a grasp task which is *accepted* by the arm control. The DLG is notified about the task state update and acknowledges task execution. As the human commands cancelling, the DLG sets the task state *cancel*. Since the arm control fails to cancel the task, it sets the task state *cancel_failed* which the DLG reacts on by expressing an excuse. Finally the task is *completed*, and the DLG acknowledges successful task execution.

Mixed-initiative interaction The Pamini dialog manager offers dialog tasks for other components, e.g. greeting the human, informing the human about anything or conversely requesting information from the human. While *human initiative* is realized whenever input from a speech understanding component is received, *robot initiative* occurs when a system component requests a dialog task to be executed. Situation permitting, the dialog manager will accept the dialog task, go into interaction with the human, and finally complete the dialog task. Thus, it can react to the system's and the human's initiative using the same task-state protocol

Learning within interaction The task state protocol supports robotic learning within interaction by establishing mechanisms for information transfer from the dialog system to the robotic subsystem. Once information is available from the human, Pamini augments the task specification

with the new information and sets the task state *result_available*. Since this transition may be taken multiple times, given information can be corrected. Also, mixed-initiative enables *active learning*, where the learner provokes a situation providing new information instead of waiting until such situation eventually presents itself.

2.2 Interaction patterns

In an interaction, dialog acts are not unrelated events, but form coherent sequences. For example, a question is usually followed by an answer, and a request is typically either accepted or rejected. Influenced by the concepts of adjacency pairs (Schegloff and Sacks, 1973), conversation policies (Winograd, 1986) and software design patterns, we propose the concept of *interaction patterns* that describe recurring dialog structures on a high level. Interaction patterns can be formalized as transducer augmented with internal state actions, consisting of

- a set of human dialog acts H and a set of robot dialog acts R, e.g. *H.request* or *R.assert*;
- a set of incoming task events T, e.g. *accepted* or *failed*;
- a set of states S representing the interaction state;
- a set of actions A the dialog manager performs, e.g. initiating or updating a task or reset interaction;
- an input alphabet $\Sigma \subset (H \cup T)$;
- an output alphabet $\Lambda \subset R$;
- a transition function $T : S \times \Sigma^* \longrightarrow S \times A^* \times \Lambda^*$.

By admitting task events as input and internal actions that perform task initiation and update, the dialog level is linked with the domain level. The patterns have been implemented as statecharts (Harel, 1987), an extended form of finite state machines, which provides both an executable model and an understandable graphical representation as shown in figure 5. For instance, the *cancellable*

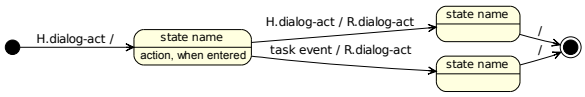


Figure 5: Interaction patterns are represented as transducer that takes as input human dialog acts and task events and produces robot dialog acts as output.

action request pattern shown in figure 4 describes an action request initiated by the human that can be cancelled during execution. The normal course of events is that the human requests the action to be executed, the dialog manager initiates the domain task, the responsible system component accepts execution so that the dialog manager will assert execution. Finally, the task is completed and the robot acknowledges. In contrast, the *correctable information request* pattern is initiated by the human. Here, on receiving the respective dialog task request, the dialog manager will ask for the desired information and accept the dialog task. Once the human provides the answer, the robot will repeat it as implicit confirmation that can be corrected if necessary. Table 2 lists all patterns that have been identified so far.

Initiated by user	Initiated by robot
Cancellable action request	Self-initiated cancellable action
Simple action request	Self-initiated simple action
Information request	Correctable information request
Interaction opening	Simple information request
Interaction closing	Clarification
Interaction restart	
System reset	

Table 2: Available interaction patterns.

Pattern configuration The patterns themselves do not determine what kind of task is to be executed or what kind of information to obtain exactly. These specifics are defined by the configuration associated with each pattern, and a concrete scenario is realized by configuring a set of patterns using a domain-specific language and registering them with the dialog manager.

In detail, it needs to be specified for the human's dialog acts what kind of (possibly multimodal) input is interpreted as a given dialog act which is done by formulating conditions over the input. For the robot's dialog acts, their surface form needs to be specified. Up to now, speech output and pointing gestures are implemented as output modalities and can be combined. Moreover, also the task communication needs to be configured, i.e. the task specification itself as well as possible task specification updates. In addition, the developer can define context variables and use them to parameterize the robot's dialog acts and in task specification updates. This is how e.g. for the robot's information request the answer is transferred from the human to the responsible system component.

Interleaving patterns during interaction During interaction, the registered patterns are employed in a flexible way by admitting patterns to be interrupted by other patterns and possibly resumed later which leads to interleaving patterns. By default, simpler patterns are permitted to be nested within temporally extended patterns. For example, it seems reasonable to permit monitoring questions uttered by the human to be embedded in the robot's slow-going grasp execution as shown

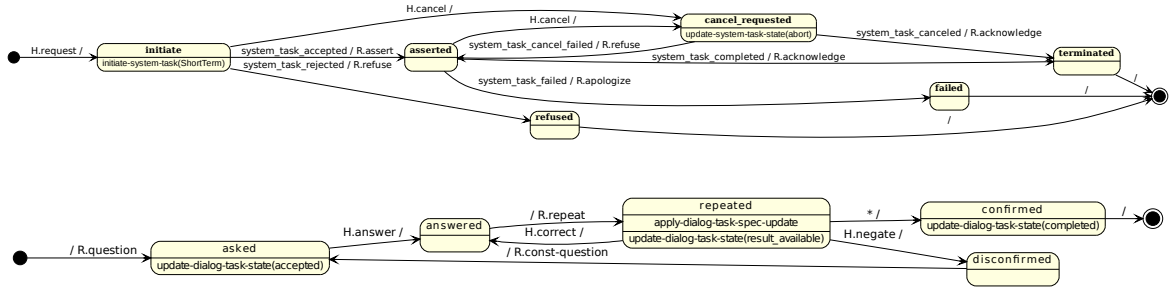


Figure 4: Two example interaction patterns. *Cancellable action request*: an action request which is initiated by the human and can be cancelled during execution. *Correctable information request*: an information request with implicit confirmation initiated by the robot where information can be corrected later if necessary.

in table 1 which equips the robot with multitasking capabilities. Interleaving is realized by organizing active patterns on a stack. Whenever an input is received, the dialog manager attempts to interpret it in the context provided by the topmost pattern. If it fails, the lower and inactive patterns are tried.

3 Discussion and Conclusion

The presented approach to dialog modeling on robots relies on the concept of interaction patterns that constitute configurable (and thus reusable) building blocks of interaction. A fine-grained task state protocol links dialog and domain level. With interleaving patterns, flexible dialog modeling is achieved that goes beyond current state-of-the-art dialog modeling on robots. Further, by encapsulating both the subtleties of dialog management and the complexity of component integration, the proposed interaction patterns support rapid prototyping of human-robot interaction scenarios.

The evaluation of the approach needs to examine framework usability, framework functionality and usability of the resulting dialogs. With respect to framework usability, we already showed that developers unfamiliar with the framework were able to build a simple interaction scenario within one hour (Peltason and Wrede, 2010). With respect to framework functionality, we demonstrated that the robot's mixed-initiative interaction capabilities enable human and robot in the Home-Tour scenario to jointly build up a common representation of their environment and even compensate for classification errors (Peltason et al., 2009). As to dialog usability, a video study indicates that the Curious Robot's proactive dialog strategy guides unexperienced users (Lütkebohle et al., 2009). Further, given a dialog system architecture that supports rapid prototyping, comparative studies become possible. Therefore, we currently prepare a study to compare the curiosity strategy with a user-directed strategy that provides more freedom but also more uncertainty to the user. Last but not least, we will evaluate the patterns themselves and pattern interleavability. Are users likely to interrupt a robot's action by asking questions or even giving new commands? Also, are there other kinds of interaction patterns that occur in a real interaction but are not captured yet?

References

A. Bauer, D. Wollherr, and M. Buss. 2009. Information retrieval system for human-robot communication asking for directions. In *International Conference on Robotics and Automation*.

H.-J. Böhme, T. Wilhelm, J. Key, C. Schauer, C. Schröter, H.-M. Groß, and T. Hempel. 2003. An approach to multimodal human-machine interaction for intelligent service robots. *Robotics and Autonomous Systems*, 44(1).

D. Bohus and A. I. Rudnicky. 2009. The ravenclaw dialog management framework: Architecture and systems. *Computer Speech & Language*, 23(3):332–361.

D. Harel. 1987. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274.

I. Lütkebohle, J. Peltason, L. Schillingmann, C. Elbrechter, B. Wrede, S. Wachsmuth, and R. Haschke. 2009. The curious robot - structuring interactive robot learning. In *International Conference on Robotics and Automation*.

J. Peltason and B. Wrede. 2010. Modeling human-robot interaction based on generic interaction patterns. In *AAAI Technical Report: Dialog with Robots*. submitted.

J. Peltason, F. Siepmann, T. Spexard, B. Wrede, M. Hanheide, and E. Topp. 2009. Mixed-initiative in human augmented mapping. In *International Conference on Robotics and Automation*.

E. A. Schegloff and H. Sacks. 1973. Opening up closings. *Semiotica*, 8(4):289–327.

T. Winograd. 1986. A language/action perspective on the design of cooperative work. In *Conference on Computer-supported cooperative work*.

Collaborating on Utterances with a Spoken Dialogue System Using an ISU-based Approach to Incremental Dialogue Management

Okko Buß, Timo Baumann, David Schlangen
Department of Linguistics
University of Potsdam, Germany
{okko|timo|das}@ling.uni-potsdam.de

Abstract

When dialogue systems, through the use of incremental processing, are not bounded anymore by strict, non-overlapping turn-taking, a whole range of additional interactional devices becomes available. We explore the use of one such device, trial intonation. We elaborate our approach to dialogue management in incremental systems, based on the Information-State-Update approach, and discuss an implementation in a micro-domain that lends itself to the use of immediate feedback, trial intonations and expansions. In an overhearer evaluation, the incremental system was judged as significantly more human-like and reactive than a non-incremental version.

1 Introduction

In human–human dialogue, most utterances have only one speaker.[1] However, the shape that an utterance ultimately takes on is often determined not just by the one speaker, but also by her addressees. A speaker intending to refer to something may start with a description, monitor while they go on whether the description appears to be understood sufficiently well, and if not, possibly extend it, rather than finishing the utterance in the form that was initially planned. This monitoring within the utterance is sometimes even made very explicit, as in the following example from (Clark, 1996):

(1) A: A man called Annegra? -
B: yeah, Allegra
A: Allegra, uh, replied and, uh, ...

In this example, A makes use of what Sacks and Schegloff (1979) called a *try marker*, a "questioning upward intonational contour, followed by a brief pause". As discussed by Clark (1996), this device is an efficient solution to the problem posed by uncertainty on the side of the speaker whether a reference is going to be understood, as it checks for understanding *in situ*, and lets the conversation partners collaborate on the utterance that is in production.

Spoken dialogue systems (SDS) typically cannot achieve the close coupling between production and interpretation that is needed for this to work, as normally the smallest unit on which they operate is the full utterance (or, more precisely, the turn). (For a discussion see e.g. (Skantze and Schlangen, 2009).) We present here an approach to managing dialogue in an incremental SDS that can handle this phenomenon, explaining how it is implemented in system (Section 4) that works in a micro-domain (which is described in Section 3). As we will discuss in the next section, this goes beyond earlier work on incremental SDS, combining the production of multimodal feedback (as in (Aist et al., 2007)) with fast interaction in a semantically more complex domain (compared to (Skantze and Schlangen, 2009)).

2 Related Work

Collaboration on utterances has not often been modelled in SDS, as it presupposes fully incremental processing, which itself is still something of a rarity in such systems. (There is work on collaborative reference (DeVault et al., 2005; Heeman and Hirst, 1995), but that focuses on written input, and on collaboration over several utterances and not within utterances.) There are two systems that are directly relevant here.

The system described in (Aist et al., 2007) is able to produce some of the phenomena that we are interested in here. The set-up is a simple reference game (as we will see, the domain we have chosen is very similar), where users can refer to objects shown on the screen, and the SDS gives continuous feedback about its understand-

[1] Though by far not all; see (Clark, 1996; Purver et al., 2009; Poesio and Rieser, 2010).

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 233–236,
The University of Tokyo, September 24-25, 2010.

ing by performing on-screen actions. While we do produce similar non-linguistic behaviour in our system, we also go beyond this by producing verbal feedback that responds to the certainty of the speaker (expressed by the use of trial intonation). Unfortunately, very little technical details are given in that paper, so that we cannot compare the approaches more fully.

Even more closely related is some of our own previous work, (Skantze and Schlangen, 2009), where we modeled fast system reactions to delivery of information in installments in a number sequence dictation domain. In a small corpus study, we found a very pronounced use of trial or installment intonations, with the first installments of numbers being bounded by rising intonation, and the final installment of a sequence by falling intonation. We made use of this fact by letting the system distinguish these situations based on prosody, and giving it different reaction possibilities (backchannel feedback vs. explicit confirmation).

The work reported here is a direct scaling up of that work. For number sequences, the notion of utterance is somewhat vague, as there are no syntactic constraints that help demarcate its boundaries. Moreover, there is no semantics (beyond the individual number) that could pose problems – the main problem for the speaker in that domain is ensuring that the signal is correctly *identified* (as in, the string could be written down), and the trial intonation is meant to provide opportunities for grounding whether that is the fact. Here, we want to go beyond that and look at utterances where it is the intended meaning whose recognition the speaker is unsure about (grounding at level 3 rather than (just) at level 2 in terms of (Clark, 1996).) This difference leads to differences in the follow up potential: where in the numbers domain, typical repair follow-ups were *repetitions*, in semantically more complex domains we can expect *expansions* or *reformulations*.

3 The Puzzle Micro-Domain

To investigate these issues in a controlled setting, we chose a domain that makes complex and possibly underspecified references likely, and that also allows a combination of linguistic and non-linguistic feedback. In this domain, the user's goal is to instruct the system to pick up and manipulate Tetris-like puzzle pieces, which are shown on the screen. We recorded human–human as well as human–(simulated) machine interactions in this domain, and indeed found frequent use of "packaging" of instructions, and immediate feedback, as in (2) (arrow indicating intonation).

(2)
IG-1: The cross in the corner↗ ...
IF-2: erm
IG-3: the red one .. yeah
IF-4: [moves cursor]
IG-5: take that.

We chose these as our target phenomena for the implementation: intra-utterance hesitations, possibly with trial intonation (as in line 2);[2] immediate execution of actions (line 4), and their grounding role as display of understanding ("yeah" in line 3). The system controls the mouse cursor, e.g. moving it over pieces once it has a good hypothesis about a reference; other actions are visualised similarly.

4 Implementation

4.1 Overview

Our system is realised as a collection of incremental processing modules in the InproToolKit (Schlangen et al., 2010), a middle-ware package that implements some of the features of the model of incremental processing of (Schlangen and Skantze, 2009). The modules used in the implementation will be described briefly below.

4.2 ASR, Prosody, Floor Tracker & NLU

For speech recognition, we use Sphinx-4 (Walker et al., 2004), with our own extensions for incremental speech recognition (Baumann et al., 2009), and our own domain-specific acoustic model. For the experiments described here, we used a recognition grammar.

Another module performs online prosodic analysis, based on pitch change, which is measured in semi-tone per second over the turn-final word, using a modified YIN (de Cheveigné and Kawahara, 2002). Based on the slope of the f_0 curve, we classify pitch as rising or falling.

This information is used by the floor tracking module, which notifies the dialogue manager (DM) about changes in floor status. These status changes are classified by simple rules: silence following rising pitch leads to a timeout signal

[2] Although we chose to label this "intra-utterance" here, it doesn't matter much for our approach whether one considers this example to consist of one or several utterances; what matters is that differences in intonation and pragmatic completeness have an effect.

```
{< a ( 1 action=A=take; 2 prepare(A)   ; 3 U),
     ( 4 tile=T       ; 5 highlight(T) ; 6 U),
     ( 7              ; 8 execute(A,T) ; 9 U) >

 < b (10 action=A=del ;11 prepare(A)   ;12 U),
     (13 tile=T       ;14 highlight(T) ;15 U),
     (16              ;17 execute(A,T) ;18 U) >}
```

Figure 1: Example iQUD

sent to the DM faster (200ms) than silence after falling pitch (500ms). (Comparable to the rules in (Skantze and Schlangen, 2009).)

Natural language understanding finally is performed by a unification-based semantic composer, which builds simple semantic representations out of the lexical entries for the recognised words; and a resolver, which matches these representations against knowledge of the objects in the domain.

4.3 Dialogue Manager and Action Manager

The DM reacts to input from three sides: semantic material coming from the NLU, floor state signals from the floor tracker, and notifications about execution of actions from the action manager.

The central element of the information state used in the dialogue manager is what we call the iQUD (for *incremental* Question under Discussion, as it's a variant of the QUD of (Ginzburg, 1996)). Figure 1 gives an example. The iQUD collects all relevant sub-questions into one structure, which also records what the relevant non-linguistic actions are (RNLAs; more on this in a second, but see also (Buß and Schlangen, 2010), where we've sketched this approach before), and what the grounding status is of that sub-question.

Let's go through example (2). The iQUD in Figure 1 represents the state after the system has asked "what shall I do now?". The system anticipates two alternative replies, a *take* request, or a *delete* request; this is what the specification of the slot value in 1 and 10 in the iQUD indicates. Now the user starts to speak and produces what is shown in line 1 in the example. The floor tracker reacts to the rising pitch and to the silence of appropriate length, and notifies the dialogue manager. In the meantime, the DM has received updates from the NLU module, has checked for each update whether it is *relevant* to a sub-question on the iQUD, and if so, whether it *resolves* it. In this situation, the material was relevant to both 4 and 13, but did not resolve it. This is a precondition for the *continuer-questioning* rule, which is triggered by the signal from the floor tracker. The system then back-channels as in the example, indicating acoustic understanding (Clark's level 2), but failure to operate on the understanding (level 3). (As an aside, we found that it is far from trivial to find the right wording for this prompt. We settled on an "erm" with level pitch.)

The user then indeed produces more material, which together with the previously given information resolves the question. This is where the RNLAs come in: when a sub-question is resolved, the DM looks into the field for RNLAs, and if there are any, puts them up for execution to the action manager. In our case, slots 4 and 13 are both applicable, but as they have compatible RNLAs, this does not cause a conflict. When the action has been performed, a new question is accommodated (not shown here), which can be paraphrased as "was the understanding displayed through this action correct?". This is what allows the user reply in line 3 to be integrated, which otherwise would need to be ignored, or even worse, would confuse a dialogue system. A relevant continuation, on the other hand, would also have resolved the question. We consider this modelling of grounding effects of *actions* an important feature of our approach.

Similar rules handle other floor tracker events; not elaborated here for reasons of space. In our current prototype the rules are hard-coded, but we are preparing a version where rules and information-states can be specified externally and are read in by a rule-engine.

4.4 Overhearer Evaluation

Evaluating the contribution of one of the many modules in an SDS is notoriously difficult (Walker et al., 1998). To be able to focus on evaluation of the incremental dialogue strategies and avoid interference from ASR problems (and more technical problems; our system is still somewhat fragile), we opted for an overhearer evaluation. (Such a setting was also used for the test of the incremental system of (Aist et al., 2007).)

We implemented a non-incremental version of the system that does not give non-linguistic feedback during user utterances and has only one, fixed, timeout of 800ms (comparable to typical settings in commercial dialogue systems). Two of the authors then recorded 30 minutes of interactions with the two versions of the system.We then identified and discarded "outlier" interactions, i.e. those with technical problems, or where

recognition problems were so severe that a non-understanding state was entered repeatedly. These criteria were meant to be fair to both versions of the system, and indeed we excluded similar numbers of failed interactions from both versions (around 10 % of interactions in total).

We measured the length of interactions in the two sets, and found that the interactions in the incremental setting were significantly shorter (t-test, $p < 0.005$). This was to be expected, of course, as the incremental strategies allow faster reactions (execution time can be folded into the user utterance); other outcomes would have been possible, though, if the incremental version had systematically more understanding problems.

We then had 8 subjects (university students, not involved in the research) watch and directly judge (questionnaire, Likert-scale replies to questions about human-likeness, helpfulness, and reactivity) 34 randomly selected interactions from either condition. Human-likeness and reactivity were judged significantly higher for the incremental version (Wilcoxon rank-sum test; $p < 0.05$ and $p < 0.005$, respectively), while there was no effect for helpfulness ($p = 0.06$).

5 Conclusions

We described our incremental micro-domain dialogue system, which is capable of reacting to subtle signals from the user about expected feedback, and is able to produce overlapping non-linguistic actions, modelling their effect as displays of understanding. Interactions with the system were judged by overhearers to be more human-like and reactive than with a non-incremental variant. We are currently working on extending and generalising our approach to incremental dialogue management, porting it to other domains.

Acknowledgments Funded by an ENP grant from DFG.

References

Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, Mary Swift, and Michael K. Tanenhaus. 2007. Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proceedings of Decalog (Semdial 2007)*, Trento, Italy.

Timo Baumann, Michaela Atterer, and David Schlangen. 2009. Assessing and Improving the Performance of Speech Recognition for Incremental Systems. In *Proceedings of NAACL-HLT 2009*, Boulder, USA.

Okko Buß and David Schlangen. 2010. Modelling sub-utterance phenomena in spoken dialogue systems. In *Proceedings of Semdial 2010 ("Pozdial")*, pages 33–41, Poznan, Poland, June.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge.

Alain de Cheveigné and Hideki Kawahara. 2002. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930.

David DeVault, Natalia Kariaeva, Anubha Kothari, Iris Oved, and Matthew Stone. 2005. An information-state approach to collaborative reference. In *Short Papers, ACL 2005*, Michigan, USA, June.

Jonathan Ginzburg. 1996. Interrogatives: Questions, facts and dialogue. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*. Blackwell, Oxford.

Peter A. Heeman and Graeme Hirst. 1995. Collaborating on referring expressions. *Computational Linguistics*, 21(3):351–382.

Massimo Poesio and Hannes Rieser. 2010. Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1(1):1–89.

Matthew Purver, Christine Howes, Eleni Gregoromichelaki, and Patrick Healey. 2009. Split utterances in dialogue: a corpus study. In *Proceedings of the SIGDIAL 2009*, pages 262–271, London, UK, September.

Harvey Sacks and Emanuel A. Schegloff. 1979. Two preferences in the organization of reference to persons in conversation and their interaction. In George Psathas, editor, *Everyday Language: Studies in Ethnomethodology*, pages 15–21. Irvington Publishers, Inc., New York, NY, USA.

David Schlangen and Gabriel Skantze. 2009. A general, abstract model of incremental dialogue processing. In *Proceedings of EACL 2009*, pages 710–718, Athens, Greece, March.

David Schlangen, Timo Baumann, Hendrik Buschmeier, Okko Buß, Stefan Kopp, Gabriel Skantze, and Ramin Yaghoubzadeh. 2010. Middleware for incremental processing in conversational agents. In *Proceedings of SIGDIAL 2010*, Tokyo, Japan.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of EACL 2009*, pages 745–753, Athens, Greece, March.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1998. Evaluating spoken dialogue agents with PARADISE: Two case studies. *Computer Speech and Language*, 12(3).

Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, Evandro Gouvea, Peter Wolf, and Joe Woelfel. 2004. Sphinx-4: A flexible open source framework for speech recognition. Technical report, Sun Microsystems Inc.

Cross-Domain Speech Disfluency Detection

Kallirroi Georgila, Ning Wang, Jonathan Gratch
Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Playa Vista, CA 90094, USA
{kgeorgila,nwang,gratch}@ict.usc.edu

Abstract

We build a model for speech disfluency detection based on conditional random fields (CRFs) using the Switchboard corpus. This model is then applied to a new domain without any adaptation. We show that a technique for detecting speech disfluencies based on Integer Linear Programming (ILP) (Georgila, 2009) significantly outperforms CRFs. In particular, in terms of F-score and NIST Error Rate the absolute improvement of ILP over CRFs exceeds 20% and 25% respectively. We conclude that ILP is an approach with great potential for speech disfluency detection when there is a lack or shortage of in-domain data for training.

1 Introduction

Speech disfluencies (also known as speech repairs) occur frequently in spontaneous speech and can pose difficulties to natural language processing (NLP) since most NLP tools (e.g. parsers and part-of-speech taggers) are traditionally trained on written language. However, speech disfluencies are *not* noise. They are an integral part of how humans speak, may provide valuable information about the speaker's cognitive state, and can be critical for successful turn-taking (Shriberg, 2005). Speech disfluencies have been the subject of much research in the field of spoken language processing, e.g. (Ginzburg et al., 2007).

Speech disfluencies can be divided into three intervals, the *reparandum*, the *editing term*, and the *correction* (Heeman and Allen, 1999; Liu et al., 2006). In the example below, "it left" is the reparandum (the part that will be repaired), "I mean" is the editing term, and "it came" is the correction:

```
(it left) * (I mean) it came
```

The asterisk marks the interruption point at which the speaker halts the original utterance in order to start the repair. The editing term is optional and consists of one or more filled pauses (e.g. uh, um) or discourse markers (e.g. you know, well). Our goal here is to automatically detect repetitions (the speaker repeats some part of the utterance), revisions (the speaker modifies the original utterance), or restarts (the speaker abandons an utterance and starts over). We also deal with complex disfluencies, i.e. a series of disfluencies in succession ("it it was it is sounds great").

In previous work many different approaches to detecting speech disfluencies have been proposed. Different types of features have been used, e.g. lexical features only, acoustic and prosodic features only, or a combination of both (Liu et al., 2006). Furthermore, a number of studies have been conducted on human transcriptions while other efforts have focused on detecting disfluencies from the speech recognition output.

In our previous work (Georgila, 2009), we proposed a novel two-stage technique for speech disfluency detection based on Integer Linear Programming (ILP). ILP has been applied successfully to several NLP problems, e.g. (Clarke and Lapata, 2008). In the first stage of our method, we trained state-of-the-art classifiers for speech disfluency detection, in particular, Hidden-Event Language Models (HELMs) (Stolcke and Shriberg, 1996), Maximum Entropy (ME) models (Ratnaparkhi, 1998), and Conditional Random Fields (CRFs) (Lafferty et al., 2001). Then in the second stage and during testing, each classifier proposed possible labels which were then assessed in the presence of local and global constraints using ILP. These constraints are hand-crafted and encode common disfluency patterns. ILP makes the

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 237–240,
The University of Tokyo, September 24-25, 2010.

final decision taking into account both the output of the classifier and the constraints. Our approach is similar to the work of (Germesin et al., 2008) in the sense that they also combine machine learning with hand-crafted rules. However, we use different machine learning techniques and ILP.

When we evaluated this approach on the Switchboard corpus (available from LDC and manually annotated with disfluencies) using lexical features, we found that ILP significantly improves the performance of HELMs and ME models with negligible cost in processing time. However, the improvement of ILP over CRFs was only marginal. These results were achieved when each classifier was trained on approx. 35,000 occurrences of disfluencies. Then we experimented with varying training set sizes in Switchboard. As soon as we started reducing the amount of data for training the classifiers, the improvement of ILP over CRFs rose and became very significant, approx. 4% absolute reduction of error rate with 25% of the training set (approx. 9,000 occurrences of disfluencies) (Georgila, 2009). This result showed that ILP is particularly helpful when there is no much training data available.

However, Switchboard is a unique corpus because the amount of disfluencies that it contains is very large. Thus even 25% of our training set contains more disfluencies than a typical corpus of human-human or human-machine interactions. In this paper, we investigate what happens when we move to a new domain when there is no in-domain data annotated with disfluencies to be used for training. This is usually the case when we start developing a dialogue system in a new domain, when the system has not been fully implemented yet, and thus no data from users interacting with the system has been collected. Since the improvement of ILP over HELMs and ME models was very large even when the models were both trained and tested on Switchboard (approx. 15% and 20% absolute reduction of error rate when 100% and 25% of the training set was used for training the classifiers respectively (Georgila, 2009)), in this paper we focus only on comparing CRFs versus CRFs+ILP. Our goal is to evaluate if and how much ILP improves CRFs in the case that no training data is available at all.

The structure of the paper is as follows: In section 2 we describe our data sets. In section 3 we concisely describe our approach. Then in section 4 we present our experiments. Finally in section 5 we present our conclusion.

2 Data Sets

To train our classifiers we use Switchboard (available from LDC), which is manually annotated with disfluencies, and is traditionally used for speech disfluency experiments. We transformed the Switchboard annotations into the following format:

```
it BE was IE a IP it was good
```

BE (beginning of edit) is the point where the reparandum starts and IP is the interruption point (the point before the repair starts). In the above example the beginning of the reparandum is the first occurrence of "it", the interruption point appears after "a", and every word between BE and IP is tagged as IE (inside edit). Sometimes BE and IP occur at the same point, e.g. "it BE-IP it was". In (Georgila, 2009) we divided Switchboard into training, development, and test sets. Here we use the same training and development sets as in (Georgila, 2009) containing 34,387 occurrences of BE labels and 39,031 occurrences of IP labels, and 3,146 occurrences of BE labels and 3,499 occurrences of IP labels, respectively.

We test our approach on a smaller corpus collected in the framework of the Rapport project (Gratch et al., 2007). The goal of the Rapport project is to study how rapport is achieved in human-human and human-machine interaction. By rapport we mean the harmony, fluidity, synchrony and flow that someone feels when they are engaged in a good conversation.

The Rapport agent is a virtual human designed to elicit rapport from human participants within the confines of a dyadic narrative task (Gratch et al., 2007). In this setting, a speaker narrates some previously observed series of events, i.e. the events in a sexual harassment awareness and prevention video, and the events in a video of the Tweety cartoon. The central challenge for the Rapport agent is to provide the non-verbal listening feedback associated with rapportful interaction (e.g. head nods, postural mirroring, gaze shifts, etc.). Our ultimate goal is to investigate possible correlations between disfluencies and these types of feedback.

We manually annotated 70 sessions of the Rapport corpus with disfluencies using the labels described above (BE, IP, IE and BE-IP). In each session the speaker narrates the events of one video. These annotated sessions served as our reference data set (gold-standard), which contained 738 and 865 occurrences of BE and IP labels respectively.

3 Methodology

In the first stage we train our classifier. Any classifier can be used as long as it provides more than one possible answer (i.e. tag) for each word in the utterance. Valid tags are BE, BE-IP, IP, IE or O. The O tag indicates that the word is outside the disfluent part of the utterance. ILP will be applied to the output of the classifier during testing.

Let N be the number of words of each utterance and i the location of the word in the utterance (i=1,...,N). Also, let $C_{BE}(i)$ be a binary variable (1 or 0) for the BE tag. Its value will be determined by ILP. If it is 1 then the word will be tagged as BE. In the same way, we use $C_{BE-IP}(i)$, $C_{IP}(i)$, $C_{IE}(i)$, $C_O(i)$ for tags BE-IP, IP, IE and O respectively. Let $P_{BE}(i)$ be the probability given by the classifier that the word is tagged as BE. In the same way, let $P_{BE-IP}(i)$, $P_{IP}(i)$, $P_{IE}(i)$, $P_O(i)$ be the probabilities for tags BE-IP, IP, IE and O respectively. Given the above definitions, the ILP problem formulation can be as follows:

$$max[\sum_{i=1}^{N}[P_{BE}(i)C_{BE}(i) + P_{BE-IP}(i)C_{BE-IP}(i) + P_{IP}(i)C_{IP}(i) + P_{IE}(i)C_{IE}(i) + P_O(i)C_O(i)]] \quad (1)$$

subject to constraints, e.g.:

$$C_{BE}(i) + C_{BE-IP}(i) + C_{IP}(i) + C_{IE}(i) + C_O(i) = 1 \qquad \forall i \in (1, ..., N) \quad (2)$$

Equation 1 is the linear objective function that we want to maximize, i.e. the overall probability of the utterance. Equation 2 says that each word can have one tag only. In the same way, we can define constraints on which labels are allowed at the start and end of an utterance. There are also some constraints that define the transitions that are allowed between tags. For example, IP cannot follow an O directly, which means that we cannot start a disfluency with an IP. There has to be a BE after O and before IP. Details are given in (Georgila, 2009).

We also formulate some additional rules that encode common disfluency patterns. The idea here is to generalize from these patterns. Below is an example of a long-context rule. If we have the sequence of words "she was trying to well um she was talking to a coworker", we expect this to be tagged as "she BE was IE trying IE to IP well O um O she O was O talking O to O a O coworker O", if we do not take into account the context in which this pattern occurs. Basically the pattern here is that two sequences of four words separated by a discourse marker ("well") and a filled pause ("um") differ only in their third word. That is, "trying" and "talking" are different words but have the same part-of-speech tag (gerund). We incorporate this rule into our ILP problem formulation as follows: Let (w_1,...,w_N) be a sequence of N words where both w_3 and w_{N-3} are verbs (gerund), the word sequence w_1,w_2,w_4 is the same as the sequence w_{N-5},w_{N-4},w_{N-2}, and all the words in between (w_5,...,w_{N-6}) are filled pauses or discourse markers. Then the probabilities given by the classifier are modified as follows: $P_{BE}(1)$=$P_{BE}(1)$+$b1$, $P_{IE}(2)$=$P_{IE}(2)$+$b2$, $P_{IE}(3)$=$P_{IE}(3)$+$b3$, and $P_{IP}(4)$=$P_{IP}(4)$+$b4$, where $b1$, $b2$, $b3$ and $b4$ are empirically set boosting paremeters with values between 0.5 and 1 computed using our Switchboard development set. We use more complex rules to cover cases such as "she makes he doesn't make", and boost the probabilities that this is tagged as "she BE makes IP he O doesn't O make O".

In total we apply 17 rules and each rule can have up to 5 more specific sub-rules. The largest context that we take into account is 10 words, not including filled pauses and discourse markers.

4 Experiments

For building the CRF model we use the CRF++ toolkit (available from `sourceforge`). We used only lexical features, i.e. words and part-of-speech (POS) tags. Switchboard includes POS information but to annotate the Rapport corpus with POS labels we used the Stanford POS tagger (Toutanova and Manning, 2000). We experimented with different sets of features and we achieved the best results with the following setup (i is the location of the word or POS in the sentence): Our word features are $\langle w_i \rangle$, $\langle w_{i+1} \rangle$, $\langle w_{i-1}, w_i \rangle$, $\langle w_i, w_{i+1} \rangle$, $\langle w_{i-2}, w_{i-1}, w_i \rangle$, $\langle w_i, w_{i+1}, w_{i+2} \rangle$. Our POS features have the same structure as the word features. For ILP we use the `lp_solve` software also available from `sourceforge`. We train on Switchboard and test on the Rapport corpus.

For evaluating the performance of our models we use standard metrics proposed in the literature, i.e. Precision, Recall, F-score, and NIST Error Rate. We report results for BE and IP. F-score is the harmonic mean of Precision and Recall (we equally weight Precision and Recall). Precision is the ratio of the correctly identified tags X to all the tags X detected by the model (where X is BE or IP). Recall is the ratio of the correctly identified tags X to all the tags X that appear in the reference

	BE			
	Prec	**Rec**	**F-score**	**Error**
CRF	74.52	36.45	48.95	73.44
CRF+ILP	77.44	64.63	70.46	47.56
	IP			
	Prec	**Rec**	**F-score**	**Error**
CRF	86.36	41.73	56.27	64.62
CRF+ILP	88.75	72.95	80.08	35.61

Table 1: Comparative results between our models.

utterance. The NIST Error Rate is the sum of insertions, deletions and substitutions divided by the total number of reference tags (Liu et al., 2006).

Table 1 presents comparative results between our models. As we can see, now the improvement of ILP over CRFs is not marginal as in Switchboard. In fact, in terms of F-score and NIST Error Rate the absolute improvement of ILP over CRFs exceeds 20% and 25% respectively. The results are statistically significant ($p<10^{-8}$, Wilcoxon signed-rank test). The main gain of ILP comes from the large improvement in Recall. This result shows that using ILP has great potential for speech disfluency detection when there is a lack of in-domain data for training, and when we use lexical features and human transcriptions. Furthermore, the cost of applying ILP is negligible since the process is fast and applied during testing.

Note that the improvement of ILP over CRFs is significant even though the two corpora, Switchboard and Rapport, differ in genre (conversation versus narrative).

The reason for the large improvement of ILP over CRFs is the fact that as explained above ILP takes into account common disfluency patterns and generalizes from them. CRFs can potentially learn similar patterns from the data but do not generalize that well. For example, if the CRF model learns that "she she" is a repetition it will not necessarily infer that any sequence of the same two words is a repetition (e.g. "and and"). Of course here, since we deal with human transcriptions we do not worry about speech recognition errors. Preliminary results with speech recognition output showed that ILP retains its advantages but more modestly. In this case, when deciding which boosting rules to apply, it makes sense to consider speech recognition confidence scores per word. For example, a possible repetition "to to" could be the result of a misrecognition of "to do". But these types of problems also affect plain CRFs, so in the end ILP is expected to continue outperforming CRFs. This is one of the issues for future work together with using prosodic features.

5 Conclusion

We built a model for speech disfluency detection based on CRFs using the Switchboard corpus. This model was then applied to a new domain without any adaptation. We showed that a technique for detecting speech disfluencies based on ILP significantly outperforms CRFs. In particular, in terms of F-score and NIST Error Rate the absolute improvement of ILP over CRFs exceeds 20% and 25% respectively. We conclude that ILP is an approach with great potential for speech disfluency detection when there is a lack or shortage of in-domain data for training.

Acknowledgments

This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

J. Clarke and M. Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

K. Georgila. 2009. Using integer linear programming for detecting speech disfluencies. In *Proc. of NAACL*.

S. Germesin, T. Becker, and P. Poller. 2008. Hybrid multistep disfluency detection. In *Proc. of MLMI*.

J. Ginzburg, R. Fernández, and D. Schlangen. 2007. Unifying self- and other-repair. In *Proc. of DECALOG*.

J. Gratch, N. Wang, J. Gerten, E. Fast, and R. Duffy. 2007. Creating rapport with virtual agents. In *Proc. of International Conference on Intelligent Virtual Agents (IVA)*.

P. Heeman and J. Allen. 1999. Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialogue. *Computational Linguistics*, 25:527–571.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML*.

Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper. 2006. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Trans. Audio, Speech and Language Processing*, 14(5):1526–1540.

A. Ratnaparkhi. 1998. *Maximum Entropy Models for natural language ambiguity resolution*. Ph.D. thesis, University of Pennsylvania.

E. Shriberg. 2005. Spontaneous speech: How people really talk, and why engineers should care. In *Proc. of Interspeech*.

A. Stolcke and E. Shriberg. 1996. Statistical language modeling for speech disfluencies. In *Proc. of ICASSP*.

K. Toutanova and C.D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proc. of EMNLP/VLC*.

Validation of a Dialog System for Language Learners

Alicia Sagae, W. Lewis Johnson, Stephen Bodnar
Alelo, Inc.
Los Angeles, CA
{asagae,ljohnson,sbodnar}@alelo.com

Abstract

In this paper we present experiments related to the validation of spoken language understanding capabilities in a language and culture training system. In this application, word-level recognition rates are insufficient to characterize how well the system serves its users. We present the results of an annotation exercise that distinguishes instances of non-recognition due to learner error from instances due to poor system coverage. These statistics give a more accurate and interesting description of system performance, showing how the system could be improved without sacrificing the instructional value of rejecting learner utterances when they are poorly formed.

1 Introduction

Conversational practice in real-time dialogs with virtual humans is a compelling element of training systems for communicative competency, helping learners acquire procedural skills in addition to declarative knowledge (Johnson, Rickel et al. 2000). Alelo's language and culture training systems allow language learners to engage in such dialogs in a serious game environment, where they practice task-based missions in new linguistic and cultural settings (Barrett and Johnson 2010). To support this capability, Alelo products apply a variety of spoken dialog technologies, including automatic speech recognition (ASR) and agent-based models of dialog that capture theories of politeness (Wang and Johnson 2008), and cultural expectations (Johnson, 2010; (Sagae, Wetzel et al. 2009).

To properly assess these dialog systems, we must take several issues into account. First, users who interact with these systems are language learners, who can be expected occasionally to produce invalid speech, and who may benefit from the corrective signal of recognizer rejection. Second, word recognition is one step in a social simulation pipeline that allows virtual humans to respond to learner input (Samtani, Valente et al. 2008). Consequently, the system goals extend beyond word-level decoding into meaning interpretation and response planning.

As a result, Word Error Rate (WER) and related metrics, such as those described by Hunt (1990) for evaluating ASR performance, are insufficient to characterize how well the speech understanding component of the dialog system performs. We need a meaningful way to account for the performance of the dialog system as a whole, which can distinguish acceptable interpretation failures from unacceptable ones.

We present a validation process for assessing speech understanding in dialog systems for language training applications. The process involves annotation of historical user data acquired from learner interaction with the Tactical Language and Culture Training System (Johnson and Valente 2009). The results indicate that learner mistakes make up the majority of non-recognitions, confirming the hypothesis that "recognition failures" are a complex category of events that are only partly explained by lack of coverage in speech understanding components such as ASR.

2 Metrics for Dialog System Assessment

Speech recognition errors in the dialog system result in at least two sub-types of error: *non-understandings*, where the system cannot find an interpretation for user input, and *misunderstandings*, where the system finds an interpretation that does not match the learner's intent (McRoy and Hirst 1995).

These classes generalize beyond speech recognition to speech understanding. This is shown in Figure 1, where "act" refers to a message

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 241–244,
The University of Tokyo, September 24-25, 2010.

modeled along the lines of Traum & Hinkleman (1992). In the context of speech-enabled dialog systems, the understanding task is more critical, since it more closely models the overall success of the communication between the human user and the virtual human interlocutor.

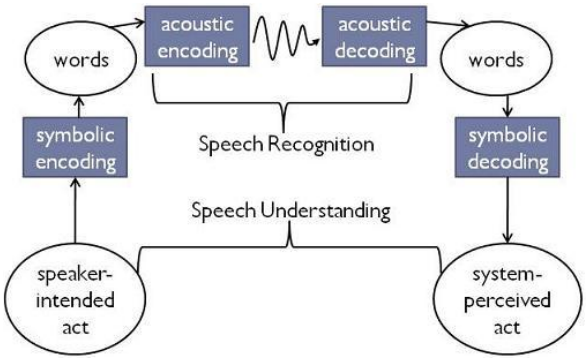


Figure 1. Speech understanding pipeline.

As a result, a variety of metrics have been suggested that assess performance at the level of intent recognition, rather than word recognition. Examples include PARADISE (Walker, Litman et al. 1998) and the work of Suendermann, Liscombe, et al (2009).

We propose an assessment procedure that uses expert annotation to compare speaker-intended acts to the acts recognized by the speech-understanding component of the dialog system. Like the metrics mentioned above, it evaluates the system's ability to recognize intent as well as words. However we focus our attention on adaptations that characterize interactions with *language learners*, who are a special type of user. As a result, we can distinguish system non-understandings and mis-understandings that are due to system error from those that are caused by learner mistakes.

Our goal is to use this information to reduce mis-understandings due to system errors; such mis-understandings can yield confusing dialog behavior, causing learners to lose confidence in the accuracy of the speech recognizer. Non-understandings may be less serious, since they occur in real life between learners and native speakers. Non-understandings due to learner error may be beneficial if the additional practice that results from non-understandings leads to an increase in language accuracy.

3 Procedure

To assess performance, we recruited two annotators to provide judgments on historical log data regarding the accuracy of the system interpretations at multiple levels, including word-level recognition and act recognition.

3.1 Annotation team and data collection

The annotators are Alelo team members with expertise in General Linguistics, French and Spanish Linguistics, Translation, and Teaching English as a Foreign Language (TEFL). Their combined experience in content authoring for Alelo courses covers more than 10 languages.

The data was collected in the fall of 2009 as part of a field test for Alelo courses teaching Iraqi Arabic and Sub-Saharan French. Naval personnel at several sites around the United States volunteered to complete the courses in self-study. The training systems generated user logs, capturing recordings of learner turns and system recognition results for each turn. From these logs, samples of beginner-level and intermediate-level dialogs were selected and anonymized for annotation.

3.2 Speech understanding accuracy

The point of this exercise is to explore how often the system fails to understand what a learner is trying to say during spoken dialog.

Annotation was performed on a total of 345 learner turns. To determine the act-level accuracy of the speech understanding system, annotators listened to the recording of each turn and selected the act they heard from a drop-down list. The results were compared with the system-perceived act result recovered from the log. Speech understanding rejections, where the system determined that no meaningful act could be perceived from the learner turn, were labeled with the act name "garbage". Human annotators could also select the garbage act for recordings where no meaningful interpretation could be made.

4 Results

To analyze the results, we measure system accuracy at two levels. First, we determine accuracy on distinguishing meaningful utterances (utterances that annotators labeled with an act) from non-meaningful speech attempts (labeled as garbage by annotators). The results are shown in Table 1. Inter-annotator agreement as measured by Cohen's Kappa on the first task is 0.8, indicating good agreement between our two experts.

Next, we examine the utterances classified as meaningful by both the system and the annota-

tors, to assess correctness at a finer level of granularity: given that the system identified the utterance as meaningful, did the meaning that it assigned match our annotators' judgments? If not, mis-understandings occur. These results are shown in Table 2. System mis-understandings over all meaningful utterances. Inter-annotator agreement on the non-understanding classification task was 0.73, suggesting that there is substantial agreement between our raters.

4.1 Correct interpretations

Numbers in the bottom-right cells of Table 1 and the first row of Table 2 represent correct system interpretations, according to an annotator. In these instances, the annotator assigned an act to the turn that matched the system interpretation for that turn (in Table 2), or both the annotator and the system assigned the label "garbage" (in Table 1). On average these examples account for 62% of the total turns.

An important result from this procedure is that it reveals the class of appropriate rejections by the speech understanding component. These "garbage-in, garbage-out" instances are instructive cases where the system indicates to the learner that he or she should re-try the utterance.

4.2 Mis-understandings

In Table 2, the row labeled "Incorrect" contains mis-understandings, where the system made an interpretation but failed to match the expert annotation. Mis-understandings account for around 3.5% of the turns in our data set, on average. The low rate of mis-understandings is an encouraging result for the overall quality of the understanding component. Prior to the introduction of the garbage model into the speech recognizer the mis-understanding rate had been relatively high, and these results indicate a significant improvement.

		Annotator 1	
		Act	Garbage
System	Act	175	3
	Garbage	94	73

		Annotator 2	
		Act	Garbage
System	Act	176	2
	Garbage	134	33

Table 1. Distinguishing meaningful utterances (corresponding to an Act) from non-meaningful attempts (Garbage).

System	**Annotator 1**	**Annotator 2**
Correct	167	160
Incorrect	8	16

Table 2. System mis-understandings over all meaningful utterances.

4.3 Non-understandings

Instances from the data set where the annotator was able to interpret an act, but the system returned "garbage," are shown in the lower-left cells of Table 1. These are system non-understandings, since the speech understanding component was not able to map the learner input to a meaningful act, even though the annotators were. Non-understandings account for 33% of turns in our data set, on average.

To understand the impact of these non-understandings on dialog system quality, we must consider the specialized case of language learners. Several components of the speech understanding pipeline are tuned with language learners in mind. For example, acoustic models used in the automatic speech recognizer are trained on a mixture of native and non-native data. The goal is for the system to be as tolerant as possible of pronunciation variability, while still catching learner mistakes.

We expect learner speech attempts to occur on a continuum, ranging from fully correct to minor mistakes to unrecoverable errors. In the first procedure, the annotators were instructed to label a recording with a meaningful act in all cases where they could do so, using garbage only for unintelligible attempts. As a result, we consciously placed the annotator tolerance at the far end of this spectrum.

Since the system is less forgiving, we hypothesize that the non-understandings we found mask two different sub-classes: instances where the system truly failed to interpret a well-formed utterance, and instances where the system was (perhaps appropriately) rejecting a learner mistake: an intelligible but malformed utterance.

In a follow-up procedure, the annotators revisited instances labeled as non-understandings. In this second round, they distinguished instances where the learner successfully performed an act that was simply outside the coverage of the speech understanding system from instances where they perceived a learner error, either in pronunciation or grammar. The results are summarized in Table 3.

We found that most of the cases of non-recognition were actually due to learner error, rather than system error.

Annotator 1	
Error Type	Count
Learner Grammar	0
Learner Pronunciation	58 (62%)
System Error	36
Total	94

Annotator 2		
Error Type	Count	κ
Learner Grammar	2	0
Learner Pronunciation	85 (63%)	0.65
System Error	47	0.65
Total	134	0.73

Table 3. Classification of non-understandings. Inter-annotator agreement (κ) is substantial over all classes.

5 Conclusions and Future Work

By applying a method for assessment that goes beyond word recognition rate, we have produced an analysis of the speech understanding components in a dialog system for language learners. Expert annotators found that most system-understood speech attempts were interpreted correctly, with mis-understandings occurring only 3% of the time. While non-understandings occurred much more frequently, a follow-up exercise showed that learner pronunciation error was the most frequent cause; these cases are legitimate candidates for system rejection, leaving 12% of all instances as non-understandings where the system was at fault. These instances represent the most beneficial errors to correct when making refinements to the speech understanding module.

In this exercise, one could interpret the human-assigned acts as a model of recognition by an extremely sympathetic hearer. Although this model may be too lenient to provide learners with realistic communication practice, it could be useful for the dialog engine to recognize some poorly-formed utterances, for the purpose of providing feedback. For example, a learner who repeatedly attempts the same utterance with unacceptable but intelligible pronunciation could trigger a tutoring-style intervention ("Are you trying to say *bonjour*? Try it more like this...").

The assessment methods and analysis presented in this paper are a first step toward this type of system improvement, one that meets the needs of language learners as a unique type of dialog-system user.

Acknowledgments

The authors thank Rebecca Row and Mickey Rosenberg for their contributions to the experiments described here, and three anonymous reviewers for comments that improved the clarity of the paper. This work was sponsored by PM TRASYS, Voice of America, the Office of Naval Research, and DARPA. Opinions expressed here are those of the author and not of the sponsors or the US Government.

References

Barrett, K. A. and W. L. Johnson (2010). Developing serious games for learning language-in-culture. Interdisciplinary Models and Tools for Serious Games: Emerging Concepts and Future Directions. R. V. Eck. Hershey, PA, IGI Global.

Hunt, M. J. (1990). "Figures of Merit for Assessing Connected Word Recognisers." Speech Communication **9**: 239-336.

Johnson, W. L., J. Rickel, et al. (2000). "Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments." Journal of Artificial Intelligence in Education **11**: 47--78.

Johnson, W. L. and A. Valente (2009). "Tactical Language and Culture Training Systems: Using AI to Teach Foreign Languages and Cultures." AI Magazine **30**(2).

McRoy, S. W. and G. Hirst (1995). "The repair of speech act misunderstandings by abductive inference." Computational Linguistics **21**(4): 435--478.

Sagae, A., B. Wetzel, et al. (2009). Culture-Driven Response Strategies for Virtual Human Behavior in Training Systems. SLaTE-2009, Warwickshire, England.

Samtani, P., A. Valente, et al. (2008). Applying the SAIBA framework to the Tactical Language and Culture Training System. AAMAS 2008 Workshop on Functional Markup Language (FML).

Suendermann, D., J. Liscombe, et al. (2009). A handsome set of metrics to measure utterance classification performance in spoken dialog systems. SigDial 2009.

Walker, M. A., D. J. Litman, et al. (1998). "Evaluating spoken dialogue agents with PARADISE: Two case studies." Computer Speech & Language **12**(4): 317-347.

Wang, N. and W. L. Johnson (2008). The Politeness Effect in an Intelligent Foreign Language Tutoring System. ITS 2008.

I've said it before, and I'll say it again: An empirical investigation of the upper bound of the selection approach to dialogue

Sudeep Gandhe and **David Traum**
Institute for Creative Technologies
13274 Fiji way, Marina del Rey, CA 90292
{gandhe,traum}@ict.usc.edu

Abstract

We perform a study of existing dialogue corpora to establish the theoretical maximum performance of the selection approach to simulating human dialogue behavior in unseen dialogues. This maximum is the proportion of test utterances for which an exact or approximate match exists in the corresponding training corpus. The results indicate that some domains seem quite suitable for a corpus-based selection approach, with over half of the test utterances having been seen before in the corpus, while other domains show much more novelty compared to previous dialogues.

1 Introduction

There are two main approaches toward automatically producing dialogue utterances. One is the *selection* approach, in which the task is to pick the appropriate output from a corpus of possible outputs. The other is the *generation* approach, in which the output is dynamically assembled using some composition procedure, e.g. grammar rules used to convert information from semantic representations and/or context to text.

The generation approach has the advantage of a more compact representation for a given generative capacity. But for any finite set of sentences produced, the selection approach could perfectly simulate the generation approach. The generation approach generally requires more analytical effort to devise a good set of grammar rules that cover the range of desired sentences but do not admit undesirable or unnatural sentences. Whereas, in the selection approach, outputs can be limited to those that have been observed in human speech. This affords complex and human-like sentences without much detailed analysis. Moreover, when the output is not just text but presented as speech, the system may easily use recorded audio clips rather than speech synthesis. This argument also extends to multi-modal performances, e.g. using artist animation motion capture or recorded video for animating virtual human dialogue characters. Often one is willing to sacrifice some generality in order to achieve more human-like behavior than is currently possible from generation approaches.

The selection approach has been used for a number of dialogue agents, including question-answering characters at ICT (Leuski et al., 2006; Artstein et al., 2009; Kenny et al., 2007), FAQ bots (Zukerman and Marom, 2006; Sellberg and Jönsson, 2008) and web-site information characters. It is also possible to use the selection approach as a part of the process, e.g. from words to a semantic representation or from a semantic representation to words, while using other approaches for other parts of dialogue processing.

The selection approach presents two challenges for finding an appropriate utterance:

- *Is there a good enough utterance to select?*
- *How good is the selection algorithm at finding this utterance?*

We have previously attempted to address the second question, by proposing the information ordering task for evaluating dialogue coherence (Gandhe and Traum, 2008). Here we try to address the first question, which would provide a theoretical upper bound in quality for any selection approach. We examine a number of different dialogue corpora as to their suitability for the selection approach.

We make the following assumptions to allow automatic evaluation across a range of corpora. Actual human dialogues represent a gold-standard for computer systems to emulate; i.e. choosing an actual utterance in the correct place is the best possible result. Other utterances can be evaluated as to how close they come to the original utterance,

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 245–248,
The University of Tokyo, September 24-25, 2010.

using a similarity metric.

Our methodology is to examine a test corpus of human dialogue utterances to see how well a selection approach could approximate these, given a training corpus of utterances in that domain. We look at exact matches as well as utterances having their similarity score above a threshold. We investigate the effect of the size of training corpora, which lets us know how much data we might need to achieve a certain level of performance. We also investigate the effect of domain of training corpora.

2 Dialogue Corpora

We examine human dialogue utterances from a variety of domains. Our initial set contains six dialogue corpora from ICT as well as three other publicly available corpora.

SGT Blackwell is a question-answering character who answers questions about the U.S. Army, himself, and his technology. The corpus consists of visitors interacting with SGT Blackwell at an exhibition booth at a museum. **SGT Star** is a question-answering character, like SGT Blackwell, who talks about careers in the U.S. Army. The corpus consists of trained handlers presenting the system. **Amani** is a bargaining character used as a prototype for training soldiers to perform tactical questioning. The **SASO** system is a negotiation training prototype in which two virtual characters negotiate with a human "trainee" about moving a medical clinic. The **Radiobots** system is a training prototype that responds to military calls for artillery fire. **IOTA** is an extension of the Radiobots system. The corpus consists of training sessions between a human trainee and a human instructor on a variety of missions. Yao et al. (2010) provides details about the ICT corpora.

Other corpora involved dialogues between two people playing specific roles in planning, scheduling problem for railroad transportation, the **Trains-93** corpus (Heeman and Allen, 1994) and for emergency services, the **Monroe** corpus (Stent, 2000). The **Switchboard** corpus (Godfrey et al., 1992) consists of telephone conversations between two people, based on provided topics.

We divided the data from each corpus into a training set and a test set, as shown in Table 1. The data consists of utterances from one or more human speakers who engage in dialogue with either virtual characters (Radiobots, Blackwell, Amani, Star, SASO) or other humans (Switchboard, Monroe, IOTA, Trains-93). These corpora differ along a number of dimensions such as the size of the corpus, dialogue genre (question-answering, task-oriented or conversational), types of tasks (artillery calls, moving and scheduling resources, information seeking) and motivation of the participants (exploring a new technology – SGT Blackwell , presenting a demo – SGT Star, undergoing training – Amani, IOTA or simply for collecting the corpus – Switchboard, Trains-93, Monroe). While the set of corpora we include does not cover all points in these dimensions, it does present an interesting range.

3 Dialogue Utterance Similarity Metrics

To answer the question of whether an adequate utterance exists in our training corpus that could be selected and used, we need an appropriateness measure. We assume that an utterance produced by a human in a dialogue is appropriate, and thus the problem becomes one of constructing an appropriate similarity function to compare the human-produced utterance with the utterances available from the training corpus. Given a training corpus U_{train} and a similarity function f , we calculate the score for a test utterance u_t as, $maxsim_f(u_t) = \max_i f(u_t, u_i); u_i \in U_{train}$ There are several choices for the utterance similarity function f . Ideally such a function would take meaning and context into account rather than just surface similarity, but these aspects are harder to automate, so for our initial experiments we look at several surface metrics, as described below.

Exact measure returns 1 if the utterances are exactly same and 0 otherwise. **1-WER**, a similarity measure related to word error rate, is defined as $min\,(0, 1 - levenshtein(u_t, u_i)/length(u_t))$. **METEOR** (Lavie and Denkowski, 2009), one of the automatic evaluation metrics used in machine translation is a good candidate for f. METEOR finds optimal word-to-word alignment between test and reference strings based on several modules that match exact words, stemmed words and synonyms. METEOR is a tunable metric and for our analysis we used the default parameters tuned for the Adequacy & Fluency task. All previous measures take into account the word ordering of test and reference strings. In contrast, document similarity measures used in information retrieval generally follow the *bag of words* assumption, where a

Domain	Train		Test		$mean(maxsim_f)$				% of utterances		
										METEOR	
	# utt	words	# utt	words	METEOR	1-WER	Dice	Cosine	Exact	≥ 0.9	≥ 0.8
Blackwell	17755	84.7k	2500	12.0k	0.913	0.878	0.917	0.921	69.6	75.8	82.1
Radiobots	995	6.8k	155	1.2k	0.905	0.864	0.920	0.924	53.6	67.7	83.2
SGT Star	2974	16.6k	400	2.2k	0.897	0.860	0.906	0.911	65.0	70.5	78.0
SASO	3602	23.3k	510	3.6k	0.821	0.742	0.830	0.837	38.4	48.6	62.6
IOTA	4935	50.4k	650	5.6k	0.768	0.697	0.800	0.808	36.2	42.8	51.4
Trains 93	5554	47.2k	745	6.0k	0.729	0.633	0.758	0.769	34.5	36.9	42.8
SWBD[1]	19741	138.2k	3173	21.5k	0.716	0.628	0.736	0.753	35.8	37.9	44.2
Amani	1455	15.8k	182	1.9k	0.675	0.562	0.694	0.706	18.7	25.8	30.8
Monroe	5765	43.0k	917	8.8k	0.594	0.491	0.639	0.658	22.3	23.6	26.1

Table 1: Corpus details and within domain results

string is converted to a set of tokens. Here we also considered **Cosine** and **Dice** coefficients using the standard boolean model. In our experiments, the surface text was normalized and all punctuation was removed.

4 Experiments

Results Within a Domain

In our first experiment, we computed $maxsim_f$ scores for all test corpus utterances in a given domain using the training utterances from the same domain. For the domains Blackwell, SGT Star, SASO, Amani & Radiobots which are implemented dialogue systems our corpus consists of user utterances only. For Trains 93 and Monroe corpora, we make sure to match the speaker roles for u_t and u_i. For Switchboard, where speakers do not have any special roles and for IOTA, where the speaker information was not readily accessible, we ignore the speaker information and select utterances from either speaker.

Table 1 reports the mean of $maxsim_f$ scores. These can be interpreted as the expectation of $maxsim_f$ score for a new test utterance. The higher this expectation, the more likely it is that an utterance similar to the new one has been seen before and thus the domain will be more amenable to selection approaches. This table also shows the percentage of utterances that had a $maxsim_{Meteor}$ score above a certain threshold. The correlation between $maxsim_f$ for different choices of f (except Exact match) is very high (Pearson's $r > 0.94$). The histogram analysis shows that SGT Star, Blackwell, Radiobots and SASO domains are better suited for *selection* approaches. Domains like Trains-93, Monroe, Switchboard and Amani have a more diffuse distribution and are not best suited for *selection* approaches, at least with the amount of data we have available. The IOTA domain falls somewhere in between these two domain classes.

[1]Switchboard (SWBD) is a very large corpus and for running our experiments in a reasonable computing time we only selected a small portion of it.

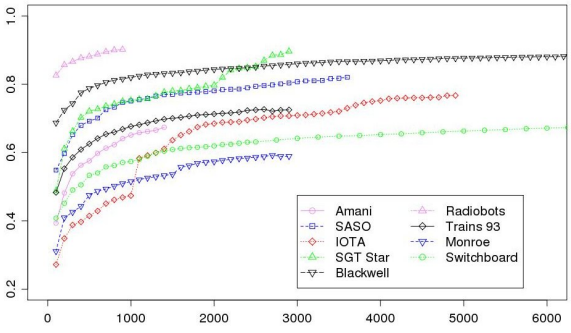


Figure 1: $maxsim_{Meteor}$ vs # utterances in training data for different domains

Effect of Training Data Size

Figure 1 shows the effect of training data size on the $maxsim_{Meteor}$ score. Radiobots shows very high scores even for small amounts of training data. SGT Star and SGT Blackwell also converge fairly early. Switchboard, on the other hand, does not achieve very high scores even with a large number of utterances. For all domains, with around 2500 training utterances $maxsim_{Meteor}$ reaches 90% of its maximum possible value for the training set.

Comparing Different Domains

In order to understand the similarities between different dialogue domains, we computed $maxsim_{Meteor}$ for a test domain using training

		Training Domains								
		IOTA	Radio-bots	SGT Star	Black-well	Amani	SASO	Trains-93	Monroe	SWBD
Testing Domains	IOTA	**0.768**	0.440	0.247	0.334	0.196	0.242	0.255	0.297	0.334
	Radiobots	0.842	**0.905**	0.216	0.259	0.161	0.183	0.222	0.270	0.284
	SGT Star	0.324	0.136	**0.897**	0.622	0.372	0.438	0.339	0.417	0.527
	Blackwell	0.443	0.124	0.671	**0.913**	0.507	0.614	0.424	0.534	0.696
	Amani	0.393	0.134	0.390	0.561	**0.675**	0.478	0.389	0.420	0.509
	SASO	0.390	0.125	0.341	0.516	0.459	**0.821**	0.443	0.454	0.541
	Trains 93	0.434	0.112	0.214	0.468	0.272	0.429	**0.753**	0.627	0.557
	Monroe	0.409	0.119	0.217	0.428	0.276	0.404	0.534	**0.630**	0.557
	SWBD	0.368	0.110	0.280	0.490	0.362	0.383	0.562	0.599	**0.716**

Table 2: Mean of $maxsim_{Meteor}$ for comparing different dialogue domains. The **bold-faced** values are the highest in the corresponding row.

sets from other domains. In this exercise, we ignored the speaker information. Table 2 reports the mean values of $maxsim_{Meteor}$ for different training domains. For all the testing domains, using the training corpus from the same domain produces the best results. Notice that Radiobots also has good performance with the IOTA training data. This is as expected since IOTA is an extension of Radiobots and should cover a lot of utterances from the Radiobots domain. Switchboard and Blackwell training corpora have a overall higher score for all testing domains. This may be due to the breadth and size of these corpora. On the other extreme, the Radiobots training domain performs very poorly on all testing domains other than itself.

5 Discussion

We have examined how well suited a corpus-based selection approach to dialogue can succeed at mimicking human dialogue performance across a range of domains. The results show that such an approach has the potential of doing quite well for some domains, but much less well for others. Results also show that for some domains, quite modest amounts of training data are needed for this operation. Applying this method across corpora from different domains can also give us a similarity metric for dialogue domains. Our hope is that this kind of analysis can help inform the decision of what kind of language processing methods and dialogue architectures are most appropriate for building a dialogue system for a new domain, particularly one in which the system is to act like a human.

Acknowledgments

This work has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. We would like to thank Ron Artstein and others at ICT for compiling the ICT Corpora used in this study.

References

R. Artstein, S. Gandhe, J. Gerten, A. Leuski, and D. Traum. 2009. Semi-formal evaluation of conversational characters. In *Languages: From Formal to Natural. Essays Dedicated to Nissim Francez on the Occasion of His 65th Birthday*, volume 5533 of *LNCS*. Springer.

S. Gandhe and D. Traum. 2008. Evaluation understudy for dialogue coherence models. In *Proc. of SIGdial 08.*

J. J. Godfrey, E. C. Holliman, and J. McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proc. of ICASSP-92*, pages 517–520.

P. A. Heeman and J. Allen. 1994. The TRAINS 93 dialogues. TRAINS Technical Note 94-2, Department of Computer Science, University of Rochester.

P. Kenny, T. Parsons, J. Gratch, A. Leuski, and A. Rizzo. 2007. Virtual patients for clinical therapist skills training. In *Proc. of IVA 07*, Paris, France. Springer.

A. Lavie and M. J. Denkowski. 2009. The meteor metric for automatic evaluation of machine translation. *Machine Translation*, 23:105–115.

A. Leuski, R. Patel, D. Traum, and B. Kennedy. 2006. Building effective question answering characters. In *Proc. of SIGdial 06*, pages 18–27, Sydney, Australia.

L. Sellberg and A. Jönsson. 2008. Using random indexing to improve singular value decomposition for latent semantic analysis. In *Proc. of LREC'08*, Morocco.

A. J. Stent. 2000. The monroe corpus. Technical Report 728, Computer Science Dept. University of Rochester.

X. Yao, P. Bhutada, K. Georgila, K. Sagae, R. Artstein, and D. Traum. 2010. Practical evaluation of speech recognizers for virtual human dialogue systems. In *LREC 2010.*

I. Zukerman and Y. Marom. 2006. A corpus-based approach to help-desk response generation. In *CIMCA/IAWTIC '06.*

Autism and Interactional Aspects of Dialogue

Peter A. Heeman, Rebecca Lunsford, Ethan Selfridge, Lois Black, and **Jan van Santen**
Center for Spoken Language Understanding
Oregon Health & Science University
heemanp@ohsu.edu

Abstract

Little research has been done to explore differences in the interactional aspects of dialogue between children with Autistic Spectrum Disorder (ASD) and those with typical development (TD). Quantifying the differences could aid in diagnosing ASD, understanding its nature, and better understanding the mechanisms of dialogue processing. In this paper, we report on a study of dialogues with children with ASD and TD. We find that the two groups differ substantially in how long they pause before speaking, and their use of fillers, acknowledgments, and discourse markers.

1 Introduction

Autism Spectrum Disorders (ASD) form a group of severe neuropsychiatric conditions whose features can include impairments in reciprocal social interaction and in communication (APA, 2000). These impairments may take different forms, ranging from individuals with little or no communication to fully verbal individuals with fluent, grammatically correct speech. In this latter verbal group, shortcomings in communication have been noted, including using and processing social cues during conversations. This is no surprise, since negotiating a conversation requires many abilities, several of which are generally impaired in ASD, such as generating appropriate prosody (Kanner, 1943) and "theory of mind" (Baron-Cohen, 2000).

We make a distinction between transactional and interactional aspects of dialogue (Brown and Yule, 1983). The transactional aspect refers to message content and interactional focuses on expressing social relations and personal attitudes. In this paper, we focus on surface behaviors that speakers use to help manage the interaction, namely turn-taking, and the use of fillers, discourse markers, and acknowledgments. One advantage of these behaviors is that they do not require complete understanding of the dialogue, and thus lend themselves to automatic analysis. In addition, these behaviors are under the speaker's control and should be robust to what the other speaker is doing. We hypothesize that just as interactional aspects in general are affected in ASD, so are these surface behaviors. However, to our knowledge, little or no work has been done on this.

Investigating how the interactional aspects of dialogue are affected in ASD serves several purposes. First, it can help in the diagnostic process. Currently, diagnosing ASD is subjective. Objective measures based on dialogue interaction could improve the reliability of the diagnostic process. Second, it can help us refine the behavioral phenotypes of ASD, which is critical for progress on the basic science front. Third, it can help us refine therapy for people with ASD to address dialogue interaction deficits. Fourth, understanding what dialogue aspects are affected in high-functioning verbal children with ASD can help determine which aspects of dialogue are primarily social in nature. For example, do speakers use fillers to signal that there is a communication problem, or are fillers a symptom of it (cf. Clark and Fox Tree, 2002)?

In this paper, we report on a study of interactional aspects of dialogues between clinicians and children with ASD. The dialogues were recorded during administration of the Autism Diagnostic Observation Schedule (Lord et al., 2000), which is an instrument used to assist in diagnosing ASD. We compare the performance of these children with a group of children with typical development (TD).

2 Data

The data used in this paper was collected during administration of the ADOS on 22 TD children and 26 with ASD, ranging in age from 4 to 8 years old. The children with ASD were high-functioning and verbal. The speech of the clinician and child was transcribed into utterance-like units, with a start and an end time. Activities were annotated in a separate tier. The transcriptions included the punctuation marks '.', '!', and '?' to mark syntactically and semantically complete sen-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 249–252,
The University of Tokyo, September 24-25, 2010.

tences, and '>' to mark incomplete ones. As a single audio channel was used, the timing of overlapping speech was marked as best as possible. Each child on average said 2221 words, 574 utterances, and 316 turns.

3 Results

Pauses between Turns: We first examine how long children wait before starting their turn. We hypothesized that children with ASD would wait longer on average to respond, either because they are less aware of (a) the turn-taking cues, (b) the social obligation to minimize inter-turn pauses, or (c) they have a slower processing and response times. For this analysis, we look at all turns in which there is no overlap between the beginning of the child's turn and the clinician's speech. Data is available on 4412 pauses for the TD children and 5676 for the children with ASD. The grand means of the children's pauses are shown in Table 1 along with the standard deviations. The TD children's average pause length is 0.876s. For the children with ASD, it is 1.115s, 27.3% longer. This difference is significant, *a-priori* independent t-test t=2.34 (df=39), p<.02 one-tailed.

	TD	ASD
all	0.876 (0.24)	1.115 (0.45)
after question	0.748 (0.25)	1.005 (0.40)
after non-question	1.076 (0.37)	1.329 (0.74)

Table 1: Pauses before new turns.

We also examine the pauses following questions by the clinician versus non-questions. Questions are interesting as they impose a social obligation for the child to respond, and they have strong prosodic cues at their ending. We identified questions as utterances transcribed with a question mark, which might include rhetorical questions. After a non-question (e.g., a statement), the average pause is 1.076s for the TD children and 1.329s for children with ASD. This difference is not statistically significant by independent t-test, t<1.6, NS. After a question, the average pause is 0.748s for the TD children and 1.005s for the children with ASD, a significant difference by *a-priori* independent t-test t=2.72 (df=42), p<.005 one-tailed. The ASD children on average take 34.4% longer to respond. Thus, after a question, the difference between children with TD and ASD is more pronounced.

Pauses by Activity: The ADOS includes having the child engage in different activities. For this research, we collapse the activities into three types: *converse* is when there is no non-speech task; *describe* is when the child is doing a mental task, such as describing a picture; and *play* is when the child is interacting with the clinician in a play session. To better understand the difference between questions and non-questions, we examine the pauses in each activity (Table 2).

	TD				ASD			
	question		non-ques.		question		non-ques.	
converse	0.730	0.30	0.656	0.27	0.890	0.34	0.932	0.88
describe	0.853	0.44	0.879	0.37	1.056	0.51	1.282	1.21
play	0.720	0.34	1.825	0.78	1.289	1.51	1.887	1.37

Table 2: Pauses for each type of activity.

After a question, the TD children tend to respond with similar pauses in each activity (the differences in column 2 between activities are not significant by pairwise paired t-test, all t's<1.6, NS). After a question, the child has a social obligation to respond, and this does not seem to be overridden by whether there is a separate task they are involved in. Even after a non-question, conversants have a social obligation to keep the speaking floor occupied and so to minimize inter-utterance pauses (Sacks et al., 1974). However, as seen in the third column, the pauses are affected by the type of activity, and the differences are statistically significant by pairwise paired t-test, (df=21), two-tailed: converse-describe t=2.24, p<.04; describe-play t=5.68, p<.0001; converse-play t=6.87, p<.0001. The biggest difference is with *play*. Here, it seems that the conversants physical interaction lessens the social obligation of maintaining the speaking floor. These findings are interesting for social-linguistics as it suggests that the social obligations of turn-taking are altered by the presence of a non-speech task.

We next compare the children with ASD to the TD children. For the *converse* activity, we see that the children with ASD take longer to respond, after questions and non-questions. The difference after questions is significant by independent t-test, t=1.74 (df=46) p<.05, one-tailed, whereas the difference after non-questions is marginal, t=1.47 (df=28) p<.08. This result could be explained by the slower processing and response times associated with ASD.

Just as with the TD children, we see that after a non-question, the children with ASD take longer to respond when there is another task. The differences in pause lengths between *converse* and *play* are significant, by paired t-test, t=2.89 (df=23)

p<.009, two-tailed. The difference between *describe* and *play* is marginal, t=2.03 (df=25) p<.06, and there was no significant difference between *converse* and *describe*, t<1, NS.

After a question, the children with ASD take longer to respond when there is another task, especially for *play*, although the pairwise differences in pause length between activities are not significant. This suggests that the children with ASD become distracted when there is another task, and so become less sensitive to either the question prosody or the social obligation of questions.

Fillers: We next examine the rate of fillers, at the beginning of turns, beginning of utterances, and in the middle of utterances. We look at these contexts individually as fillers can serve different roles, such as turn-taking, stalling for time or as part of a disfluency, and their role is correlated to their position in a turn. The rates are reported in Table 3, along with the total number of fillers within each category. Interestingly, the rate of 'uh' between children with TD and ASD is similar for all positions (independent t-test, all g's<1, NS).

	uh		um	
	TD	ASD	TD	ASD
turn init.	1.70% 112	1.84% 159	3.86% 243	1.65% 146
utt. init.	1.31% 43	1.20% 33	2.29% 73	0.52% 10
utt. medial	0.25% 103	0.31% 137	1.03% 492	0.21% 123

Table 3: Rate of fillers.

The more interesting finding, though, is in the usage of 'um'. Children with ASD use it significantly less than the TD children in every position, from $1/2$ the rate in turn-initial position to $1/5$ in utterance-medial position, independent two-tailed t-test: turn initial t=2.74 (df=38), p<.01; utterance initial t=2.53 (df=31), p<.02; and utterance medial t=3.94 (df=24), p<.001.

	TD		ASD	
converse	1.76%	569	0.56%	190
describe	1.15%	115	0.33%	31
play	0.96%	124	0.45%	58

Table 4: Use of 'um' by activity.

We also examined the overall usage of 'um' in each activity (Table 4). The TD children use 'um' more often in each activity than the children with ASD, and the differences are statistically significant by independent two-tailed t-test: converse t=3.62 (df=29), p<.002; describe t=2.83 (df=27), p<.01; play t=2.42 (df=33), p<.03. This result supports the robustness of the findings about 'um'.

Many researchers have speculated on the role of 'um' and 'uh'. In recent work, Clark and Fox Tree (2002) argued that they signal a delay, and that 'um' signals more delay than 'uh'. They view both as linguistic devices that are planned for, just as any other word is. Our work suggests that 'um' and 'uh' arise from different cognitive processes, and that the process that accounts for 'uh' is not affected by ASD, while the process for 'um' is.[1]

Acknowledgments: We next look at the rate of acknowledgments: single word utterances that are used to show agreement or understanding. Thus, the use of acknowledgments requires awareness of the other person's desire to ensure mutual understanding. As the corpus did not have these words explicitly marked, we identify a word as an acknowledgment if it meets the following criteria: (a) it is one of the words listed in Table 5 (based on Heeman and Allen, 1999); (b) it is first in the speaker's turn; and (c) it does not follow a question by the clinician. The TD children used acknowledgments in 17.42% of their turns that did not follow a question, while the children with ASD did this only 13.39% of the time (Table 5), a statistically significant difference by *a-priori* independent t-test t=1.78 (df=46), p<.05 one-tailed.

	TD		ASD	
total	17.42%	568	13.39%	459
yeah	7.49%	248	5.87%	215
no	2.78%	78	2.06%	63
mm-hmm	2.06%	75	1.07%	35
mm	0.99%	29	1.35%	42
ok	1.87%	65	0.83%	27
yes	0.92%	32	0.88%	32
right	0.14%	5	0.23%	8
hm	0.73%	21	0.69%	20
uh-huh	0.44%	15	0.42%	17

Table 5: Use of acknowledgments.

Discourse Markers: We next examine discourse markers, which are words such as 'well' and 'oh' that express how the current utterance relates to the discourse context (Schiffrin, 1987). We classified a word as a discourse marker if it was the first word in an utterance and is one of the words in Table 6 (Heeman and Allen, 1999). As shown in Table 6, the children with ASD use discourse markers significantly less than the TD children in both conditions by *a-priori* independent, one-tailed t-test: turn-initial t=3.24 (df=43) p<.002; utterance-initial t=4.01 (df=44) p<.0001.

[1] In Lunsford et al. (2010) we investigate the rate and length of pauses after 'uh' and 'um'. In addition, we verified the t-tests using Wilcoxon rank sum tests.

As can be seen, most of the difference is in the use of 'and'. The data for the other discourse markers was sparse, so we compared 'and' against all of the others combined. The decreased usage of 'and' in the ASD children is statistically significant for both conditions by *a-priori* independent, one-tailed t-test: turn-initial t=4.47 (df=30), p<.0001; utterance-initial t=3.79 (df=43), p<.0002. There is little difference in the use of all of the other discourse markers combined, and the difference is not statistically significant.

	Turn Initial				Utterance Initial			
	TD		ASD		TD		ASD	
all	19.2%	1290	12.8%	1196	28.7%	2053	19.4%	1330
and	10.7%	731	5.0%	471	19.5%	1419	12.0%	844
then	0.6%	38	1.0%	89	1.5%	97	1.4%	79
but	2.1%	144	1.3%	113	3.6%	238	2.7%	194
well	2.2%	143	2.7%	271	1.1%	74	1.2%	79
oh	2.0%	135	1.8%	160	1.0%	67	1.3%	68
so	1.2%	75	0.7%	60	1.6%	129	0.7%	49
wait	0.2%	9	0.2%	21	0.2%	17	0.2%	15
actually	0.2%	15	0.1%	11	0.2%	12	0.0%	2
not and	8.5%	559	7.8%	725	9.2%	634	7.4%	486

Table 6: Use of discourse markers.

The use of 'and' is also lower in each activity for the ASD children (Table 7), a significant difference by *a-priori* independent one-tailed t-test: converse t=3.00 (df=41), p<.003; describe t=4.79 (df=38), p<.0001, play t=4.07 (df=30), p<.0002.

	TD		ASD	
converse	13.36%	1139	7.95%	755
describe	21.77%	587	10.76%	339
play	12.97%	424	5.18%	221

Table 7: Use of 'and' in each activity.

One explanation for the decreased usage of 'and' and not the other discourse markers might be that, of all the discourse markers, 'and' seems to have the least meaning. It simply signifies that there is some continuation between the new speech and the previous context. This might make it difficult for children with ASD to learn its use. A second explanation is that the children with ASD are using 'and' correctly, but simply do not produce as many utterances that are related to the previous context (cf. Bishop et al., 2000).

4 Conclusion

In this paper, we examined a number of interactional aspects of dialogue in the speech of children with ASD and TD. We found that children with ASD have a lower rate of the filler 'um', acknowledgments, and the discourse marker 'and.' We also found that in certain situations, they take longer to respond. These deficits might prove useful for improved diagnosis of ASD. We also found that children with ASD have a lower rate of 'um' but not of 'uh', and that only the discourse marker 'and' seems to be affected. This might prove useful for both better understanding the nature of ASD as well as better understanding the role of these phenomena in dialogue. Although the results reported in this work are preliminary, they do show the potential of our approach. More work is needed to ensure that our automatic identification of turn-taking events, discourse markers, and acknowledgments is correct and to explore alternate explanations for the results that we observed.

Acknowledgments

Funding gratefully received from the National Institute of Heath under grants IR21DC010239 and 5R01DC007129, and the National Science Foundation under IIS-0713698. The views herein are those of the authors and reflect the views neither of the funding agencies.

References

American Psychiatric Association, Washington DC, 2000. *Diagnostic and Statistical Manual of Mental Disorders, 4th Edition, Text Revision (DSM-IV-TR)*.

S. Baron-Cohen. 2000. Theory of mind and autism: A review. In L. M. Glidden, editor, *International Review of Research in Mental Retardation*, volume 23: Autism, pages 170–184. Academic Press.

D. Bishop et al. 2000. Conversational responsiveness in specific language impairment: Evidence of disproportionate pragmatic difficulties in a subset of children. *Development and Psychopathology*, 12(2):177–199.

G. Brown and G. Yule. 1983. *Discourse Analysis*. Cambridge University Press.

H. Clark and J. Fox Tree. 2002. Using uh and um in spontaneous speaking. *Cognition*, 8:73–111.

P. Heeman and J. Allen. 1999. Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialog. *Computational Linguistics*, 25(4):527–572.

L. Kanner. 1943. Autistic disturbances of affective content. *Nervous Child*, 2:217–250.

C. Lord et al. 2000. The autism diagnostic observation schedule-generic: a standard measure of social and communication deficits associted with the spectrum of autium. *Journal of Austism Developmental Disorders*, 30(3):205–223, June.

R. Lunsford et al. 2010. Autism and the use of fillers: differences between 'um' and 'uh'. In *5th Workshop on Disfluency in Spontaneous Speech*, Tokyo.

H. Sacks, E. Schegloff, and G. Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.

D. Schiffrin. 1987. *Discourse Markers*. Cambridge University Press, New York.

Detection of time-pressure induced stress in speech via acoustic indicators *

Matthew Frampton, Sandeep Sripada, Ricardo Augusto Hoffmann Bion and Stanley Peters
Center for the Study of Language and Information
Stanford University, Stanford, CA, 94305 USA
{frampton@,ssandeep@,ricardoh@,peters@csli.}stanford.edu

Abstract

We use automatically extracted acoustic features to detect speech which is generated under stress, achieving 76.24% accuracy with a binary logistic regression. Our data are task-oriented human-human dialogues in which a time-limit is unexpectedly introduced partway through. Analysis suggests that we can detect approximately when this event occurs. We also consider the importance of normalizing the acoustic features by speaker, and detecting stress in new speakers.

1 Introduction

The term *stressed speech* can refer to speech generated under psychological stress (Sigmund et al., 2007). Stress alters an individual's mental and physiological state, which then affects their speech. The ability to identify stressed speech would be very valuable to Spoken Dialogue Systems (SDSs), especially in "stressful" applications such as search-and-rescue robots. Speech recognizers are usually trained on normal speech, and so can struggle badly on other speech. Techniques exist for making ASR robust to noise/stress (Hansen and Patil, 2007), but knowing when to apply them will in general require the ability to detect stressed speech. This ability is clearly also needed when the user's stress level should affect how the SDS responds. An SDS should sometimes generate stressed speech itself—for example, to impart a sense of urgency on the user.

This paper investigates spectral-based acoustic indicators of stress in human-human, task-oriented dialogues in which stress is induced in the latter stages by the unexpected introduction of time-pressure. Unlike previous studies, we detect stress in whole utterances in the raw audio, which is more realistic for applications. We also consider the importance of normalizing the features, and detection of both the introduction of the stressor, and stress in new speakers.

2 Related work

Stressors and clip sizes: The stressors in previous studies include logical problems, images of human bodies with skin diseases/severe accident injuries (Tolkmitt and Scherer, 1986), loss of control of a helicopter (Protopapas and Liberman, 2001), university examinations (Sigmund et al., 2007), and an increasingly difficult air controller simulation and verbal quiz (Scherer et al., 2008). Sigmund et al. (2007) detect stress in approximately 2000 voiced segments of 5 vowels. Tolkmitt and Scherer (1986), Protopapas and Liberman (2001) and Scherer et al. (2008) detect stress in whole utterances, but these are respectively, read from a card, quiz answers, and with verbal content removed. Studies on the Speech under Simulated and Actual Stress (SUSAS) corpus (Hansen and Bou-Ghazale, 1997) detect stress in words. These include (Hansen, 1996; Zhou, 1999; Hansen and Womack, 1996; Zhou, 2001; Casale et al., 2007). The SUSAS corpus contains aircraft communication words from a common highly confusable vocabulary set of 35, and they are divided into different speaking styles.

Acoustic cues: The most widely investigated acoustic cues relate to *fundamental frequency* (F0, also called pitch), formant frequencies and spectral composition e.g. (Tolkmitt and Scherer, 1986; Hansen, 1996; Zhou, 1999; Protopapas and Liberman, 2001; Sigmund et al., 2007; Scherer et al., 2008). Mel-Frequency Cepstral Coefficients

*The research reported in this paper was sponsored by the Department of the Navy, Office of Naval Research, under grant number N00014-017-1-1049. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 253–256,
The University of Tokyo, September 24-25, 2010.

Category	Examples
F0-related	Median, mean, minimum, time of minimum as % thr' clip, max, time of max as % thr' clip, range (max-min), standard deviation, mean absolute slope, mean slope without octave jumps, number of voiced frames.
Intensity-related	Median, mean, minimum, time of minimum as % thr' clip, max, time of max as % thr' clip, range (max-min), standard deviation.
Formant-related (for F1-F3)	Mean, minimum, time of minimum as % through clip, max, time of max as % through clip, range (max-min).
Spectral tilt-related	Mean, minimum, maximum, range (max-min).

Table 1: The acoustic features which are extracted from the audio clips using Praat (Boersma and Weenink, 2010).

(MFCCs)[1] and Teager Energy Operator (TEO)[2] (Kaiser, 1990) based features have also been considered e.g. (Hansen and Womack, 1996; Zhou, 2001; Casale et al., 2007).

Features of all these types have proved useful in detecting stressed speech. The classification methods employed are various, including a traditional binary hypothesis detection-theory method (Zhou, 1999) and neural networks (Hansen and Womack, 1996; Scherer et al., 2008), while Casale et al. (2007) used genetic algorithms for feature selection. Of the two more recent studies which detected stress in whole utterances, Protopapas and Lieberman found that mean and maximum F0 within an utterance correlate highly with subject stress ratings, and Scherer et al.'s neural network outperformed a human baseline. Note that findings/results in these and other previous studies are not directly comparable with our own, because we detect stress in whole utterances in raw audio.

3 Data

The original data (Eberhard et al., 2010) are 4 task-oriented dialogues between 2 native English-speaking participants. Hence there are 8 speakers in total (7 male, 1 female), and the dialogues contain 263, 172, 228 and 210 utterances respectively.

During a dialogue, the participants (the *director* and *member*) are on a floor with corridors and rooms that contain various colored boxes. The director stays in one room, and gives task instructions via walkie-talkie to the member, providing directions with a map which is partially complete and accurate for box locations. The tasks are locating boxes which are unmarked on the map, and transferring blocks between and retrieving specified boxes. Initial instructions do not mention a time limit, but at the end of the 7^{th} minute, the director is given a timer and told there are 3 minutes to complete the current tasks, plus one new task.

We use the Nuance speech recognizer (V. 9.0) to end-point each dialogue's audio signal, and the resulting clips are mostly 1 to 3 seconds. In preliminary experiments (not reported), denoising seemed to remove acoustic information which is indicative of stress. Hence we use raw audio.

Stressed speech: For present purposes, we assume that all speech after the introduction of the time limit is stressed. Hence 448 of the 663 audio clips in our experimental data are unstressed, and 215 are stressed. In future we plan to use the *Amazon Mechanical Turk* to obtain perceived stress ratings on a scale with more gradations.

4 Experiments

Acoustic features: We use Praat (Boersma and Weenink, 2010) to compute *F0*, *intensity*, *formant* and *spectral tilt-related* features for each clip (Table 1). F0 (pitch) corresponds to the rate of vocal cord vibration in Hertz (*Hz*), and Intensity, to the sound's loudness in decibels (*dB*), (derived from the amplitude or increase in air pressure). A formant is a concentration of acoustic energy around a particular frequency in the speech wave. There are several, each corresponding to a resonance in the vocal tract, and we consider the lowest three (*F1-F3*). Spectral tilt measures the difference in energy between the 1^{st} and 2^{nd} formants, and so estimates the degree to which energy at the fundamental dominates in the glottal source waveform.

Comparing different normalization methods: We evaluate binary logistic regression models with 10-fold cross-validation, and try the following 4 methods for normalizing each clip's acoustic features according to its speaker.

- *Maximum normalization*: Due to the possibility of outliers, we divide each feature value

[1] MFCCs model the human auditory system's nonlinear filtering in measuring spectral band energies.

[2] The TEO is a nonlinear operator which uses mechanical and physical considerations to extract the signal energy.

Normalization	% Accuracy		US %correct		S %correct		MCB	
Maximum normalization	74.4	(74.25)	86.67	(85.05)	48.5	(54.3)	67.8	(67.1)
Z-score	73.5	(73.78)	84.89	(84.12)	49.53	(52.7)	67.8	(67.1)
US Average	75.61	(76.24)	86.63	(86.2)	53.5	(55.9)	67.8	(67.1)
S Average	75.31	(75)	84.67	(84.375)	55.6	(55.9)	67.8	(67.1)
No normalization	68.52	(70.45)	84.34	(82.8)	37.4	(45.2)	67.8	(67.1)

Table 2: Binary logistic regression 10-fold cross validation with different feature normalization approaches: Scores within brackets are when the female speaker data is removed; S = Stressed, US = Unstressed, MCB = Majority Class Baseline.

by the 95^{th} percentile value for that feature, rather than the maximum.

- *Z-score*: Using the mean and standard deviation for each feature, the feature vector is converted to Z-scores[3].
- *Unstressed (US) average*: Each feature is normalized by its mean value in the unstressed region.
- *Stressed (S) average*: Each feature is normalized by its mean value in the stressed region.

Table 2 shows the results. All those generated with feature normalization are significantly better ($p < 0.005$) than the majority class baseline (MCB), (i.e. classifying all utterances as unstressed). Without normalization, the overall accuracy drops about 5—6%, and the stressed speech class about 11—18%. Different normalization methods do not produce very different results, but US average gives the best overall accuracy (75.61%). When we remove the female speaker, this increases to 76.24%, and feature normalization remains important.

We also tested our assumption that the speech before and after the introduction of time-pressure is unstressed and stressed respectively, by checking that they really are different. As before, we considered 7 minutes unstressed, and 3 stressed, and used US average normalization. However we now assigned different minutes to the unstressed and stressed categories: first we swapped the 6^{th} and 8^{th}, then also the 5^{th} and 9^{th}, and then also the 7^{th} and 10^{th}. As a result, classification accuracy dropped, (to 75%, then 68.71%, then 67.66%), which supports our assumption.

Feature contribution analysis: Table 3 shows the US average normalized features with *information gain* greater than zero. Intensity and pitch features are ranked most predictive (i.e. maximum and mean intensity, and mean and median pitch), but *Spectral tilt mean* and a couple of formant features are also predictive. In general, higher values for the most predictive pitch and intensity features (e.g. *Intensity max* and *Pitch mean*) seem to indicate stress. An interaction term for *Intensity max* and *Pitch mean* caused a significant improvement in the fit of the model—the χ^2 value (or change in the -2 Log likelihood) was 4.952 ($p < 0.05$).

Feature	Info. Gain
Intensity max	.101
Pitch mean	.099
Intensity mean	.099
Pitch median	.088
Pitch max	.059
Intensity min	.046
Spectral tilt mean	.042
Pitch min	.041
F1 min	.038
Intensity range	.034
Intensity std. dev.	.033
F3 range	.033
Intensity median	.031

Table 3: Unstressed average normalized features ranked by information gain.

Detecting the introduction of the stressor: Figure 1 shows the percentage of audio clips in each minute that were classified as stressed. As we would hope, there is a dramatic increase from the 7^{th} to the 8^{th} minute (around 20% to over 50%). Such an increase could be used to detect the introduction of the stressor, time-pressure.

Detecting stress in new speakers: To detect stressed speech in new speakers, we evaluate the logistic regression with an 8-fold cross-validation, in each fold training on 7 speakers, and testing on the other. We apply US average normalization, initially with the average values for the new speaker's unstressed speech, and then with the average values in unstressed speech across all "seen" speakers (speakers in the training set). Evaluation scores (Table 4) are now lower, especially for the latter approach, but the former remains significantly

[3]A Z-score indicates the number of standard deviations between an observation and the mean.

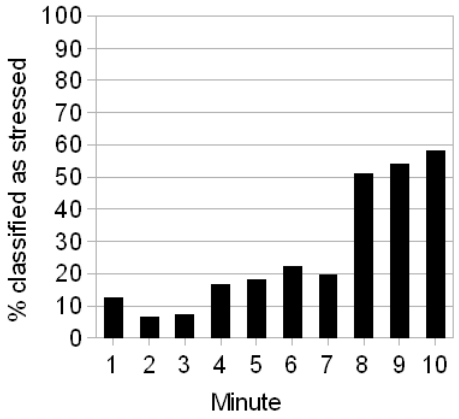


Figure 1: The percentage of clips in each minute of the dialogues which our classifier marks as stressed, (note that time-pressure is introduced at the end of minute 7).

better than the MCB. Since the female speaker's stress class F-score is 0, we tried normalizing the 7 male speakers based on only seen male data, and then average accuracy for a male rose from 67.09% to 68.02% (not statistically significant).

Spkr	% Accuracy		F-unstress		F-stress	
1	62.5	(62.2)	.77	(.74)	.07	(0.34)
2	75	(75)	.84	(.84)	.09	(0.44)
3	57.6	(72.9)	.62	(.81)	.49	(0.5)
4	71.73	(74)	.84	(.83)	0	(0.43)
5	71.62	(73.0)	.77	(.77)	.62	(0.68)
6	77.6	(80.4)	.86	(.88)	.51	(0.52)
7	64.36	(65.6)	.74	(.77)	.4	(0.35)
8	60.97	(71.7)	.67	(.8)	.49	(0.46)
Av.	67.67	(71.9)	.76	(.80)	.33	(0.46)

Table 4: Predicting stress in new speakers: New speaker features are normalized based on unstressed speech for all speakers in training set (unbracketed) and on their own unstressed speech (bracketed). Speaker 4 is the female.

5 Conclusion

For detecting stressed speech, we demonstrated the importance of normalizing acoustic features by speaker, and achieved 76.24% classification accuracy with a binary logistic regression model. The most indicative features were maximum and mean intensity within an utterance, and mean and median pitch. After the introduction of time-pressure, the percentage of clips classified as stressed increased dramatically, showing that it is possible to detect approximately when this event occurs. We also attempted to detect stressed speech in new speakers, and as expected, results were poorer.

In future work we plan to expand our data-set with more dialogues, and test accuracy for detecting the introduction of the stressor. We want to use MFCCs and TEO features, and also non-acoustic features such as disfluency features. As mentioned previously, we also hope to move beyond binary classification, by acquiring perceived stress ratings on a scale with more gradations.

References

P. Boersma and D. Weenink. 2010. Praat: doing phonetics by computer (version 5.1.29). Available from http://www.praat.org/. [Computer program].

S. Casale, A. Russo, and S. Serrano. 2007. Multistyle classification of speech under stress using feature subset selection based on genetic algorithms. *Speech Communication*, 49:801–810.

K. Eberhard, H. Nicholson, S. Kubler, S. Gundersen, and M. Scheutz. 2010. The Indiana "Cooperative Remote Search Task" (CReST) Corpus. In *Proc. of LREC*.

J.H.L. Hansen and S. Bou-Ghazale. 1997. Getting started with SUSAS: a Speech Under Simulated and Actual Stress database. In *Eurospeech-97: International Conference on Speech Communication and Technology*.

J. Hansen and S. Patil, 2007. *Speaker Classification I: Fundamentals, Features, and Methods*, chapter Speech Under Stress: Analysis, Modeling and Recognition, pages 108–137. Springer-Verlag, Berlin, Heidelberg.

J. Hansen and B. Womack. 1996. Feature analysis and neural network based classification of speech under stress. *IEEE Transactions on Speech & Audio Processing*, 4(4):307–313.

J. Hansen. 1996. Analysis and compensation of speech under stress and noise for environmental robustness in speech recognition. *Speech Communications, Special Issue on Speech Under Stress*, 20(2):151–170.

J.F. Kaiser. 1990. On a simple algorithm to calculate the energy of a signal. In *Proc. of ICASSP*.

A. Protopapas and P. Liberman. 2001. Fundamental frequency of phonation and perceived emotional stress. *Journal of the Acoustical Society of America*, 101(4):2267–2277.

S. Scherer, H. Hofmann, M. Lampmann, M. Pfeil, S. Rhinow, F. Schwenker, and G. Palm. 2008. Emotion recognition from speech: Stress experiment. In *Proc. of LREC*.

M. Sigmund, A. Prokes, and Z. Brabec. 2007. Statistical analysis of glottal pulses in speech under psychological stress. In *Proc. of the 16th European Signal Processing Conference*.

F. J. Tolkmitt and K. R. Scherer. 1986. Effect of experimentally induced stress on vocal parameters. *Journal of Experimental Psychology*, 12(3):302–313.

G. Zhou. 1999. *Nonlinear speech analysis and acoustic model adaptation with applications to stress classification and speech recognition*. Ph.D. thesis, Duke University.

G. Zhou. 2001. Nonlinear feature based classification of speech under stress. *IEEE Transactions on Speech & Audio Processing*, 9:201–216.

How to Drink from a Fire Hose:
One Person Can Annoscribe 693 Thousand Utterances in One Month

David Suendermann, Jackson Liscombe, Roberto Pieraccini
SpeechCycle Labs
New York, USA
{david, jackson, roberto}@speechcycle.com

Abstract

Transcription and semantic annotation (annoscription) of utterances is crucial part of speech performance analysis and tuning of spoken dialog systems and other natural language processing disciplines. However, the fact that these are manual tasks makes them expensive and slow. In this paper, we will discuss how annoscription can be partially automated. We will show that annoscription can reach a throughput of 693 thousand utterances per person month under certain assumptions.

1 Introduction

Ever since spoken dialog systems entered the commercial market in the mid 1990s, the caller's speech input is subject to collection, transcription, and often also semantic annotation. Utterance transcriptions and annotations (annoscriptions) are used to measure speech recognition and spoken language understanding performance of the application. Furthermore, they are used to improve speech recognition and application functionality by tuning grammars, introducing new transitions in the call flow to cover more of the callers' demands, or changing prompt wording or application logic to influence the speech input. Annoscriptions are also crucial for training statistical language models and utterance classifiers for call routing or other unconstrained speech input contexts (Gorin et al., 1997). Since very recently, statistical methods are used to replace conventional rule-based grammars in every recognition context of commercial spoken dialog systems (Suendermann et al., 2009b). This replacement is only possible by collecting massive amounts of annoscribed data from all contexts of an application. To give the reader an idea of what *massive* means in this case, in (Suendermann et al., 2009b), we used 2,184,203 utterances to build a complex call routing system. In (Suendermann et al., 2009a), 4,293,898 utterances were used to localize an English Internet troubleshooting application to Spanish.

Considering that professional service providers may charge as much as 50 US cents for annoscribing a single utterance, the usage of these amounts of data seems prohibitive since costs for such a project could potentially add up to several million US dollars. Furthermore, one has to consider the average speed of annoscription which rarely exceeds 1000 utterances per hour and person. This means that the turn-around of a project as mentioned above would be several years unless teams of many people work simultaneously. However, the integration of the work of a large team becomes the more tricky the more people are involved. This is especially true for the annotation portion since it requires a thorough understanding of the spoken dialog system's domain and design and very often can only be conducted under close supervision by the interaction designer in charge of the project. Furthermore, there are crucial issues related to intra- and inter-labeler inconsistency becoming more critical the more people work on the same or similar recognition contexts of a given project.

This paper is to show how it is possible to automate large portions of both transcription and annotation while meeting human performance[1] standards. As an example case, we show how the proposed automation techniques can increase annoscription speed to nearly 693 thousand utterances per person and month.

2 Automatic Transcription

2.1 Two Fundamentals

Automatic transcription of spoken utterances may not sound as something new to the reader. In fact, the entire field of automatic speech recognition is about machine transcription. So, why is it worth dedicating a full section to something well-covered in research and industry for half a century? The reason is the demand for *achieving human performance* as formulated in the introduction which, as is also well-known, cannot be satisfied by any of the large-vocabulary speech recognizers ever developed. In order to demonstrate that there is indeed a way to achieve human transcription performance using automatic speech recognition, we would like to refer to two fundamental observations on the performance of speech recog-

[1] In this paper, *performance* stands for *quality* or *accuracy* of transcription or annotation. It does not refer to *speed* or *throughput*.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 257–260,
The University of Tokyo, September 24-25, 2010.

nition:

(1) Speech recognition performance can be very high for contexts of constrained vocabulary. An example is the recognition of isolated letters in the scope of a name spelling task as discussed in (Waibel and Lee, 1990) that achieved a word error rate of only 1.1%. In contrast, the word error rate of large-vocabulary continuous speech recognition can be as high as 40 to 65% on telephone speech (Yuk and Flanagan, 1999).

(2) The positive dependence between speech recognition performance and amount of data used to train acoustic and language models, so far, did not reach a saturation point even considering billions of training tokens (Och, 2006).

Both of these fundamentals can be applied to the transcription task for utterances collected on spoken dialog production systems as follows:

(1) The vocabulary of spoken dialog systems can be rather complex. E.g., the caller utterances used for the localization project mentioned in Section 1 distinguish more than 13,000 types. However, the nature of commercial spoken dialog applications being mostly system-driven strongly constrains the vocabulary in many recognition contexts. E.g., when the prompt reads

> You can say: *recording problems*, *new installation*, *frozen screen*, or *won't turn on*

callers mostly respond things matching the proposed phrases, occasionally altering the wording, and only seldomly using completely unexpected utterances.

(2) The continuous data feed available on high-traffic spoken dialog systems in production processing millions of calls per month can provide large numbers of utterances for every possible recognition context. Even if the context appears to be of a simple nature, as for a yes/no question, the continuous collection of more data will still have an impact on the performance of a language model built using this data.

2.2 How to Achieve Human Performance

Even though we have suggested that the recognition performance in many contexts of spoken dialog systems may be very high, we have still not shown how our observations can be utilized to achieve *human performance* as demanded in Section 1. How would a context-dependent speech recognizer respond when the caller says something completely unexpected such as *let's wreck a nice beach* when asked for the cell phone number? While a human transcriber may still be able to correctly transcribe this sentence, automatic speech recognition will certainly fail even with the largest possible training set. The answer to this question is that the speech recognizer should *not respond at all* in this case but admit that it had trouble recognizing this utterance. Rejection of hypotheses based on confidence scores is common practice in many speech and language processing tasks and is heavily used in spoken dialog systems to avoid mis-interpretation of user inputs.

So, we now know that we can limit automatic transcriptions to hypotheses of a minimum reliability. However, how do we prove that this limited set resembles *human performance*? What is actually *human performance*? Does the human make errors transcribing? And, if so, how do we measure human error? What do we compare it against?

To err is human. Accordingly, there is an error associated with manual transcription which can only be estimated by comparing somebody's transcription with somebody else's due to a lack of ground truth. Preferably, one should have a good number of people transcribe the same speech utterances and than compute the average word error rate comparing every transcription batch with every other producing a reliable estimate of the manual error inherent to the transcription task of spoken dialog system utterances. In order to do so, we compared transcriptions of 258,843 utterances collected from a variety of applications and recognition contexts partially shared by up to six transcribers and found that they averaged at an inter-transcriber word error rate of $\text{WER}_0 = 1.3\%$.

Now, for every recognition context a language model had been trained, we performed automatic speech recognition on held-out test sets of $N = 1000$ utterances producing N hypotheses and their associated confidence scores $P = \{p_1, \ldots p_N\}$. Now, we determined that minimum confidence threshold p_0 for which the word error rate between the set of hypotheses and manual reference transcriptions was not statistically significantly greater than WER_0:

$$p_0 = \arg\min_{p \in P} \text{WER}(V(p)) \;\tilde{\not>}\; \text{WER}_0; \quad (1)$$

$$V(p) = \{\nu_1, \ldots, \nu_K\} : \nu_k \in \{1, \ldots, N\}, p_{\nu_k} \geq p.$$

Statistical significance was achieved when the delta resulted in a p value greater than 0.05 using the χ^2 calculus. For the number of test utterances, 1000, this point is reached when the word error on the test set falls below $\text{WER}_1 = 2.2\%$. This means that Equation 2.2's 'not statistically significantly greater than' sign can be replaced by a regular smaller-than sign as

$$\text{WER} \;\tilde{\not>}\; \text{WER}_0 \Leftrightarrow \text{WER} < \text{WER}_1. \quad (2)$$

This essentially means that there is a chance that the error produced by automatic transcription *is* greater than that of manual transcription, however, on the test set it could not be found to be of significance. Requesting to lower the p value or even demanding that the test set performance falls below the reported manual error can drastically lower the automation rate and, in the latter case, is not even reasonable—how can a machine possibly commit

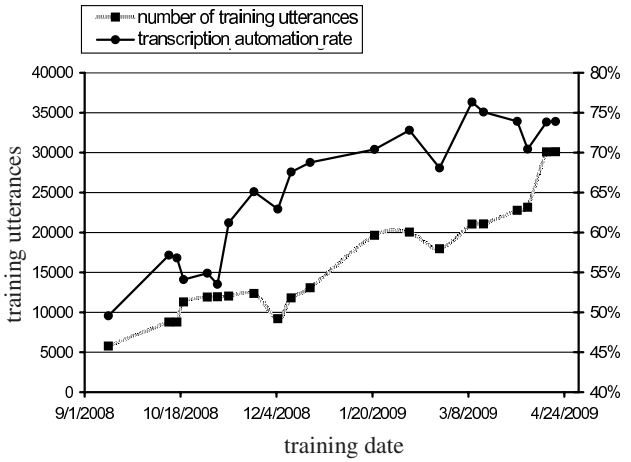


Figure 1: Dependency between amount of training data and transcription automation rate

less errors than a human being as it is trained on human transcriptions?

As a proof of concept, we ran automatic transcription against the same set of utterances used to determine the manual transcription error, and we found that the average word error rate between manual and automatic annotation was as low as 1.1% for all utterances whose confidence score exceeded the context-dependent threshold trained as described above. In this initial experiment, a total of 60,608 utterances, i.e., 23.4%, had been automated.

2.3 On Automation Rate

Formally, transcription automation rate is the ratio of utterances whose confidence exeeded p_0 in Equation 2.2:

$$\text{transcription automation rate} = \frac{|V(p_0)|}{N} \quad (3)$$

where $|V|$ refers to the cardinality of the set V, i.e., the number of V's members.

The above example's transcription automation rate of 23.4% does not yet sound tremendously high, so we should look at what can be done to increase the automation rate as much as possible. It is predictable that the two fundamentals formulated in Section 2.1 have a large impact on recognition performance and, hence, the transcription automation rate:

(1) In large-scale experiments, we were able to show a significant (negative) correlation between the annotation automation rate and task complexity. Since this study does not fit the present paper's scope, we will refrain from reporting on details at this point.

(2) As an example which influence the amount of training data can have on the transcription automation rate, Figure 1 shows statistics drawn from twenty runs of language model training carried out over the course of seven months while collecting more and more data.

3 Automatic Annotation

Semantic annotation of utterances into one of a final set of classes is a task which may require profound understanding of the application and recognition context the specific utterances were collected in. Examples include simple contexts such as yes/no questions which may be easily manageable also by annotators unfamiliar with the application, high-resolution open prompt contexts with hundreds of technical and highly application-specific classes, or number collection contexts allowing for billions of classes. All these contexts can benefit from two rules which help to significantly reduce an annotator's workload:

(A) **Never do anything twice.** This simple statement means that there should be functionality built into the annotation software or the underlying database that

- lets the annotator process multiple utterances with identical transcription in a single step and
- makes sure that whenever a new utterance shows up with a transcription identical to a formerly annotated one, the new utterance gets assigned the same class automatically.

Figure 2 demonstrates the impact of Rule (A) with two typical examples. The first is a yes/no context allowing for the additional global commands *help*, *hold*, *agent*, *repeat*, and *i don't know*. The other is an open prompt context distinguishing 79 classes.

When using the token/type distinction, the impact of Rule (A) is that annotation effort becomes linear with the number of *types* to work on. While the ratio between types and tokens in a given corpus can be very small (i.e., the automation rate is very high, e.g., 95% in the above yes/no example), this ratio reaches saturation at some point. In the yes/no example, there is only a gradual difference between the automation rates for 10 thousand and 1 million utterances. Hence, at a certain point, the effort becomes virtually linear with the number of *tokens* to be processed.

(B) **Predict as much as possible.** Most of the recognition contexts for which utterances are transcribed and annotated use grammars to implement speech recognition functionality. Many of these

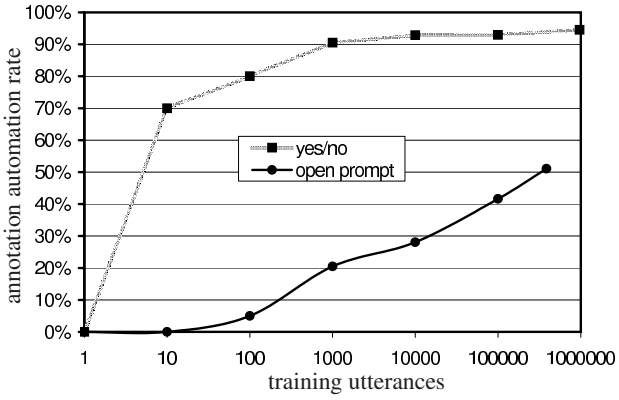


Figure 2: Dependency between number of collected utterances and annotation automation rate based on Rule (A) for two different contexts

Table 1: Annotation automation rates for three different recognition contexts based on Rule (B)

grammar	#symptoms	ann. auto. rate
modem type	43	70.3%
blue/black/snow	10	77.0%
yes/no	10	88.6%

grammars will be rule-based grammars. Even if the grammars are statistical, most often, earlier in time, rule-based grammars had been used in the same recognition context. Hence, we can assume that we are given rule-based grammars for many recognition contexts of the dialog system in question. Per definition, rule-based grammars shall contain canonical rules expressing the relationship between expected utterances in a given context and the semantic classes these utterances are to be associated with. Consequently, whenever for an utterance recorded in the context under consideration there is a rule in the grammar, it provides the correct class for this utterance, and it can be excluded from annotation. These rules can be strongly extended to allow for complex prefix and suffix rules, repetitions, sub-grammars &c. making sure that the majority of utterances will be covered by the rule-based grammars thereby minimizing the annotation effort. Table 1 shows three example grammars of different complexity: One that collects the type of the caller's modem, one for the identification of a TV set's picture color (blue/black/snow), and a yes/no context with global commands. Annotation automation rates for these grammars that were not specifically tuned for maximizing automation but directly taken from the production dialog systems varied between 70.3% and 88.6%.

To never ever touch a formerly annotated utterance type again and to blindly rely on (mayby outdated or erroneous) rule-based grammars to provide baseline annotations may result in annotation mistakes, possibly major ones when frequent utterances are concerned. So, how do we make sure that high annotation performance standards are met?

To answer this question, the authors have developed a set of techniques called C^7 taking care of completeness, consistency, congruence, correlation, confusion, coverage, and corpus size of an annotation set (Suendermann et al., 2008). The mentioned techniques are also useful in the frequent event of changes to the number or scope of annotation classes. This can happen e.g. due to functional changes to the application, changes to prompts, user behavior, or to contexts preceeding the current annotation context. Another frequent reason is the introduction of additional classes to enlarge the scope of the current context[2].

[2]In a specific context, callers may be asked whether they want A, B, or C, but they may respond D. The introduction of a new class D which the application is able to handle requires the re-annotation of all utterances falling into D's scope.

4 693 Thousand Utterances

Finally, we want to return to the initial statement of this paper claiming that one person is able to annoscribe 693 thousand utterances within one month. An approximated automation rate of 80% for transcription and 90% for annotation is possible when there is already a massive database of annoscriptions available to be exploited for automation. These rates result in about 139 thousand transcriptions and 69 thousand annotations outstanding. At a pace of 1000 transcribed or 2000 annotated utterances per hour, the required time would be 139 hours transcription and 35 hours annotation which averages at 40 hours per week[3].

5 Conclusion

This paper has demonstrated how automated annoscription of utterances collected in the production scope of spoken dialog systems can effectively accelerate this conventionally entirely manual effort. When allowing for some overtime, we have shown that a single person is able to produce 693 thousand annoscriptions within one month.

References

A. Gorin, G. Riccardi, and J. Wright. 1997. How May I Help You? *Speech Communication*, 23(1/2).

F. Och. 2006. Challenges in Machine Translation. In *Proc. of the TC-Star Workshop*, Barcelona, Spain.

D. Suendermann, J. Liscombe, K. Evanini, K. Dayanidhi, and R. Pieraccini. 2008. C^5. In *Proc. of the SLT*, Goa, India.

D. Suendermann, J. Liscombe, K. Dayanidhi, and R. Pieraccini. 2009a. Localization of Speech Recognition in Spoken Dialog Systems: How Machine Translation Can Make Our Lives Easier. In *Proc. of the Interspeech*, Brighton, UK.

D. Suendermann, J. Liscombe, K. Evanini, K. Dayanidhi, and R. Pieraccini. 2009b. From Rule-Based to Statistical Grammars: Continuous Improvement of Large-Scale Spoken Dialog Systems. In *Proc. of the ICASSP*, Taipei, Taiwan.

A. Waibel and K.-F. Lee. 1990. *Readings in Speech Recognition*. Morgan Kaufmann, San Francisco, USA.

D. Yuk and J. Flanagan. 1999. Telephone Speech Recognition Using Neural Networks and Hidden Markov Models. In *Proc. of the ICASSP*, Phoenix, USA.

[3]The original title of this paper claimed that one person could annoscribe even one million utterances in a month. However, after receiving multiple complaints about the unlawfulness of a 58-hour workweek, we had to change the title accordingly to avoid disputes with the Department of Labor. Furthermore, as discussed earlier, at the starting point of an annoscription project, automation rates are much lower than later.

Advances in the Witchcraft Workbench Project

Alexander Schmitt, Wolfgang Minker
Institute for Information Technology
University of Ulm, Germany
alexander.schmitt,
wolfgang.minker@uni-ulm.de

Nada Ahmed Hamed Sharaf
German University in Cairo, Egypt
nada.sharaf@student.guc.edu.eg

Abstract

The *Workbench for Intelligent exploraTion of Human ComputeR conversaTions* is a new platform-independent open-source workbench designed for the analysis, mining and management of large spoken dialogue system corpora. What makes Witchcraft unique is its ability to visualize the effect of classification and prediction models on ongoing system-user interactions. Witchcraft is now able to handle predictions from binary and multi-class discriminative classifiers as well as regression models. The new XML interface allows a visualization of predictions stemming from any kind of Machine Learning (ML) framework. We adapted the widespread CMU Let's Go corpus to demonstrate Witchcraft.

1 Introduction

Substantial effort has been invested in the past years in exploring ways to render Spoken Dialogue Systems (SDS) more adaptive, natural and user friendly. Recent studies investigated the recognition of and adaption to specific user groups, e.g. the novices and expert users, or the elderly (Bocklet et al., 2008). Further, there is a massive effort on recognizing angry users, differentiate between genders (Burkhardt et al., 2007), spotting dialects, estimating the cooperativeness of users or user satisfaction (Engelbrecht et al., 2009) and finally, predicting task completion (Walker et al., 2002). When applied *online*, i.e. during the interaction between user and system, these models can add valuable information to the dialogue system which would allow for an adaption of the dialogue strategy, see Figure 1.

Until now we can report that these models[1] work more or less well in batch-test scenarios offline. An anger classifier might deliver 74% accuracy when evaluated on utterance level. But which impact would the deployment of this recognizer have on specific dialogues when being employed in a real system? Would it fail or would it succeed? Similarly, at what point in time would models predicting gender, speaker age, and expert status deliver a reliable statement that can indeed be used for adapting the dialogue? What we need prior to deployment is an evaluation of the models and a statement on how well the models would work when being shifted on dialogue level. At this point, the Witchcraft Workbench enters the stage.

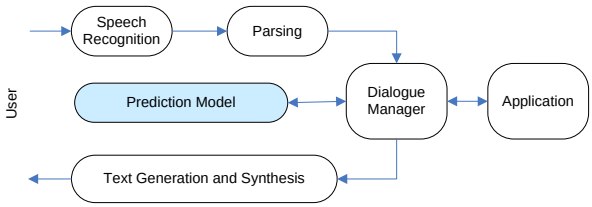


Figure 1: Enhanced SDS: The prediction model that is used to render the dialogue system more user-friendly delivers additional information to the dialogue manager.

2 The Role of Witchcraft

For a more detailed introduction on the Witchcraft Workbench please refer to (Schmitt et al., 2010a). In a nutshell, Witchcraft allows managing, mining and analyzing large dialogue corpora. It brings logged conversations back to life in such that it simulates the interaction between user and system based on system logs and audio recordings. Witchcraft is first of all not an annotation or transcription tool in contrast to other workbenches such as NITE (Bernsen et al., 2002), Transcriber[2]

[1] please note that we use the expression recognizer, classifier and prediction model interchanging in this context

[2] http://trans.sourceforge.net

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 261–264,
The University of Tokyo, September 24-25, 2010.

or DialogueView[3]. Although we also employ it for annotation, its central purpose is a different one:Witchcraft contrasts dialogue flows of specific dialogues which are obtained from a dialogue corpus with the estimations of arbitrary prediction and classification models. By that it is instantly visible which knowledge the dialogue system would have at what point in time in the dialogue. Imagine a dialogue system would be endowed with an anger recognizer, a gender recognizer and a recognizer that should predict the outcome of a dialogue, i.e. task completion. Each of the three recognizers would be designed to deliver an estimation at each point in the dialogue. How likely is the user angry? How likely is he male or female and how likely will the task be completed based on what we have seen so far in the dialogue. To which extent the recognizers deliver a correct result can be verified within Witchcraft.

3 Handling Models in Witchcraft

Witchcraft had several shortcomings when we first reported on it in (Schmitt et al., 2010a). It was only working with a proprietary industrial corpus and was heavily tailored to our needs. It worked only with specific models from binary discriminative classifiers. Since then we have put substantial effort to generalize the functionality and to make it available to the community.

To allow an analysis of other recognizers the system has been extended to further handle predictions from multiclass discriminative classification and regression tasks. Witchcraft does not contain "intelligence" on its own but makes use of and manages the predictions of recognizers. We assume that a recognizer is implemented either as stand-alone recognizer or with help of a Machine Learning (ML) framework. We emphasize that Witchcraft itself does neither perform feature extraction nor classification. Witchcraft operates on turn level requesting the recognizer to deliver a prediction based on information available at the currently processed dialogue turn of a specific dialogue. Where and how the recognizer accomplishes this is not part of the architecture. The ML framework of our choice that was originally supported natively, i.e. directly accessed by Witchcraft (Schmitt et al., 2010a) was RapidMiner[4], an ML framework that covers a vast majority of supervised and unsupervised machine learning techniques. The initial plan to interface other ML frameworks natively (such as MatLab, the R framework, BoosTexter, Ripper, HTK that are frequently used in research) turned out not to be practical. In order to still be able to cover the broadest possible range of ML tools we introduced a new generic XML interface. For simplicity we removed the RapidMiner interface. An overview of the dependency between Witchcraft and a recognizer is depicted in Figure 2.

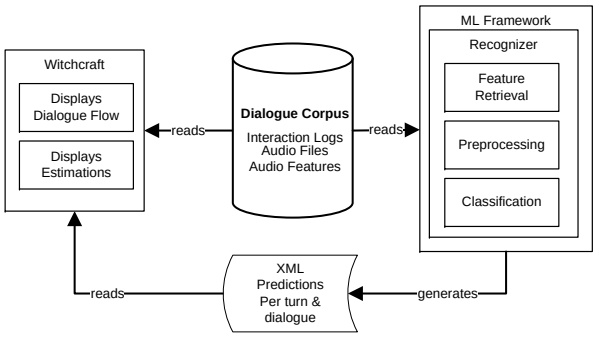


Figure 2: Dependency of Witchcraft and related recognizers that are implemented within an ML framework.

Witchcraft has been extended to support an arbitrary number of models, see Figure 3. They can now be one of the types "discriminative binary", "discriminative multiclass classification" and "regression".

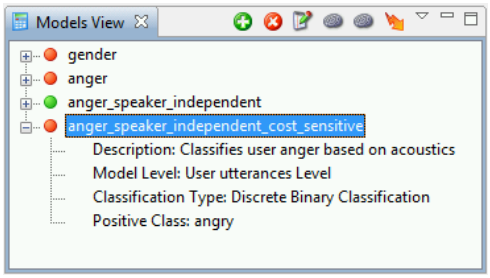


Figure 3: Definition of a model within Witchcraft. External recognizers have to deliver predictions for the defined models as XML documents.

A recognizer implemented in an ML framework has to be defined in such a way that it delivers XML documents that fit the model definition in Witchcraft. Each XML document represents the prediction of the recognizer for a specific dialogue turn of a specific dialogue. It contains for discriminative classification tasks, such as gender, or emotion the number of the turn that has been classified,

[3]http://cslu.cse.ogi.edu/DialogueView/
[4]www.rapid-i.net

the actual class label and the confidence scores of the classifier.

```
<xml>
<turn>
<number>1</number>
<label>anger</label>
<prediction>non-anger</prediction>
<confidence class='anger'>0.08</confidence>
<confidence class='no-ang'>0.92</confidence>
</turn>
</xml>
```

In regression tasks, such as the prediction of user satisfaction, retrieving cooperativeness scores etc., the returned result contains the turn number, the actual label and the prediction of the classifier:

```
<xml>
<turn>
<number>1</number>
<label>5</label>
<prediction>3.4</prediction>
</turn>
</xml>
```

After performing recognition on a number of dialogues with the recognizer Witchcraft reads in the XML files and creates statistics based on the predictions and calculates dialogue-wise *accuracy, f-score, precision and recall values, root mean squared error etc.* The values give some indication of how precisely the classifier worked on dialogue level. That followed it allows to search for dialogues with a low overall prediction accuracy, or e.g. dialogues with high true positive rates, high or low class-wise f-scores etc. via SQL. Now a detailed analysis of the recognizer's performance on dialogue level and possible reasons for the failure can be spotted.

4 Evaluating Models

In Figure 4 we see prediction series of two recognizers that have been applied on a specific dialogue: a gender recognizer that predicts the gender on turn basis and an emotion recognizer that predicts the user's emotional state (angry vs. non-angry) at the current turn. The red line symbolizes the confidence of the recognizers for each of the predicted classes. For example, in the emotion model the blue line is the confidence for a non-angry utterance (0-100%), the red line for an angry one. Exemplary for the two models we take a closer look at the gender model. It predicts the gender on turn basis, i.e. it takes the current speech sample and delivers estimations on the speaker's gender. As we can see, there are a number of misrecognitions in this call. It stems from a female speaker but the recognizer frequently estimated a male speaker. The call could be spotted by searching within Witchcraft for calls that yield a low accuracy for gender. It turned out that the misrecognized turns originate from the fact that the user performed off-talk with other persons in the background which caused the misrecognition. This finding suggests training the gender recognizer with non-speech and cross-talk samples in order to broaden the recognition from two (male, female) to three (male, female, non-speech) classes. Further it appears sensitive, to create a recognizer that would base its recognition on several speech samples instead of one, which would deliver a more robust result.

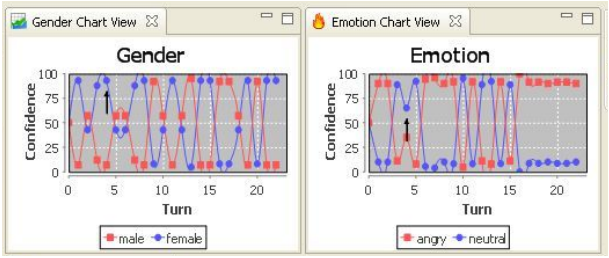


Figure 4: Screenshot of charts in Witchcraft based on turn-wise predictions an anger and a gender recognizer.

5 Portability towards other Corpora

Witchcraft has now been extended to cope with an unlimited number of corpora. An integration of new corpora is straight-forward. Witchcraft requires an SQL database containing two tables. The *dialogues* table hosts information on the overall dialogues (such as the dialogue ID, the category, filename of complete recording) and the *exchanges* table containing the turn-wise interactions (dialogue ID, turn number, system prompt, ASR parse, ASR confidence, semantic interpretation, hand transcription, utterance recording file, barged in, etc.). Both tables are linked through a $1 : n$ relationship, i.e. one entry in the dialogues table relates to n entries in the interactions table, cf. Figure 5. To demonstrate portability and in order to create a sample corpus that is deployed with Witchcraft, we included the CMU Let's Go bus information system from 2006 as demo corpus (Raux et al., 2006). It contains 328 dialogues including full recordings. The Witchcraft project includes a parser that allows to transform raw log data from the Let's Go system into the Witchcraft table structure.

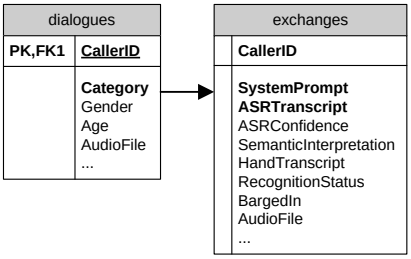


Figure 5: Dialogue and exchanges table with 1:n relationship. Bold database columns are required, others are optional.

6 Conclusion and Discussion

Witchcraft turned out to be a valuable framework in our everyday work when dealing with large dialogue corpora. At the current stage several students are working with it in multi-user mode to listen, analyze and annotate dialogues from three different corpora consisting of up to 100,000 dialogues each. Witchcraft allows them to search for dialogues relevant to the current task. The SQL-based access allows a powerful and standardized querying and retrieval of dialogues from the database. Witchcraft provides an overview and presents decisive information about the dialogue at one glance and allows to sort and group different types of dialogue for further research. Moreover, Witchcraft allows us to test arbitrary recognizers that provide additional information to the dialogue manager. Witchcraft tells us at which point in time a dialogue system would possess which knowledge. Further it allows us to conclude the reliability of this knowledge for further employment in the dialogue. For an evaluation of recognizers within Witchcraft please refer to (Schmitt et al., 2010b) where the deployment of an anger recognizer is simulated.

Witchcraft is now freely and publically available to the community. It is hosted under GNU General Public License at Sourceforge under witchcraftwb.sourceforge.org. The employed component architecture allows for the development of third-party plug-ins and components for Witchcraft without the need for getting into detail of the existing code. This facilitates the extension of the workbench by other developers. We hope that Witchcraft will help to foster research on future dialogue systems and we encourage the community to contribute.

Acknowledgements

The research leading to these results has received funding from the Transregional Collaborative Research Centre SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems” funded by the German Research Foundation (DFG). The authors would like to thank the CMU Let’s Go Lab from Carnegie Mellon University in Pittsburgh for their permission to deploy the Let’s Go Bus Information Corpus jointly with Witchcraft.

References

Niels Ole Bernsen, Laila Dybkjaer, and Mykola Kolodnytsky. 2002. The nite workbench - a tool for annotation of natural interactivity and multimodal data. In *Proc. of LREC*, pages 43–49, Las Palmas, Spain.

Tobias Bocklet, Andreas Maier, Josef Bauer, Felix Burkhardt, and Elmar Nöth. 2008. Age and gender recognition for telephone applications based on gmm supervectors and support vector machines. In *Proc. of ICASSP*, volume 1, pages 1605–1608.

Felix Burkhardt, Florian Metze, and Joachim Stegmann, 2007. *Speaker Classification for Next Generation Voice Dialog Systems*. Advances in Digital Speech Transmission. Wiley.

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, and Sebastian Möller. 2009. Modeling user satisfaction with hidden markov model. In *Proc. of SIGDIAL 2009*, pages 170–177.

Antoine Raux, Dan Bohus, Brian Langner, Alan W. Black, and Maxine Eskenazi. 2006. Doing research on a deployed spoken dialogue system: One year of lets go! experience. In *Proc. of Interspeech*, September.

Alexander Schmitt, Gregor Bertrand, Tobias Heinroth, and Jackson Liscombe. 2010a. Witchcraft: A workbench for intelligent exploration of human computer conversations. In *Proc. of LREC*, Valetta, Malta, May.

Alexander Schmitt, Tim Polzehl, and Wolfgang Minker. 2010b. Facing reality: Simulating deployment of anger recognition in ivr systems. In *Proc. of IWSDS*, September.

Marilyn Walker, I Langkilde-Geary, H W Hastie, J Wright, and A Gorin. 2002. Automatically training a problematic dialogue predictor for a spoken dialogue system. *Journal of Artificial Intelligence Research*, (16):293–319.

MPOWERS: a Multi Points Of VieW Evaluation Refinement Studio

Marianne Laurent, Philippe Bretier
Orange Labs
Lannion, France
{marianne.laurent, philippe.bretier}@orange-ftgroup.com

Abstract

We present our Multi Point Of vieW Evaluation Refinement Studio (MPOWERS), an application framework for Spoken Dialogue System evaluation that implements design conventions in a user-friendly interface. It ensures that all evaluator-users manipulate a unique shared corpus of data with a shared set of parameters to design and retrieve their evaluations. It therefore answers both the need for convergence among the evaluation practices and the consideration of several analytical points of view addressed by the evaluators involved in Spoken Dialogue System projects. After introducing the system architecture, we argue the solution's added value in supporting a both data-driven and goal-driven process. We conclude with future works and perspectives of improvement upheld by human processes.

1 Introduction

The evaluation of Spoken Dialogue Systems (SDS) is a twofold issue. On the one hand, the lack of convention on evaluation criteria and the many different evaluation needs and situations along with SDS projects lead to nomadic evaluation setups and interpretations. We inventoried seven job families contributing to these projects: the marketing people, the business managers, the technical and ergonomics experts, the hosting providers, the contracting owners as well as the actual human operators which integrate SDS in their activity (Laurent et al., 2010). Various experimental protocols for data collection and analytical data processing flourish in the domain. On the other hand, however they may not share evaluation needs and methods, the various potential evaluators need to cooperate inside and across projects. This claims for a convergence of evaluation practices toward standardized methodologies. The domain has put a lot of efforts toward the definition of commensurable metrics (Paek, 2007) for comparative evaluations and improved transparency over communications on systems' performances.

Nonetheless, we believe that no one-size-fits-all solution may cover all evaluation needs (Laurent and Bretier, 2010). We therefore work onto the *rationalization* - not the standardization - of evaluation practices. By rationalization, we refer to the definition of common norms to describe the evaluation protocols; common thinking models and vocabulary, for evaluators to make their procedures explicit. Our *Multi Points Of VieW Evaluation Refinement Studio* (MPOWERS) facilitates the design, from a unique corpus of parameters, of personalized evaluations adapted to the particular contexts. It does not compete with workbenches like MeMo (Möller et al., 2006) or WITcHCRafT (Schmitt et al., 2010) for which the overall evaluation process is predefined within the tool.

The following section details the solution architecture. Then, we present the MPOWERS's purposes, emphasizing on its added value for evaluators. Last, we explain the technical and process-related aspects that must support the system.

2 Architecture of the system

The application is built on a classical Business Intelligence (BI) solution that aims to provide decision makers with personalized information (See Fig. 1). We store, in a single datamart, parameters retrieved from heterogeneous sources: interaction logs, user questionnaires and third-party annotations relative to the evaluation campaigns arranged on the evaluated system(s). Then, data are cleaned, transformed and aggregated into Key Performance Indicators (KPIs). It guarantees that the indicators used across teams and projects are defined, calculated and maintained in the same place.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 265–268,
The University of Tokyo, September 24-25, 2010.

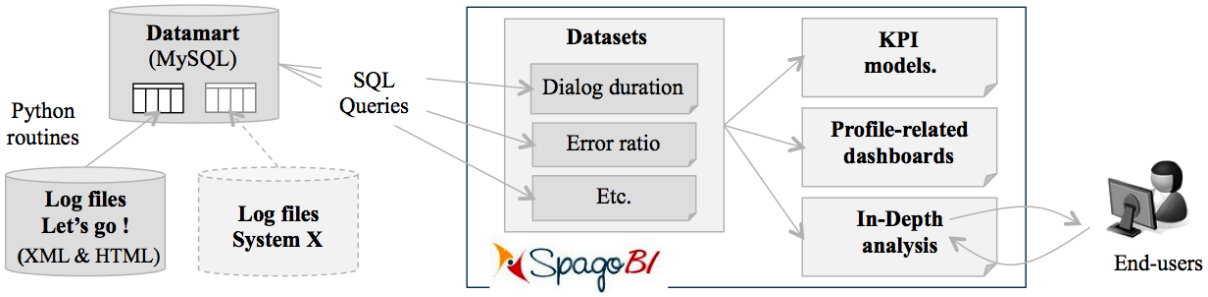


Figure 1: The MPOWERS architecture

On the upper layer, evaluators define and retrieve personalized reports and dashboards.

We use the Let's Go! System corpus shared by the Carnegie Mellon University. It contains log files generated since from 2003 from the Pittsburgh's telephone-based bus information system log files, one per module composing the system, and a summary HTML file. At our stage of the project the html summary allows the calculation of a satisfying number of parameters to support the system development and refinement. We compute the dialogue duration, the number of system and user turns, the number of barge-ins, the ratio between user and system turn number, the number of help requests and of no-matches per call and the ratio of successful interactions.

The application relies on the SpagoBI 2.6 open source solution[1]. Once parametrized, it enables non-technical stakeholders to retrieve personalized KPIs reports based on shared resources. For now, it delivers basic dashboards for two user profiles. One focuses on the service monitoring for marketing people and business managers and the other one provides the development team with usability-related performance figures (see fig. 4). The unique datamart guarantees all users to work from similar data. Its population requires parsing routines to identify and extract the relevant data.

3 Evaluation process and added value

By automating tractable tasks, MPOWERS supports the evaluator-users in their evaluation process driven by decision-making objectives. As sketched in figure 2, our application-supported process is slightly modified from the one defined by Stufflebeam (1980): a process through which one defines, obtains and delivers useful pieces of information that enable to settle between the alternative possible decisions.

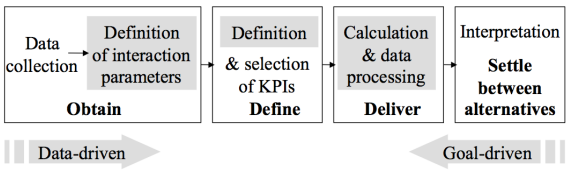


Figure 2: Evaluation process with MPOWERS (grey-tinted stages are supported by the system)

Custom-made Python[2] routines enable to extract relevant data from the log files. They provide CSV[3] formatted files to be converted into SQL scripts. The datamart is designed to be gradually populated from successive evaluation campaigns on one or several SDS. As data may originates from diverse sources, it arrays in different formats and often displays different parameters. Adapted ad hoc routines permit the manipulation into consistent format. We anticipate the use of separate tables in the datamart from comparative evaluations ons distinct systems.

The retrieval of KPIs in SpagoBI requires *datasets* pre-parametrized over SQL-Queries. They describe the SDS's performance and behaviour. We defined the parameters relative to the system performance according to the ITU-T Rec. P.Sup24 (2005). Yet, unless input corpora are defined accordingly not all the recommendation's parameters can be implemented. Three modes to display these datasets are proposed to evaluators:

- *A summary of high-level KPIs* provides a general view on the evaluated system with "red-light indicators" (see fig. 3). Links to more detailed charts or analysis tools are displayed next to each of them.

[1] http://www.spagoworld.org/

[2] http://www.python.org/

[3] Comma-Separated Value

Figure 3: High-level KPIs with link to more detailed documents. Please note that the success ratio is calculated via an ad-hoc query and does not necessarily corresponds to the user being or not satisfied.

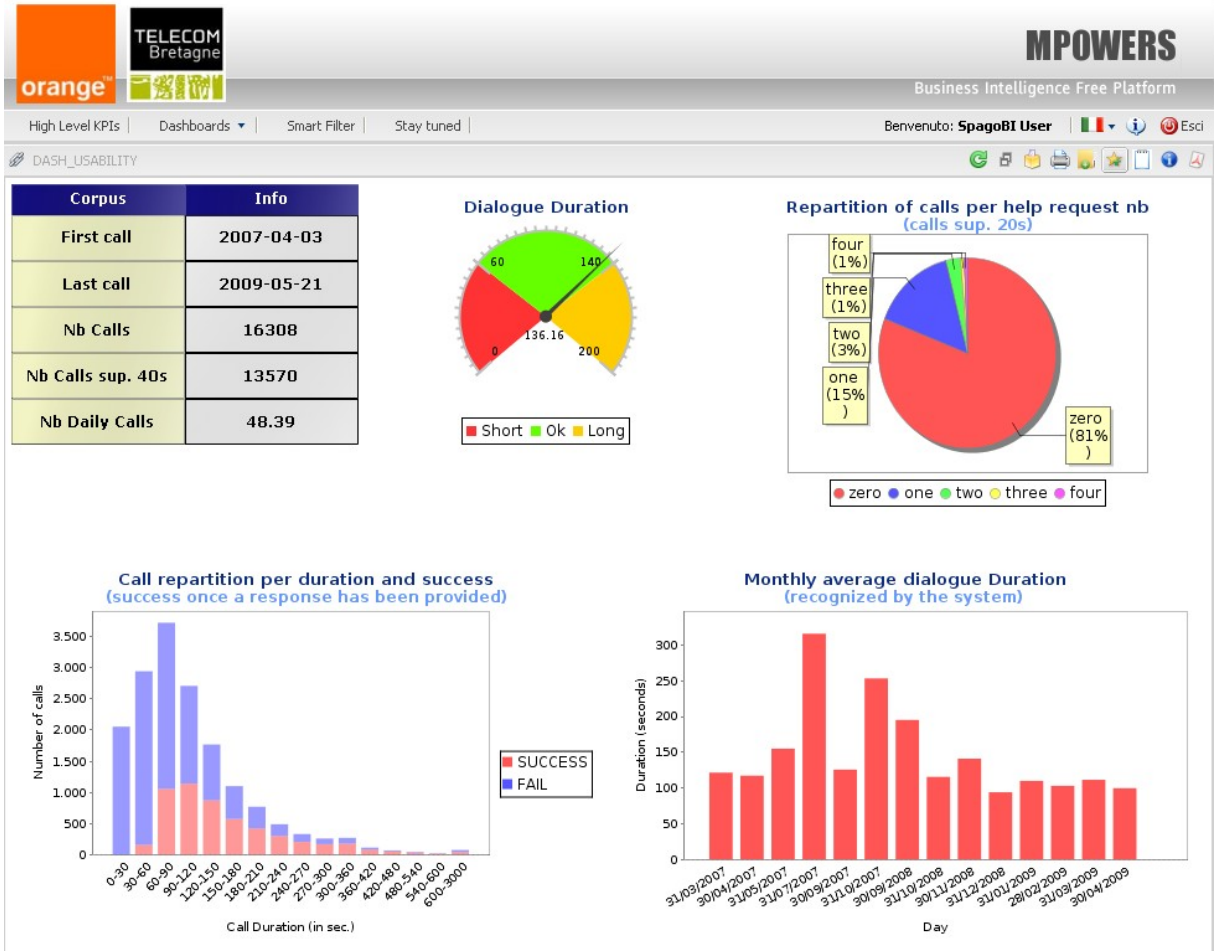


Figure 4: Dashboard dedicated to a high-level view on usability performance.

- *Visual dashboards* display pre-processed data according to pre-defined evaluation profiles (see fig. 4).
- *Tools for in-depth individual analysis* Filtered queries permit evaluators to individually adjust their analysis according to local evaluation objectives. Queries can be stored for later use or saved in PDF documents for distribution to non-MPOWERS users.

End-users, i.e. the evaluators, are limited to display the results and proceed to in-depth queries. An administrator access allows for prior data processing and the configuration of datasets, KPIs and dashboards. With collaborative enhancement purposes, the application supports communication between users with built-in discussion threads information feeds and shared to-do-lists to suggest and negotiate future configurations.

These distinct outlooks on the corpus are complementary. They combine a high-level view on the service's behaviour and performance with detailed personalised analysis. Whatever their layouts, every information displayed to the evaluators-users is retrieved from a unique corpus and from the same SQL-queries. Therefore, even if all evaluators consider distinct features on the evaluated service, our framework brings consistency to their evaluation practices.

4 Future work

MPOWERS is on its first development stages. Several perspectives of enhancement are planned. First, it requires to be augmented with more KPIs and in-depth analytical features. Second, as it only manipulates automated log files, user questionnaires and third-party annotations are expected to enrich its evaluation possibilities. Third, we intent MPOWERS to perform comparative evaluations between distinct services in the future. And last, the framework would benefit from being employed within real evaluators' daily activity.

5 Conclusion

The paper presents a platform that supports the SDS project stakeholders in their evaluation task. While advocating for a rationalization of evaluation practices among project teams and across organizations, it promotes the existence of different cohabiting points of view instead of disregarding them. When most evaluation contributions cover the overall evaluation process, from experimental data collection set-ups to guidance for interpretation, we limit to a user-centric framework, where evaluators remain in charge of the evaluation design. We actually provide them with an operational framework and unified tools to design and process their evaluations. This may help initiate individual, as well as community-wide, gradual refinements of methodologies.

Acknowledgments

The demo makes use of the *Let's Go!* log files provided by the Carnegie Mellon University. We thank Telecom Bretagne, Q. Jin, X. Chen, S. Zarrad, F. Agez and A. Bolze for their contribution in the platform deployment.

References

M. Eskenazi, A. W. Black, A. Raux, and B. Langner. 2008. Let's Go Lab: a platform for evaluation of spoken dialog systems with real world users. In *Interspeech 2008*, Brisbane, Australia.

M. Laurent and P. Bretier. 2010. Ad-hoc evaluations along the lifecycle of industrial spoken dialogue systems: heading to harmonisation? In *LREC 2010*, Malta.

M. Laurent, I. Kanellos, and P. Bretier. 2010. Considering the subjectivity to rationalise evaluation approaches: the example of Spoken Dialogue Systems. In *QoMEx'10*, Trondheim, Norway.

S. Möller, R. Englert, K.-P. Engelbrecht, V. Hafner, A. Jameson, A. Oulasvirta, A. Raake, and N. Reithinger. 2006. MeMo: towards automatic usability evaluation of spoken dialogue services by user error. *9th International Conference on Spoken Language*.

T. Paek. 2007. Toward evaluation that leads to best practices: reconciling dialog evaluation in research and industry. In *Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, pages 40–47, New York. ACL, Rochester.

ITU-T Rec. P.Sup24. 2005. Parameters describing the interaction with spoken dialogue systems.

A. Schmitt, G. Bertrand, T. Heinroth, W. Minker, and J. Liscombe. 2010. Witchcraft: A workbench for intelligent exploration of human computer conversations. In *LREC 2010*, Malta.

D. L. Stufflebeam. 1980. *L'évaluation en éducation et la prise de décision*. Ottawa.

Statistical Dialog Management Methodologies for Real Applications

David Griol
Dept. of Computer Science
Carlos III University of Madrid
Av. Universidad, 30, 28911, Leganés
dgriol@inf.uc3m.es

Zoraida Callejas, Ramón López-Cózar
Dept. of Languages and Computer Systems, CITIC-UGR
University of Granada
C/ Pdta. Daniel Saucedo Aranda, 18071, Granada
{zoraida, rlopezc}@ugr.es

Abstract

In this paper we present a proposal for the development of dialog systems that, on the one hand, takes into account the benefits of using standards like VoiceXML, whilst on the other, includes a statistical dialog module to avoid the effort of manually defining the dialog strategy. This module is trained using a labeled dialog corpus, and selects the next system response considering a classification process that takes into account the dialog history. Thus, system developers only need to define a set of VoiceXML files, each including a system prompt and the associated grammar to recognize the users responses to the prompt. We have applied this technique to develop a dialog system in VoiceXML that provides railway information in Spanish.

1 Introduction

When designing a spoken dialog system, developers need to specify the system actions in response to user utterances and environmental states that, for example, can be based on observed or inferred events or beliefs. In addition, the dialog manager needs a dialog strategy that defines the conversational behavior of the system. This is the fundamental task of dialog management (Paek and Pieraccini, 2008), as the performance of the system is highly dependent on the quality of this strategy. Thus, a great effort is employed to empirically design dialog strategies for commercial systems. In fact, the design of a good strategy is far from being a trivial task since there is no clear definition of what constitutes a good strategy (Schatzmann et al., 2006). Once the strategy has been designed, the implementation of the system is leveraged by programming languages such as VoiceXML, for which different programming environments and tools have been created to help developers.

As an alternative of the previously described rule-based approaches, the application of statistical approaches to dialog management makes it possible to consider a wider space of dialog strategies (Georgila et al., 2006; Williams and Young, 2007; Griol et al., 2009). The main reason is that statistical models can be trained from real dialogs, modeling the variability in user behaviors. The final objective is to develop dialog systems that have a more robust behavior and are easier to adapt to different user profiles or tasks.

(Pieraccini et al., 2009) highlights the impracticality of applying statistical learning approaches to develop commercial applications, in the sense that it is difficult to consider the expert knowledge of human designers. From his perspective, a hybrid approach, combining statistical and rule-based approaches, could be a good solution. The reason is that statistical approaches can offer a wider range of alternatives at each dialog state, whereas rule based approaches may offer knowledge on best practices.

For example, (Williams, 2008) proposes taking advantage of POMDPs and rule-based approaches by using POMDPs to foster robustness and at the same time being able to incorporate handcrafted constraints which cover expert knowledge in the application domain. Also (Lee et al., 2010) have recently proposed a different hybrid approach to dialog modeling in which n-best recognition hypotheses are weighted using a mixture of expert knowledge and data-driven measures by using an agenda and an example-based machine translation approach respectively. In both approaches, the hybrid method achieved significant improvements.

Additionally, speech recognition grammars for commercial systems have been usually built on the basis of handcrafted rules that are tested recursively, which in complex applications is very costly (McTear, 2004). However, as stated by (Pieraccini et al., 2009), many sophisticated com-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 269–272,
The University of Tokyo, September 24-25, 2010.

mercial systems already available receive a large volume of interactions. Therefore, industry is becoming more interested in substituting rule based grammars with statistical approaches based on the large amounts of data available.

As an attempt to improve the current technology, we propose to merge statistical approaches with VoiceXML. Our goal is to combine the flexibility of statistical dialog management with the facilities that VoiceXML offers, which would help to introduce statistical approaches for the development of commercial (and not strictly academic) dialog systems. To this end, our technique employs a statistical dialog manager that takes into account the history of the dialog up to the current dialog state in order to decide the next system prompt. In addition, the system prompts and the grammars for ASR are implemented in VoiceXML-compliant formats, for example, JSGF or SRGS. As it is often difficult to find or gather a human-machine corpus which cover an identical domain as the system which is to be implemented, our approach is also based on the compilation of corpora of interactions of simulated users, which is a common practice when using machine learning approaches for system development.

In contrast with other hybrid approaches, our main aim is not to incorporate knowledge about best strategies in statistical dialog management, but rather to take advantage of an implementation language which has been traditionally used to build rule-based systems (such as VoiceXML), for the development of statistical dialog strategies. Expert knowledge about deployment of VoiceXML applications, development environments and tools can still be exploited using our technique. The only change is in the transition between states, which is carried out on a data-driven basis (i.e., is not deterministic). We have applied our technique to develop a dialog system that provides railway information, for which we have developed a statistical dialog management technique in a previous study.

2 Our Proposal to Introduce Statistical Methodologies in Commercial Applications

As stated in the introduction, our approach to integrate statistical methodologies in commercial applications is based on the automatic learning of the dialog strategy using a statistical dialog management methodology. In most dialog systems, the dialog manager makes decisions based only on the information provided by the user in the previous turns and its own dialog model. For example, this is the case with most dialog systems for slot-filling tasks. The methodology that we propose for the selection of the next system response for this kind of task is detailed in (Griol et al., 2008). It is based on the definition of a data structure that we call Dialog Register (DR), which contains the information provided by the user throughout the dialog history. In brief, it is as follows: for each time i, the selection of the next system prompt A_i is carried out by means of the following maximization:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | DR_{i-1}, S_{i-1})$$

where the set $\mathcal{A}$ contains all the possible system responses and S_{i-1} is the state of the dialog sequence (*system-turn, user-turn*) at time i.

Each user turn supplies the system with information about the task; that is, he/she asks for a specific concept and/or provides specific values for certain attributes. However, a user turn could also provide other kinds of information, such as task-independent information. This is the case of turns corresponding to *Affirmation*, *Negation* and *Not-Understood* dialog acts. This kind of information implies some decisions which are different from simply updating the DR_{i-1}. Hence, for the selection of the best system response A_i, we take into account the DR that results from turn 1 to turn $i-1$, and we explicitly consider the last state S_{i-1}. Our model can be extended by incorporating additional information to the DR, such as some chronological information (e.g. number of turns up to the current turn) or user profiles (e.g. user experience or preferences).

The selection of the system response is carried out through a classification process, for which a multilayer perceptron (MLP) is used. The input layer receives the codification of the pair (DR_{i-1}, S_{i-1}). The output generated by the MLP can be seen as the probability of selecting each of the different system answers defined for a specific task.

To learn the dialog model we use dialog simulation techniques. Our approach for acquiring a dialog corpus is based on the interaction of a user simulator and a dialog manager simulator (Griol et al., 2007). The user simulation replaces the user intention level, that is, it provides concepts and

attributes that represent the intention of the user. This way, the user simulator carries out the functions of the ASR and NLU modules. Errors and confidence scores are simulated by a specific module in the simulator. The acquired dialogs are employed to automatically generate VoiceXML code for each system prompt and create the grammar needed to recognize the possible user utterances after each one of the system prompts.

3 Development of a railway information system using the proposed technique

To test our proposal, we have used the definitions taken to develop the DIHANA dialog system, which was developed in a previous study to provide information about train services, schedules and fares in Spanish (Griol et al., 2009; Griol et al., 2008). The DR defined for the our railway information system is a sequence of 15 fields, corresponding to the five concepts (*Hour, Price, Train-Type, Trip-Time, Services*) and ten attributes (*Origin, Destination, Departure-Date, Arrival-Date, Departure-Hour, Arrival-Hour, Class, Train-Type, Order-Number, Services*). The system generates a total of 51 different prompts.

Three levels of labeling are defined for the labeling of the system dialog acts. The first level describes general acts which are task independent. The second level is used to represent concepts and attributes involved in dialog turns that are task-dependent. The third level represents values of attributes given in the turns. The following labels are defined for the first level: *Opening*, *Closing*, *Undefined*, *Not-Understood*, *Waiting*, *New-Query*, *Acceptance*, *Rejection*, *Question*, *Confirmation*, and *Answer*. The labels defined for the second and third level were the following: *Departure-Hour*, *Arrival-Hour*, *Price*, *Train-Type, Origin*, *Destination*, *Date*, *Order-Number*, *Number-Trains*, *Services*, *Class*, *Trip-Type*, *Trip-Time*, and *Nil*. There are dialog turns which are labeled with several dialog acts.

Having this kind of labeling and the values of attributes obtained during a dialog, it is straightforward to construct a sentence in natural language. Some examples of the dialog act labeling of the system turns are shown in Figure 1.

Two million dialogs were simulated using a set of two types of scenarios. Type S1 defines one objective for the dialog, whereas Type S2 defines two. Table 1 summarizes the statistics of the acquisition for the two types of scenarios.

[SPANISH] Bienvenido al servicio de información de trenes. ¿En qué puedo ayudarle? [ENGLISH] *Welcome to the railway information system. How can I help you?* (*Opening:Nil:Nil*)
[SPANISH] El único tren es un Euromed que sale a las 0:27. ¿Desea algo más? [ENGLISH] *There is only one train, which is a Euromed, that leaves at 0:27. Anything else?* (*Answer:Departure-Hour:Departure-Hour:Departure-Hour[0.27],Number-Trains[1],Train-Type[Euromed]*) (*New-Query:Nil:Nil*)

Figure 1: Labeling examples of system turns from the DIHANA corpus

	Type S1	**Type S2**
Simulated dialogs	10^6	10^6
Successful dialogs	15,383	1,010
Different dialogs	14,921	998
Number of user turns per dialog	4.9	6.2

Table 1: Statistics of the new corpus acquisition

The 51 different system prompts have been automatically generated in VoiceXML using the proposed technique. For example, Figure 2 shows the VXML document to prompt the user for the origin city, whereas Figure 3 shows the obtained grammar for ASR.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
   xmlns:xsi="http://www.w3.org/2001/
   XMLSchema-instance"
   xsi:schemaLocation="http://www.w3.org/2001/vxml
   http://www.w3.org/TR/voicexml20/vxml.xsd"
   version="2.0" application="app-dihana.vxml">
 <form id="origin_form">
  <field name="origin">
     <grammar type="application/srgs+xml"
           src="/grammars/origin.grxml"/>
     <prompt>Tell me the origin city.</prompt>
     <filled>
       <return namelist="origin"/>
     </filled>
  </field>
 </form>
</vxml>
```

Figure 2: VXML document to require the origin city

4 Conclusions

In this paper, we have described a technique for developing dialog systems using a well known

```
#JSGF V1.0;
grammar origin;
public <origin> = [<desire>]
[<travel> <city> {this.destination=$city}]
[<proceed> <city> {this.origin=$city}];
<desire> = I want [to know] | I would like
[to know] | I would like | I want | I need
| I have to;
<travel> = go to | travel to | to go to
| to travel to;
<city> = Jaén | Córdoba | Sevilla | Huelva |
Cádiz | Málaga | Granada | Almería |
Valencia | Alicante | Castellón | Barcelona
| Madrid;
<proceed> = from | going from | go from;
```

Figure 3: Grammar defined to capture the origin city

standard like VoiceXML, and considering a statistical dialog model that is automatically learnt from a dialog corpus.

The main objective of our work is to reduce the gap between academic and commercial systems by reducing the effort required to define optimal dialog strategies and implement the system. Our proposal works on the benefits of statistical methods for dialog management and VoiceXML, respectively. The former provide an efficient means to exploring a wider range of dialog strategies, whereas the latter makes it possible to benefit from the advantages of using the different tools and platforms that are already available to simplify system development. We have applied our technique to develop a dialog system that provides railway information, and have shown that it enables creating automatically VoiceXML documents to prompt the user for data, as well as the necessary grammars for ASR. As a future work, we plan to study ways for adapting the proposed dialog management technique to more complex domains.

Additionally, we are interested in investigating possible ways for easing the adoption of our technique in industry, and the main challenges that might arise in using it to develop commercial systems.

Acknowledgments

This research has been funded by the Spanish Ministry of Science and Technology, under project HADA TIN2007-64718.

References

K. Georgila, J. Henderson, and O. Lemon. 2006. User Simulation for Spoken Dialogue Systems: Learning and Evaluation. In *Proc. of the 9th Interspeech/ICSLP*, pages 1065–1068, Pittsburgh (USA).

D. Griol, L.F. Hurtado, E. Sanchis, and E. Segarra. 2007. Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 39–42, Antwerp (Belgium).

D. Griol, L.F. Hurtado, E. Segarra, and E. Sanchis. 2008. A Statistical Approach to Spoken Dialog Systems Design and Evaluation. *Speech Communication*, 50(8–9):666–682.

D. Griol, G. Riccardi, and Emilio Sanchis. 2009. A Statistical Dialog Manager for the LUNA project. In *Proc. of Interspeech/ICSLP'09*, pages 272–275, Brighton (UK).

Cheongjae Lee, Sangkeun Jung, Kyungduk Kim, and Gary Geunbae Lee. 2010. Hybrid approach to robust dialog management using agenda and dialog examples. *Computer Speech and Language*, 24(4):609–631.

Michael F. McTear, 2004. *Spoken Dialogue Technology: Towards the Conversational User Interface*. Springer.

T. Paek and R. Pieraccini. 2008. Automating spoken dialogue management design using machine learning: An industry perspective . *Speech Communication*, 50(8–9):716–729.

Roberto Pieraccini, David Suendermann, Krishna Dayanidhi, and Jackson Liscombe. 2009. Are We There Yet? Research in Commercial Spoken Dialog Systems. *Lecture Notes in Computer Science*, 5729:3–13.

J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. In *Knowledge Engineering Review*, volume 21(2), pages 97–126.

J. Williams and S. Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. In *Computer Speech and Language*, volume 21(2), pages 393–422.

Jason D. Williams. 2008. The best of both worlds: Unifying conventional dialog systems and POMDPs. In *Proceedings of the International Conference on Spoken Language Processing*.

YouBot: A Simple Framework for Building Virtual Networking Agents

Seiji Takegata, Kumiko Tanaka-Ishii
Graduate School of Information Science and Technology, University of Tokyo
13F Akihabara Daibiru, 1-18-13 SotoKanda Chiyoda-ku, Tokyo, Japan
takegata@cl.ci.i.u-tokyo.ac.jp, kumiko@i.u-tokyo.ac.jp

Abstract

This paper proposes a simple framework for building 'virtual networking agents'; programs that can communicate with users and collect information through the internet. These virtual agents can also communicate with each other to share information that one agent does not have. The framework - 'YouBot' - provides basic functions such as protocol handling, authentication, and data storage. The behavior of the virtual agents is defined by a task processor ('TP') which can be written in a light-weight language such as JavaScript. It is very easy to add new functions to a virtual agent. The last part of this paper discusses the micro-blog system 'twitter' and other web services as information sources that a virtual agent can utilise to make its behavior more suited to the user.

1 Introduction

Recently, communicating in short sentences, such as via Instant Messenger or SMS, has become more common; the use of 'Twitter', especially, is spreading very quickly and widely. These networking tools are not only for chatting, but also for gathering information on and discussing a world of topics. Short sentences are suitable for Natural Language Interface processes like question-answering, recommendation, or reservation systems; thus, Natural Language Interfaces are becoming increasingly important in this area of communications.

There are many dialogue systems that process natural language as a user-input, like 'UC' (Wilensky 1987), 'tour guide' (Prodanov et.al. 2002), but most of them are designed for a specific individual purpose, so, have to locate different systems for different purposes. This problem has been one of the main barriers preventing dialogue systems from being adopted more widely.

Our framework -'YouBots'- can accept the user's messages as input, and respond in natural language. The behavior of these agents is defined by task processors ('TPs') which can be written in a light-weight language, eg. JavaScript. It is very easy to add new TPs to a virtual agent. Web-browsers like Firefox have a similar add-on mechanism and, through open-source collaboration, now have thousands of types of extension. We hope that, in the same way, developers will be encouraged to write new TPs for our YouBot framework.

Personal Digital Assistant is an example of this kind of application. Its schedule manager, contact manager and to-do list are easily implemented on this framework. Q&A system is another example; it would be realized by cooperating with webservice or other external system.

The framework also has a unique networking feature to help the bots communicate with each other: It is called 'Inter-bot communication', a feature which expands the ways in which the virtual agent can get preferred information for the user.

2 Outline of the Networking Bot

Most existing dialogue systems only use their internal data. So their application is often limited to a specific purpose, as in domain-specific expert systems. Using a network feature enabling bots to communicate with each other, our system can obtain many types of information from other, external systems. Figure 1 shows users communicating with their own bots, and bots communicating with each other to collect information for their users. Each connection in the figure is conducted by XMPP protocol[1].

[1] http://www.xmpp.org/

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 273–276,
The University of Tokyo, September 24-25, 2010.

Information in each bot can be linked in the same manner as web pages, and combine to form semantic structures in the way of the Semantic Web (Berners-Lee 2001), this can improve the bots behaviour.

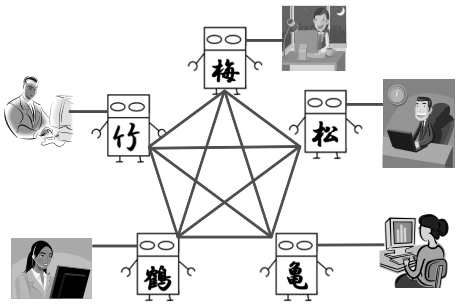


Figure 1: Network of Users and Bots.

If TPs are designed to share information through the network, a user need not know which system contains the information he or she needs. They only need to talk to their own personal bot, then the bot will find the information for them. Each user has their own bot, and can share information through these bots. The modes of interaction with other users and modes of information gathering depend on how the TPs are written.

3 Task Managing

Within our framework, a 'task manager' invokes a 'task processor' as shown in Figure 2:-

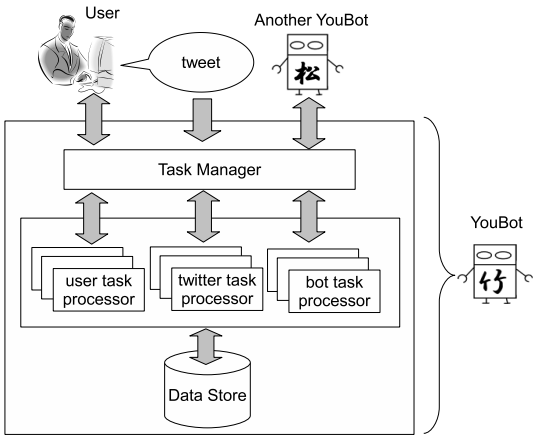


Figure 2: Task-managing

There are existing systems that process tasks with modular components - TPs; among these, we find two approaches, one is centralized and the other distributed. In the centralized approach, a user-message is analyzed by a central component of the system, often called the 'dialogue manager'. Then the dialogue manager decides which TP to invoke. The 'Smart Personal Assistant' (Nguyen et al. 2005) uses 'BDI Theory' (Bratman 1987) to determine the user's intention in the dialogue manager. Then, a TP which satisfies the user's demand can be selected. In this approach, interpretation can be carried out efficiently, but the task manager needs to be revised every time a new TP is added. This is not an easy operation unless the task manager is configured to recognize the functionality of a new TP automatically. This may be viewed as a serious weakness of the centralized approach.

On the other hand, there is 'RIME' (Nakano et al. 2008) which adopts a distributed approach - where the user-message is sent to each of the TPs, which interpret it and return a 'score' indicating how well they can handle the message. Consequently, the TP returning the highest score will process the user's message. This approach suffers from the inefficiency of having to interpret the user's message many times in each TP. On the positive side, there is no need to revise or redesign central components when a new TP is added.

We have decided to adopt the distributed approach because we think expandability is more important than speed. Our framework uses 'Scripting Engine' in which JavaScript codes can run. JavaScript is very easy to write, owing to which, many people write extensions for Firefox in which JavaScript codes can also run. How simply a TP can be written is a very important factor in the attraction of developers.

4 How to Write Task Processors

There are three types of designated TP in the YouBot system: a 'user task processor', a 'bot task processor', and a 'twitter task processor'. The 'user-TP' is for processing messages from the user - explained in the 'Basic Task Processor' subsection (see below); the 'bot-TP' is for processing inquiries from other bots - explained in the 'Inter-bot communication' subsection (see below); and the 'twitter-TP' reads the user's tweets at the Twitter site - explained in the 'Cooperation with External Services' subsection (also see below). Each TP is saved to an individual JavaScript file in the 'task' folder with a .js extension. The YouBot Framework reads these files when the program starts and when a 'reload' command is issued.

4.1 Basic Task Processors

The JavaScript code for a basic TP needs at least one variable and two functions. The variable 'type' indicates the type of task - which can either be a user-task, bot-task, or twitter-task. The mandatory functions are 'estimate' and 'process', an approach introduced in the 'Blackboard' multi-agent system (Corkill 1991). The 'estimate' function receives a user-message from the task manager and returns its score, which shows how likely it is that this TP will be the best among the other TPs to process the message. For example, when a TP uses pattern-matching for message interpretion, the score may be higher if the matched pattern is more complicated, or may be zero if no pattern matched the user message. The 'estimate' function can use not only pattern-matching, but also various data calculated or stored in different ways; such as the dialogue history or information from external systems. The YouBot Framework gathers and compares the scores returned from the TPs, then selects the processor which returned the highest score to process the message. The 'process' function of the TP handles the user-message and makes a response to the user. During the processing, this function can access the internal data store or an external system to get or save various information.

4.2 Pattern Matching

Our framework provides a handy way to do pattern-matching, using four types of placeholder:

An OR conditional placeholder is defined by "{abc |def }" format.

```
I {will go|went }to school.
```

matches both "I will go to school." and "I went to school." Optional selection can be defined with this "(abc|def)" format.

```
Yes (I do |it is).
```

matches "Yes I do", "Yes it is" and just "Yes" Using "[abc]" format, the content of the placeholder can be retrieved. For example, the pattern:

```
I went to [place].
```

matches the sentence "I went to school." or "I went to see a doctor." If the pattern matches the user's message, an object holding the contents of the placeholder will be returned. You can get the contents with the "get" function, specifying the placeholder - in this case "[place]"

To define a placeholder which matches only one specified pattern, "<abc>" format is used. For example, the placeholder "<date>" can be defined so that it matches a date expression such as 'yesterday' or 'on Sunday'. Then the pattern:

```
I went to [place] <date>.
```

matches "I went to school on Sunday", but does not match "I went to school with my brother". The content of <date> placeholder can also be retrieved with "get" function. Retrieved data can be kept in the data store and used in interaction with the user later.

4.3 The Data store

Many chatter bots don't remember what they have said before. 'A.L.I.C.E' (Wallace 2008) has a short memory - just one single interaction. Unusually, YouBot has a long-term data store for its memory. It holds key=value style properties which can be defined by the TP. To save schedule data, as in:-

```
type="schedule"
date="2010/05/14"
item="Submission dead-line"
```

- we create a new data object, set its properties, and use the 'save' function. To retrieve specific data from the data store, a 'data selector' object is provided. If the following condition is set up in the data selector:-

```
type="schedule"
date="2010/05/09"
```

- then a list of matching data is retrieved from the data store. The Youbot framework also provides a facility for responding to inquiries from other bots, and this raises security issues. In this framework, a default security filter is installed in the data selector to send information only to privileged bots. Data objects saved in the data store have security attributes for which the default is 'secret', and only the owner of the bot can access this information. This attribute can be set to 'private' or 'official' - then, the information will only be accessible to the bots which have 'private' or 'official' privilege. Developers do not have to worry about this data security setting during inter-bot communication.

4.4 Inter-bot communication

A user-TP can send an inquiry to another bot - about, for example, the user's schedule or knowledge and expertise. The TP generates an 'Inquiry Sender' object , sets the inquiry and the target bot's address, then uses the 'send' function. This

inquiry is formatted as an inter-bot message so that the receiving bot can distinguish it from user-messages. The receiving bot generates an 'Inquiry Responder' object for each of the incoming inter-bot messages; then the Task Manager sends the messages to the bot-TP. Next, the bot-TPs estimate the likelihood of processing the message and return scores - with the bot-TP which returns the highest score being selected to respond. A responding message is sent back to the inquiring bot in the inter-bot message format. then a function named 'convey' - within the inquiring TP - is invoked to make a response to the user. A function named 'timeout' is invoked when no response has been returned.

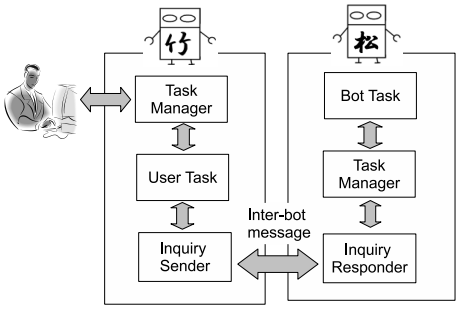


Figure 3: Inter-Bot Communication

To respond to an inquiry from another bot, a bot-TP for that inquiry has to be defined. Besides which, remote bots have to be given privilege to collect information which has a security attribute restricting access. If a TP developer fails to specify a security attribute for the data, no access will be allowed without the right privilege, because the default setting is secret.

4.5 Cooperation with External Systems

A bot can read the user's tweets at the twitter site at specified intervals. The User's tweets are sent to twitter-TPs, then estimated and processed in a same manner as user-TPs and bot-TPs. A bot can get information about a user's status, interests, and favorites; these data are useful for generating preferable responses for the user.

The Youbot framework also provides a utility function which takes URI and retrieves HTML code. This function can be used to access search engines or news sites. Services such as online shopping or recommendation engines represent the type of business model that would be suited to the application of the Youbot framework.

5 Interaction Example

The following are examples of interactions which YouBot might handle:

```
USER: I will meet John at 9 tomorrow.
SYSTEM: Is that A.M or P.M?
USER: pm
SYSTEM: There's a meeting with Mr. Smith at 8pm.
USER: It's been canceled.
SYSTEM: I see.
```

6 Conclusion

We proposed a simple framework for virtual agents. Its functionality can be easily extended by adding task processing modules written in JavaScript. The Youbot framework provides utility objects which make task processing even easier. Networking ability is also provided to expand the networked information's reach, while data security is maintained. Future work will include normalizing the estimation score. Another challenge is how best to share contextual information among TPs so they can interact to generate better responses for the user.

References

Robert Wilensky. Ther Berkley UNIX Consultant Project. *Informatik-Fachberichte*, volume 155, pages 286–296, Springer, 1987.

P. J. Prodanov, A. Drygajlo, G. Ramel, M. Meisser, and R. Siegwart. Voice enabled interface for interactive tour-guided robots. *In Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1332–1337, 2002.

T. Berners-Lee and J. Hendler and O. Lassila. The Semantic Web. *In Scientific American*, pages 34–43, May 2001.

R.S. Wallace. The Anatomy of A.L.I.C.E. *Parsing the Turing Test*, pages 181–210, Springer Netherlands, 2008.

A. Nguyen. An agent-based approach to dialogue management in personal assistants. *In Proceedings of IUI-2005*, pages 137–144. ACM Press, 2005.

M. Bratman. Intentions, Plans, and Practical Reason. Harvard University Press, 1987.

M. Nakano, K. Funakoshi, Y. Hasegawa, and H. Tsujino. A Framework for Building Conversational Agents Based on a Multi-Expert Model. *In Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 88–91. ACL, 2008.

Daniel D. Corkill. Blackboard systems. *AI Expert*, volume 6, pages 40–47, 2008.

‘How was your day?’ An affective companion ECA prototype

Marc Cavazza
School of Computing
Teesside University
Middlesbrough TS1 3BA
m.o.cavazza@tees.ac.uk

Raúl Santos de la Cámara
Telefónica I+D
C/ Emilio Vargas 6
28043 Madrid
e.rsai@tid.es

Markku Turunen
University of Tampere
Kanslerinrinne 1
FI-33014
mturunen@cs.uta.fi

José Relaño Gil
Telefónica I+D
C/ Emilio Vargas 6
28043 Madrid
joserg@tid.es

Jaakko Hakulinen
University of Tampere
Kanslerinrinne 1
FI-33014
jh@cs.uta.fi

Nigel Crook
Oxford University
Computing Laboratory
Oxford OX1 3QD
nigc@comlab.ox.ac.uk

Debora Field
Computer Science
Sheffield University
Sheffield S1 4DP
d.field@shef.ac.uk

Abstract

This paper presents a dialogue system in the form of an ECA that acts as a sociable and emotionally intelligent companion for the user. The system dialogue is not task-driven but is social conversation in which the user talks about his/her day at the office. During conversations the system monitors the emotional state of the user and uses that information to inform its dialogue turns. The system is able to respond to spoken interruptions by the user, for example, the user can interrupt to correct the system. The system is already fully implemented and aspects of actual output will be used to illustrate.

1 Introduction

Historically, Embodied Conversational Agents (ECAs) have been used in research and industry make information and complex tasks more accessible to customers and users. With the rise of new technologies in affective dialogue systems, we are beginning to see a future in which ECA dialogues are not all task-driven, but some will be focused on the social aspects of conversation. We envisage the development of ECAs that enhance the social quality of life of the user, and that build deep relationships with their users over years of use. Our affective companion ECA is our first contribution to this emerging technology.

Our emotionally intelligent dialogue system requires a high level of understanding of the meaning of the user turns, and so is quite unlike a ‘chatbot’. Affective aspects of the situation are also key in influencing the course of the conversation. The system attempts to balance both semantic and affective understanding in order to create an engaging and enjoyable companion for social conversation.

2 Scenario description

The system conversation is free-ranging and follows the user’s initiative, rather than being task-driven. We have for the time being restricted the range of conversation topics to things that happen in the life of someone who works in an office (the ‘How was your day?’ (HWYD) scenario). The user ‘offloads’ about his/her day at work by chatting to the ECA, and the ECA tries to empathise with and advise the user. The ECA asks questions to elicit information from the user, makes empathetic and sympathetic comments, and offers advice.

The system is able to process long user turns, as well as to generate long system turns (around 50 words). If the user perhaps dislikes what he/she hears, he/she can interrupt the system by barging in, and the ECA will respond in a human-like way. The conversation style is therefore considerably different from traditional task-based dialogues that consist of short dialogue turns and in which the system takes all the initiative.

The dialogue in Table 1 (actual output from a user session) illustrates some of the significant features of the system, including long system

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 277–280,
The University of Tokyo, September 24-25, 2010.

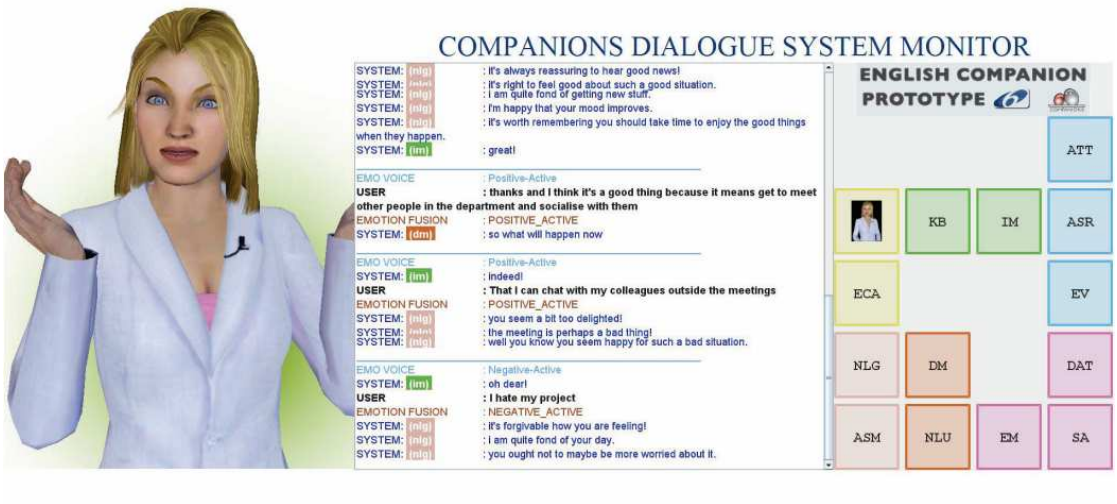


Figure 1: Screenshot of the prototype interface

turns, user interruptions and emotional analysis of user input.

User$_1$	Worst day ever. I arrived late because of the traffic.
System$_1$	You have my sympathy. What happened next?
User$_2$	I missed the meeting on project responsibilities; got more work dumped on me. Just what I don't need. I'm already likely to miss the deadline. [*ECA looks worried when user's negative tone is detected*]
System$_2$	Sorry to hear that. Who was at the meeting?
User$_3$	Only my team. They gave me a set of PowerPoint slides from the meeting. I spent the rest of the day reading them but my computer kept crashing because of updates.
System$_3$	Oh dear! It is understandable how you must be feeling. It's right to be unhappy about such a bad situation. Try not to worry about missing the deadline because of the technical problems. [*At this point user interrupts ECA by beginning turn User*$_4$] It's sad that your motivation could be affected by them. [*System stops speaking*]
User$_4$	No, I'm not worried about the technical problems, I'm worried about the amount of extra work on this project and the deadline.
System$_4$	What is the project about?

Table 1: Dialogue example

A video demonstration of the system in action is available at:
`http://www.youtube.com/watch?v=BmDMNguQUmM`

3 Architecture

Figure 1 shows a screen shot taken at run-time of actual system output. The ECA is represented on a screen as a woman (waist up) who displays natural, human-like movements and performs a wide range of complex facial expressions, bodily movements, and hand and arm gestures.

The screen also displays a transcript of the user and system turns. The user turns shown constitute the output of the Automatic Speech Recogniser (ASR). The system's analysis of the user's emotional state is also shown.

The right-most panel of the screen shows graphics which convey real-time information about how the dialogue is being processed. It presents a streamlined view of the software modules that comprise the system. Module activity is visually represented at run-time by flashing colours. This 'glass-box' approach enables detailed observation and analysis of system procedure at run-time.

The system comprises a number of distinct modules that are connected using Inamode, a hub-based message-passing framework using XML formatted messages over plain text sockets.

The system's ASR is the Nuance™ dictation engine. This is run in parallel with our own acoustic analysis pipeline which extracts low level (pitch, tone) speech features and also high-level features such as emotional characteristics. Analysis of the emotions is currently carried out

by EmoVoice (Vogt et al. (2008)). The ASR output strings are analysed for sentiment by the AFFECTiS system (Moilanen and Pulman (2007, 2009)) and classed as positive, neutral, or negative. This output is fused with the output from EmoVoice to generate a value that represents the user's current emotional state, which is expressed as a valence+arousal pairing (with five possible values).

The ASR output goes to our own Natural Language Understanding (NLU) module which performs syntactic and semantic analysis of user utterances and derives noun phrases and verb groups and associated arguments. Events relevant to the scenario (*e.g.*, promotions, redundancies, meetings, arguments, *etc*.) are recognised by the NLU and are used to populate an ontology (a model of the conversation content). The system is currently able to recognize and respond to more than 30 event types.

The events recognised in a user turn are labelled with the output of the Emotion Module for that turn; the result is a representation of both the semantic and affective information that the user might be trying to convey.

Our own rule-based Dialogue Manager (DM) takes the affect-annotated semantic output of the NLU, and from that and its model of the conversation content determines the next system turn. It will either ask a question about the events that occurred in the user's day, express an opinion on the events already described, or make empathetic comments. Whenever the system has gained sufficient understanding of a key event in the user's day, it generates a complex long turn that encapsulates comfort, opinion, warnings and advice to the user.

These long system turns are generated by our own plan-based Affective Strategy Module that makes an appraisal of the user's situation and generates an appropriate emotional strategy (Cavazza et al. (2010)). This strategy—expressed as an abstract, conceptual representation—is handed to our own Natural Language Generator (NLG) that maps it into a series of linguistic surface forms (usually 4 or 5 sentences). We use a style-controllable system using Tree-Furcating Grammars (an extension of the Tree-Adjoining Grammars formalism (Joshi et al. (1997)). This ensures the generation of a large set of different surface forms from the same semantic input.

The output of the NLG is passed to a module that adds this information to its system turn instructions for the ECA. The ECA has been developed around the Haptek™ toolkit and is controlled using an FML-like language (after Hernández et al. (2008)). This 2-D embodiment produces gestures, facial expressions, and body movements that convey the emotional state of the ECA. Its movements and expressions enable it to visually display interest and enjoyment in talking to the user, and to display empathy with the user. The speech synthesis module is our own emotion-focused extension of the Loquendo™ TTS system. It includes paralinguistic elements such as exclamations and laughter, and emotional prosody generation for negative and positive utterances.

4 Special procedural features

A significant processing design feature of the system is that there are two main processing loops from user input to system output; a 'long loop' which passes through all the components of the system; and a 'short loop' or 'feedback loop' which will now be discussed (the procedure already described in Section 3 is the long loop procedure).

4.1 Feedback loop

The feedback loop ('short loop') bypasses many linguistic components and generates immediate reactions to user activity. The main function of the short loop is maintain user engagement by preventing unnaturally long gaps of ECA inactivity. The feedback loop engages the acoustic analysis components, the TTS, and the ECA. It is responsible for the generation of real-time (< 500 ms) reactions in the ECA in response to the emotional state of the user. It attempts to align both verbal behaviour (backchannelling) and non-verbal behaviour (facial expressions, gestures, and general body language) to the emotions detected during most recent user turn. In order to achieve a reasonable level of realism, these system reactions to the perceived emotional state of the user need to be perceptibly instantaneous. Using this short feedback loop that bypasses many of the linguistic components ensures this.

The feedback loop is also occasionally used to make sympathetic comments immediately after the user stops speaking. These act as acknowledgements of the emotion expressed by the user. An example can be seen in the $System_2$ turn of the example dialogue in Table 1:

1.*"Sorry to hear that. Who was at the meeting?"*

Here, the first utterance was spoken by the system within a few tenths of a second after the end

of the previous user turn ($User_2$). The system tried to identify the user's emotion in the previous turn and then to behave linguistically and visually in an empathetic way. The actual sympathetic utterance was randomly chosen from a set of 'negative emotion utterances' (there are also 'positive' and 'neutral' sets).

The second half of the system turn in (1) was derived by the system's 'long loop'. It is a question which refers to a meeting that the user mentioned in the previous turn. This 'meeting' event has been heard by the ASR, understood by the NLU system, remembered by the DM, and is now referred to by an appropriate definite noun phrase in the output of the NLG.

The feedback and main loops run in parallel. However, the feedback loop generates its speech output almost immediately, giving time for the main dialogue loop to complete its more detailed analysis of the user's utterance.

4.2 Handling user interruptions

This system has a complex strategy for handling situations in which the user interrupts long system turns. The system's response to 'barge-in' user interruptions is overseen by the Interruption Manager (IM), which is alerted by the acoustic input modules whenever a genuine user interruption (as opposed to, say, a backchannel) is detected during a long system utterance. When alerted, the IM instructs the ECA to stop speaking when it reaches a natural stopping point in its current turn (usually the end of the current phrase). The user's interruption utterance is processed by the long loop. Its progress is tracked and controlled by the IM, for example, it makes sure that the linguistic modules know that the current utterance is an interruption, whic means it requires special treatment. The DM has a range of strategies for system recoveries from user interruptions, including different ways of continuing, replanning, and aborting. An example of a user interruption is shown in Table 1. The user interrupts the long system utterance in the $System_3$ turn. The system's response to the interruption is to stop the speech output from the ECA, abort the long system turn altogether, and instead to ask for more details about the project that the user has just mentioned during the interruption. (See (Crook et al. (2010)) for a more detailed description of the IM.)

Acknowledgements

This work was funded by Companions, a European Commission Sixth Framework Programme Information Society Technologies Integrated Project (IST-34434).

We would also like to thank the following people for their valuable contributions to the work presented here: Stephen Pulman, Ramon Granell, and Simon Dobnick (Oxford University), Johan Boye (KTH Stockholm), Cameron Smith and Daniel Charlton (Teesside University), Roger Moore, WeiWei Cheng and Lei Ye (University of Sheffield), Morena Danieli and Enrico Zovato (Loquendo).

References

Cavazza, M., Smith, C., Charlton, D., Crook, N., Boye, J., Pulman, S., Moilanen, K., Pizzi, D., Santos de la Camara, R., Turunen, M. 2010 *Persuasive Dialogue based on a Narrative Theory: an ECA Implementation*, Proc. of the 5th Int. Conf. on Persuasive Technology (Persuasive 2010), to appear 2010.

Crook, N., Smith, C., Cavazza, M., Pulman, S., Moore, R., and Boye, J. 2010 *Handling User Interruptions in an Embodied Conversational Agent* In proc. of AAMAS 2010.

Hernández, A., López, B., Pardo, D., Santos, R., Hernández, L., Relaño Gil, J. and Rodríguez, M.C. (2008) *Modular definition of multimodal ECA communication acts to improve dialogue robustness and depth of intention*. In: Heylen, D., Kopp, S., Marsella, S., Pelachaud, C., and Vilhjálmsson, H. (Eds.), AAMAS 2008 Workshop on Functional Markup Language.

Joshi, A.K. & Schabes, Y. (1997) Tree-adjoining Grammars. *Handbook of formal languages, vol. 3: Beyond Words*, Springer-Verlag New York, Inc., New York, NY, 1997.

Moilanen, K. and Pulman. S. (2009). Multi-entity Sentiment Scoring. *Proc. Recent Advances in Natural Language Processing* (RANLP 2009). September 14-16, Borovets, Bulgaria. pp. 258--263.

Moilanen, K. and Pulman. S. (2007). Sentiment Composition. *Proc. Recent Advances in Natural Language Processing* (RANLP 2007). September 27-29, Borovets, Bulgaria. pp. 378--382.

Vogt, T., André, E. and Bee, N. 2008. *EmoVoice – A framework for online recognition of emotions from voice. Proc. Workshop on Perception and Interactive Technologies for Speech-Based Systems*, Springer, Kloster Irsee, Germany, (June 2008).

F^2 – New Technique for Recognition of User Emotional States in Spoken Dialogue Systems

Ramón López-Cózar
Dept. of Languages and Computer Systems, CTIC-UGR, University of Granada, Spain
rlopezc@ugr.es

Jan Silovsky
Institute of Information Technology and Electronics, Technical University of Liberec, Czech Republic
jan.silovsky@tul.cz

David Griol
Dept. of Computer Science Carlos III University of Madrid, Spain
dgriol@inf.uc3m.es

Abstract

In this paper we propose a new technique to enhance emotion recognition by combining in different ways what we call *emotion predictions*. The technique is called F^2 as the combination is based on a double fusion process. The input to the first fusion phase is the output of a number of classifiers which deal with different types of information regarding each sentence uttered by the user. The output of this process is the input to the second fusion stage, which provides as output the most likely emotional category. Experiments have been carried out using a previously-developed spoken dialogue system designed for the fast food domain. Results obtained considering three and two emotional categories show that our technique outperforms the standard single fusion technique by 2.25% and 3.35% absolute, respectively.

1 Introduction

Automatic recognition of user emotional states is a very challenging task that has attracted the attention of the research community for several decades. The goal is to design methods to make computers interact more naturally with human beings. This is a very complex task due to a variety of reasons. One is the absence of a generally agreed definition of emotion and of qualitatively different types of emotion. Another is that we still have an incomplete understanding of how humans process emotions, as even people have difficulty in distinguishing between them. Thus, in many cases a given emotion is perceived differently by different people.

Studies in emotion recognition made by the research community have been applied to enhance the quality or efficiency of several services provided by computers. For example, these have been applied to spoken dialogue systems (SDSs) used in automated call-centres, where the goal is to detect problems in the interaction and, if appropriate, transfer the call automatically to a human operator.

The remainder of the paper is organised as follows. Section 2 addresses related work on the application of emotion recognition to SDSs. Section 3 focuses on the proposed technique, describing the classifiers and fusion methods employed in the current implementation. Section 4 discusses our speech database and its emotional annotation. Section 5 presents the experiments, comparing results obtained using the standard single fusion technique with the proposed double fusion. Finally, Section 6 presents the conclusions and outlines possibilities for future work.

2 Related work

Many studies can be found in the literature addressing potential improvements to SDSs by recognising user emotional states. A diversity of speech databases, features used for training and recognition, number of emotional categories, and recognition methods have been proposed. For example, Batliner et al. (2003) employed three different databases to detect troubles in communication. One was collected from a single experienced actor who was told to express anger because of system malfunctions. Other was collected from *naive* speakers who were asked to read neutral and emotional sentences. The third database was collected using a WOZ scenario designed to deliberately provoke user reactions to system malfunctions. The study focused on detecting two emotion categories: emotional (e.g. anger) and neutral, employing classifiers that dealt with prosodic, linguistic, and discourse information.

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 281–288,
The University of Tokyo, September 24-25, 2010.

Liscombe et al. (2005) made experiments with a corpus of 5,690 dialogues collected with the "How May I Help You" system, and considered seven emotional categories: positive/neutral, somewhat frustrated, very frustrated, somewhat angry, very angry, somewhat other negative, and very other negative. They employed standard lexical, prosodic and contextual features.

Devillers and Vidrascu (2006) employed human-to-human dialogues on a financial task, and considered four emotional categories: anger, fear, relief and sadness. Emotion classification was carried out considering linguistic information and paralinguistic cues.

Ai et al. (2006) used a database collected from 100 dialogues between 20 students and a spoken dialogue tutor, and for classification employed lexical items, prosody, user gender, beginning and ending time of turns, user turns in the dialogue, and system/user performance features. Four emotional categories were considered: uncertain, certain, mixed and neutral.

Morrison et al. (2007) compared two emotional speech data sources The former was collected from a call-centre in which customers talked directly to a customer service representative. The second database was collected from 12 non-professional actors and actresses who simulated six emotional categories: anger, disgust, fear, happiness, sadness and surprise.

3 The proposed technique

The technique that we propose in this paper to enhance emotion recognition in SDSs considers that a set of classifiers $\Omega = \{C_1, C_2, \ldots, C_m\}$ receive as input feature vectors f related to each sentence uttered by the user. As a result, each classifier generates one emotion prediction, which is a vector of pairs (h_i, p_i), $i = 1 \ldots S$, where h_i is an emotional category (e.g. Angry), p_i is the probability of the utterance belonging to h_i in accordance with the classifier, and S is the number of emotional categories considered, which forms the set $E = \{e_1, e_2, \ldots, e_S\}$.

The emotion predictions generated by the classifiers make up the input to the first fusion stage, which we call Fusion-0. This stage employs n fusion methods called F_{0i}, $i = 1 \ldots n$, to generate other predictions: vectors of pairs $(h_{0j,k}, p_{0j,k})$, $j = 1 \ldots n$, $k = 1 \ldots S$, where $h_{0j,k}$ is an emotional category, and $p_{0j,k}$ is the probability of the utterance belonging to $h_{0j,k}$ in accordance with the fusion method F_{0j}.

The second fusion stage, called Fusion-1, receives the predictions provided by Fusion-0 and generates the pair $(h_{11,1}, p_{11,1})$, where $h_{11,1}$ is the emotional category with highest probability, $p_{11,1}$. This emotional category is determined employing a fusion method called F_{11}, and represents the user's emotional state deduced by the technique. The best combination of fusion methods to be used in Fusion-0 (F_{01}, $F_{02}, \ldots, F_{0j}$, $1 \leq j \leq n$) and the best fusion method to be used in Fusion-1 (F_{11}) must be experimentally determined.

3.1 Classifiers

In the current implementation our technique employs four classifiers, which deal with prosody, acoustics, lexical items and dialogue acts regarding each utterance.

3.1.1 Prosodic classifier

The input to our prosodic classifier is an n-dimensional feature vector obtained from global statistics of pitch and energy, and features derived from the duration of voiced/unvoiced segments in each utterance. After carrying out experiments to find the appropriate feature set for the classifier, we decided to use the following 11 features: pitch mean, minimum and maximum, pitch derivatives mean, mean and variance of absolute values of pitch derivatives, energy maximum, mean of absolute value of energy derivatives, correlation of pitch and energy derivatives, average length of voiced segments, and duration of longest monotonous segment.

The classifier employs gender-dependent Gaussian Mixture Models (GMMs) to represent emotional categories. The likelihood for the n-dimensional feature vector (x), given an emotional category λ, is defined as:

$$P(x|\lambda) = \sum_{l=1}^{Q} w_l P_l(x)$$

i.e., a weighted linear combination of Q unimodal Gaussian densities $P_l(x)$. The density function $P_l(x)$ is defined as:

$$P_l(x) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma_l}} \exp\left(-\frac{1}{2}(x-\mu_l)' \Sigma_l^{-1} (x-\mu_l)\right)$$

where the μ_l's are mean vectors and the Σ_l's covariance matrices. The emotional category

deduced by the classifier, h, is decided according to the following expression:

$$h = \arg\max_{S} P\left(x\middle|\lambda^{S}\right) \quad (1)$$

where λ^S represents the models for the emotional categories considered, and the *max* function is computed employing the EM (Expectation-Maximization) algorithm. To compute the probabilities p_i for the emotion prediction of the classifier we use the following expression:

$$p_i = \beta_i / \sum_{k=1}^{S} \beta_k \quad (2)$$

where β_i is the log-likelihood of h_i, S is the number of emotional categories considered, and the β_k's are the log-likelihoods of these emotional categories.

3.1.2 Acoustic classifier

Prosodic features are nowadays among the most popular features for emotion recognition (Dellaert et al. 1996; Luengo et al. 2005). However, several authors have evaluated other features. For example, Nwe et al. (2003) employed several short-term spectral features and observed that Logarithmic Frequency Power Coefficients (LFPCs) provide better performance than Mel-Frequency Cepstral Coefficient (MFCCs) or Linear Prediction Cepstral Coefficients (LPCCs). Experiments carried out with our speech database (which will be discussed in Section 4) have confirmed this observation. However, we have also noted that when we used the first and second derivatives, the best results were obtained for MFCCs. Hence, we decided to use 39-feature MFCCs (13 MFCCs, delta and delta-delta) for classification.

The emotion patterns of the input utterances are modelled by gender-dependent GMMs, as with the prosodic classifier, but each input utterance is represented employing a sequence of feature vectors $x = \{x_1,\ldots,x_T\}$ instead of one n-dimensional vector. We assume mutual independence of the feature vectors in x, and compute the log-likelihood for an emotional category λ as follows:

$$P\left(x\middle|\lambda\right) = \sum_{t=1}^{T} \log P\left(x_t\middle|\lambda\right)$$

The emotional category deduced by the classifier, h, is decided employing Eq. (1), whereas Eq. (2) is used to compute the probabilities for the prediction, i.e. for the vector of pairs (h_i, p_i).

3.1.3 Lexical classifier

A number of previous studies on emotion recognition take into account information about the kinds of word uttered by the users, assuming that there is a relationship between words and emotion categories. For example, swear words and insults can be considered as conveying a negative emotion (Lee and Narayanan, 2005). Analysis of our dialogue corpus (which will be discussed in Section 4) has shown that users did not utter swear words or insults during the interaction with the Saplen system. Nevertheless, there were particular moments in the interaction at which their emotional state changed from Neutral to Tired or Angry. These moments correspond to dialogue states where the system had problems in recognising the sentences uttered by the users.

The reasons for these problems are basically two. On the one hand, most users spoke with strong southern Spanish accents, characterised by the deletion of the final *s* of plural words, and an exchange of the phonemes *s* and *c* in many words. On the other hand, there are words in the system's vocabulary that are very similar acoustically.

Hence, our goal has been to automatically find these words by means of a study of the speech recognition results, and deduce the emotional category for each input utterance from the emotional information associated with the words in the recognition result. To do this we have followed the study of Lee and Narayanan (2005), which employs the information-theoretic concept of *emotional salience*. The emotional salience of a word for a given emotional category can be defined as the mutual information between the word and the emotional category. Let W be a sentence (speech recognition result) comprised of a sequence of n words: $W = w_1\ w_2\ \ldots w_n$, and E a set of emotional categories, $E = \{e_1, e_2, \ldots, e_S\}$. The mutual information between the word w_i and an emotional category e_j is defined as follows:

$$mutual\_Information(w_i, e_j) = \log \frac{P(e_j \mid w_i)}{P(e_j)}$$

where $P(e_j \mid w_i)$ is the posterior probability that a sentence containing the word w_i implies the emotional category e_j, and $P(e_j)$ represents the prior probability of the emotional category.

Taking into account the previous definitions, we have defined the emotional salience of the word w_i for an emotional category e_j as follows:

$$salience(w_i, e_j) = P(e_j \mid w_i) \times mutual\_Information(w_i, e_j)$$

After the salient words for each emotional category have been identified employing a training corpus, we can carry out emotion recognition at the sentence level, considering that each word in the sentence is independent of the rest. The goal is to map the sentence W to any of the emotional categories in E. To do this, we compute an activation value a_k for each emotional category as follows:

$$a_k = \sum_{m=1}^{n} I_m w_{mk} + w_k$$

where $k = 1 \ldots S$, n is the number of words in W, I_m represents an indicator that has the value 1 if w_k is a salient word for the emotional category (i.e. $salience(w_i,e_j) \neq 0$) and the value 0 otherwise; w_{mk} is the connection weight between the word and the emotional category, and w_k represents bias. We define the connection weight w_{mk} as:

$$w_{mk} = mutual\_Information(w_m, e_k)$$

whereas the bias is computed as: $w_k = \log P(e_k)$. Finally, the emotional category deduced by the classifier, h, is the one with highest activation value a_k:

$$h = \arg\max_k (a_k)$$

To compute the probabilities p_i's for the emotion prediction, we use the following expression:

$$p_i = a_i / \sum_{j=1}^{S} a_j$$

where a_i represents the activation value of h_i, and the a_j's are the activation values of the S emotional categories considered.

3.1.4 Dialogue acts classifier

A dialogue act can be defined as the function performed by an utterance within the context of a dialogue, for example, greeting, closing, suggestion, rejection, repeat, rephrase, confirmation, specification, disambiguation, or help (Batliner et al. 2003; Lee and Narayanan, 2005; Liscombe et al. 2005).

Our dialogue acts classifier is inspired by the study of Liscombe et al. (2005), where the sequential structure of each dialogue is modelled by a sequence of dialogue acts. A difference is that they assigned one or more labels related to dialogue acts to each user utterance, and did not assign labels to system prompts, whereas we assign just one label to each system prompt and none to user utterances. This decision is made from the examination of our dialogue corpus. We have observed that users got tired or angry if the system generated the same prompt repeatedly (i.e. repeated the same dialogue act) to try to get a particular data item. For example, if it had difficulty in obtaining a telephone number then it employed several dialogue turns to obtain the number and confirm it, which annoyed the users, especially if they had employed other turns previously to correct misunderstandings. Hence, our dialogue act classifier aims to predict these negative emotional states by detecting successive repetitions of the same system's prompt types (e.g. prompts to get the telephone number).

In accordance with our approach, the emotional category of a user's dialogue turn, E_n, is that which maximises the posterior probability given a sequence of the most recent system prompts:

$$E_n = \arg\max_k P(E_k \mid DA_{n-(L*2-1)}, \ldots, DA_{n-3}, DA_{n-1})$$

where the prompt sequence is represented by a sequence of dialogue acts (DA_i's) and L is the length of the sequence, i.e. the number of system's dialogue turns in the sequence. Note that if $L = 1$ then the decision about E_n depends only on the previous system prompt. In other words, the emotional category obtained is that with the greatest probability given just the previous system turn in the dialogue. The probability of the considered emotional categories

given a sequence of dialogue acts is obtained by employing a training dialogue corpus.

By means of this equation, we decide the most likely emotional category for the input utterance, selecting the category with the highest probability given the sequence of dialogue acts of length L. This probability is used to create the pair (h_i, p_i) to be included in the emotion prediction.

3.2 Fusion methods

In the current implementation our technique employs the three fusion methods discussed in this section. When used in Fusion-0, these methods are employed to combine the predictions provided by the classifiers. When used in Fusion-1, they are used to combine the predictions generated by Fusion-0.

3.2.1 Average of probabilities (AP)

This method combines the predictions by averaging their probabilities. To do this we consider that each input utterance is represented by feature vectors $x^1,\ldots,x^m$ from feature spaces $X^1,\ldots,X^m$, where m is the number of classifiers. We also assume that each input utterance belongs to one of S emotional categories h_i, $i = 1\ldots S$. In each of the m feature spaces a classifier can be created that approximates the posterior probability $P(h_i \mid x^k)$ as follows:

$$f_i^k(x^k) = P(h_i \mid x^k) + \varepsilon_i^k(x^k)$$

where $\varepsilon_i^k(x^k)$ is the error made by classifier k. We estimate $P(h_i \mid x^k)$ by $f_i^k(x^k)$ and assuming a zero-mean error for $\varepsilon_i^k(x^k)$, we average all the $f_i^k(x^k)$'s to obtain a less error-sensitive estimation. In this way we obtain the following mean combination rule to decide the most likely emotional category:

$$P(h_i \mid x^1, \ldots, x^m) = \frac{1}{m}\sum_{k=1}^{m} f_i^k(x^k)$$

3.2.2 Multiplication of probabilities (MP)

Assuming that the feature spaces $X^1,\ldots,X^m$ are different and independent, the probabilities can be written as follows:

$$P(x^1, \ldots, x^m \mid h_i) = P(x^1 \mid h_i) \times P(x^2 \mid h_i) \times \ldots \times P(x^m \mid h_i)$$

Using Bayes rule we can obtain the following equation, which we use to decide the most likely emotional category for each input utterance (represented as feature vectors $x^1,\ldots,x^m$):

$$P(h_i \mid x^1, \ldots, x^m) = \frac{\prod_k P(h_i \mid x^k) / P(h_i)^{m-1}}{\sum_{i'}\left\{\prod_{k'} P(h_{i'} \mid x^{k'}) / P(h_{i'})^{m-1}\right\}}$$

3.2.3 Unweighted vote (UV)

This method combines the emotion predictions by counting the number of classifiers (if used in Fusion-0) or fusion methods (if used in Fusion-1) that consider an emotional category h_i as the most likely for the input utterance. If we consider three emotional categories X, Y and Z, h_i is decided as follows:

$$h_i = \begin{cases} X & \text{if } \sum_{j=1}^{m} X_j \geq \sum_{j=1}^{m} Y_j \text{ and } \sum_{j=1}^{m} X_j \geq \sum_{j=1}^{m} Z_j \\ Y & \text{if } \sum_{j=1}^{m} Y_j \geq \sum_{j=1}^{m} X_j \text{ and } \sum_{j=1}^{m} Y_j \geq \sum_{j=1}^{m} Z_j \\ Z & \text{if } \sum_{j=1}^{m} Z_j \geq \sum_{j=1}^{m} X_i \text{ and } \sum_{j=1}^{m} Z_j \geq \sum_{j=1}^{m} Y_j \end{cases}$$

where m is the number of classifiers or fusion methods employed (e.g., in our experiments, X = Neutral, Y = Tired and Z = Angry). The probability p_i for h_i to be included in the emotion prediction is computed as follows:

$$P(h_i \mid X, Y, Z) = Vh_i / \sum_{j=1}^{3} Vh_j$$

where Vh_i is the number of votes for h_i, and the Vh_j's are the number of votes for the 3 emotional categories. If we consider two emotional categories X and Y, the most likely emotional category h_i and its probability p_i are analogously computed (e.g., in our experiments, X = Non-negative and Y = Negative).

4 Emotional speech database

Our emotional speech database has been constructed from a corpus of 440 telephone-based dialogues between students of the University of Granada and the Saplen system, which was

previously developed in our lab for the fast food domain (López-Cózar et al. 1997; López-Cózar and Callejas, 2006). Each dialogue was stored in a log file in text format that includes each system prompt (e.g. *Would you like to drink anything?*), the type of prompt (e.g. AnyFoodOrDrinkToOrder?), the name of the voice samples file (utterance) that stores the user response to the prompt, and the speech recognition result for the utterance. The dialogue corpus contains 7,923 utterances, 50.3% of which were recorded by male users and the remaining by female users.

The utterances have been annotated by 4 labellers (2 male and 2 female). The order of the utterances has been randomly chosen to avoid influencing the labellers by the situation in the dialogues, thus minimising the effect of discourse context. The labellers have initially assigned one label to each utterance, either <NEUTRAL>, <TIRED> or <ANGRY> according to the perceived emotional state of the user. One of these labels has been finally assigned to each utterance according to the majority opinion of the labellers, so that 81% of the utterances are annotated as 'Neutral', 9.5% as 'Tired' and 9.4% as 'Angry'. This shows that the database is clearly unbalanced in terms of emotional categories.

To measure the amount of agreement between the labellers we employed the Kappa statistic (*K*), which is computed as follows (Cohen, 1960):

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where *P(A)* is the proportion of times that the labellers agree, and *P(E)* is the proportion of times we would expect the labellers to agree by chance. We obtained that $K = 0.48$ and $K = 0.45$ for male and female labellers, respectively, which according to Landis and Koch (1977) represents *moderate agreement*.

5 Experiments

The main goal of the experiments has been to test the proposed technique using our emotional speech database, and employing:

- Three emotional categories (Neutral, Angry and Tired) on the one hand, and two emotional categories (Non-negative and Negative) on the other. The experiments employing the former category set will be called *3-emotion* experiments, whereas those employing the latter category will be called *2-emotion* experiments.
- The four classifiers described in Section 3.1, and the three fusion methods discussed in Section 3.2.

In the 3-emotion experiments we consider that an input utterance is correctly classified if the emotional category deduced by the technique matches the label assigned to the utterance. In the 2-emotion experiments, the utterance is considered to be correctly classified if either the deduced emotional category is Non-negative and the label is Neutral, or the category is Negative and the label is Tired or Angry.

To carry out training and testing we have used a script that takes as its input a set of labelled dialogues in a corpus, and processes each dialogue by locating within it, from the beginning to the end, each prompt of the Saplen system, the voice samples file that contains the user's response to the prompt, and the result provided by the system's speech recogniser (sentence in text format). The type of each prompt is used to create a sequence of dialogue acts of length *L*, which is the input to the dialogue acts classifier. The voice samples file is the input to the prosodic and acoustic classifiers, and the speech recognition result is the input to the lexical classifier. This procedure is repeated for all the dialogues in the corpus.

Experimental results have been obtained using 5-fold cross-validation, with each partition containing the utterances corresponding to 88 different dialogues in the corpus.

5.1 Performance of Fusion-0

Table 1 sets out the average results obtained for Fusion-0 considering several combinations of the classifiers and employing the three fusion methods. As can be observed, MP is the best fusion method, with average classification rates of 89.08% and 87.43% for the 2 and 3 emotion experiments, respectively. The best classification rates (92.23% and 90.67%) are obtained by employing the four classifiers, which means that the four types of information considered (acoustic, prosodic, lexical and related to dialogue acts) are really useful to enhance classification rates.

Fusion Method	Classifiers	2 emot.	3 emot.
AP	Aco, Pro	84.15	82.46
	Lex, Pro	85.04	82.71
	DA, Pro	90.49	87.48
	Aco, Lex, Pro	89.20	86.17
	Aco, DA, Pro	90.24	88.56
	DA, Lex, Pro	90.02	88.02
	Aco, DA, Lex, Pro	**90.49**	**88.32**
	Average	88.66	86.25
MP	Aco, Pro	84.15	82.86
	Lex, Pro	85.16	83.71
	DA, Pro	91.49	89.78
	Aco, Lex, Pro	89.17	87.91
	Aco, DA, Pro	91.33	89.23
	DA, Lex, Pro	90.06	87.82
	Aco, DA, Lex, Pro	**92.23**	**90.67**
	Average	89.08	87.43
UV	Aco, Pro	88.64	85.19
	Lex, Pro	86.40	83.01
	DA, Pro	88.20	84.92
	Aco, Lex, Pro	88.76	85.54
	Aco, DA, Pro	88.91	85.89
	DA, Lex, Pro	88.47	85.61
	Aco, DA, Lex, Pro	**89.04**	**87.56**
	Average	88.35	85.39

Table 1: Performance of Fusion-0 (results in %).

5.2 Performance of Fusion-1

Table 2 shows the average results obtained when Fusion-1 is used to combine the predictions of Fusion-0. The three fusion methods are tested in Fusion-1, with Fusion-0 employing four combinations of these methods: AP,MP; AP,UV; MP,UV; and AP,MP,UV. In all cases Fusion-0 uses the four classifiers as this is the configuration that provides the highest classification accuracy according to the previous section.

Comparison of both tables shows that Fusion-1 clearly outperforms Fusion-0. The best results are attained for MP, which means that this method is preferable when the data contain small errors (emotion predictions generated by Fusion-0 with accuracy rates around 90%).

To find the reasons for these enhancements we have analysed the confusion matrix of Fusion-1 using MP. The study reveals that for the 2-emotion experiments this fusion stage works very well in predicting the Non-negative category, very slightly enhancing the classification rate of Fusion-0 (96.58% vs. 95.93%), whereas the classification rate of the Negative category is the same as that obtained by Fusion-0 (88.91%). Overall, the best performance of Fusion-1 employing MP (94.48%) outdoes that of Fusion-0 employing AP (90.49%) and MP (92.23%).

Regarding the 3-emotion experiments, our analysis shows that using MP, Fusion-1 slightly lowers the classification rate of the Neutral category obtained by Fusion-0 (97.79% vs. 97.9%), but slightly raises the rate of the Tired category (93.62% vs. 93.26%), and the Angry category (77.49% vs. 76.81%). Overall, the performance of Fusion-1 employing MP (94.02%) outdoes that of Fusion-0 employing AP (88.32%) and MP (90.67%).

Fusion methods used in Fusion-0	Fusion method used in Fusion-1 (2 emotions)			Fusion method used in Fusion-1 (3 emotions)		
	AP	MP	UV	AP	MP	UV
AP,MP	93.68	**94.48**	93.53	91.77	**94.02**	90.96
AP,UV	93.20	93.23	93.20	91.65	93.13	90.10
MP,UV	93.34	94.38	93.20	91.27	93.98	89.48
AP,MP,UV	93.23	94.36	93.17	91.57	93.97	89.06
Average	93.40	94.11	93.28	91.57	93.78	89.90

Table 2: Performance of Fusion-1 (results in %).

6 Conclusions and future work

Our experimental results show that the proposed technique is useful to improve the classification rates of the standard fusion technique, which employs just one fusion stage. Comparing results in **Table 1** and **Table 2** we can observe that for the 2-emotion experiments, Fusion-1 enhances Fusion-0 by 2.25% absolute (from 92.23% to 94.48%), while for the 3-emotion experiments, the improvement is 3.35% absolute (from 90.67% to 94.02%). These improvements are obtained by employing AP and MP in Fusion-0 to combine the emotion predictions of the four classifiers, and using MP in Fusion-1 to combine the outputs of Fusion-0.

The reason for these improvements is that the double fusion process (Fusion-0 and Fusion-1) allows us to benefit from the advan-

tages of using different methods to combine information. According to our results, the best methods are AP and MP. The former allows gaining maximally from the independent data representation available, which are the input to Fusion-0 (in our study, prosody, acoustics, speech recognition errors, and dialogue acts). The latter provides better results when the data contain small errors, which occurs when the predictions provided by Fusion-0 are the input to Fusion-1.

Future work will include testing the technique employing information sources not considered in this study. The sources we have dealt with in the experiments (prosodic, acoustic, lexical, and dialogue acts) are those most commonly employed in previous studies. However, there are also studies that suggest using other information sources, such as speaking style, subject and problem identification, and non-verbal cues.

Another future work is to test the technique employing other methods for classification and information fusion. For example, it is known that people are usually confused when they try to determine the emotional state of a speaker, given that the difference between some emotions is not always clear. Hence, it would be interesting to investigate the performance of the technique employing classification algorithms that deal with this vague boundary, such as fuzzy inference methods, and using boosting methods for improving the accuracy of the classifiers.

Finally, in terms of application of the technique to improve the system-user interaction, we will evaluate different dialogue management strategies to enable the system's adaptation to negative emotional states of users (University students). For example, a dialogue management strategy could be as follows: i) if the emotional state is Tired begin the following prompt apologising, and transfer the call to a human operator if this state is recognised twice consecutively, and ii) if the emotional state is Angry apologise and transfer the call to a human operator immediately.

Acknowledgments

This research has been funded by Spanish project HADA TIN2007-64718, and the Czech Grant Agency project no. 102/08/0707.

References

Ai, H., Litman, D. J., Forbes-Riley, K., Rotaru, M., Tetreault, J., Purandare, A. 2006. Using system and user performance features to improve emotion detection in spoken tutoring systems. Proc. of Interspeech, pp. 797-800.

Batliner, A., Fischer, K., Huber, R., Spilker, J., Nöth, E. 2003. How to find trouble in communication. *Speech Communication*, vol. 40, pp. 117-143.

Cohen, J. 1960. A coefficient of agreement for nominal scales. Educat. Psychol. Measurement, vol. 20, pp. 37-46.

Dellaert, F., Polzin, T., Waibel, A. 1996. Recognizing emotion in speech. Proc. of ICSLP, pp. 1970-1973.

Devillers, L., Vidrascu, L. 2006. Real-life emotions detection with lexical and paralinguistic cues on human-human call center dialogs. Proc. of Interspeech, pp. 801-804.

Landis, J. R., Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics*, vol. 33, pp. 159-174.

Lee, C. M., Narayanan, S. S. 2005. Toward detecting emotions in spoken dialogs. *IEEE Transactions on Speech and Audio Processing*, vol. 13(2), pp. 293-303.

Liscombe, J., Riccardi, G., Hakkani-Tür, D. 2005. Using context to improve emotion detection in spoken dialogue systems. Proc. of Interspeech, pp. 1845-1848.

López-Cózar, R., García, P., Díaz, J., Rubio, A. J. 1997. A voice activated dialog system for fast-food restaurant applications. Proc. of Eurospeech, pp. 1783-1786.

López-Cózar, R., Callejas, Z. 2006. Combining Language Models in the Input Interface of a Spoken Dialogue System. *Computer Speech and Language*, 20, pp. 420-440.

Luengo, I., Navas, E., Hernáez, I, Sanchez, J. 2005. Automatic emotion recognition using prosodic parameters. Proc. of Interspeech, pp.493-496.

Morrison, D., Wang, R., De Silva, L. C. 2007. Ensemble methods for spoken emotion recognition in call-centres. *Speech Communication*, vol. 49(2) pp. 98-112.

Nwe, T. L., Foo, S. V., De Silva, L. C. 2003. Speech emotion recognition using hidden Markov models. *Speech Communication*, vol. 41(4), pp. 603-623.

Online Error Detection of Barge-In Utterances by Using Individual Users' Utterance Histories in Spoken Dialogue System

Kazunori Komatani* **Hiroshi G. Okuno**

Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto 606-8501, Japan
{komatani,okuno}@kuis.kyoto-u.ac.jp

Abstract

We develop a method to detect erroneous interpretation results of user utterances by exploiting utterance histories of individual users in spoken dialogue systems that were deployed for the general public and repeatedly utilized. More specifically, we classify barge-in utterances into correctly and erroneously interpreted ones by using features of individual users' utterance histories such as their barge-in rates and estimated automatic speech recognition (ASR) accuracies. Online detection is enabled by making these features obtainable without any manual annotation or labeling. We experimentally compare classification accuracies for several cases when an ASR confidence measure is used alone or in combination with the features based on the user's utterance history. The error reduction rate was 15% when the utterance history was used.

1 Introduction

Many researchers have tackled the problem of automatic speech recognition (ASR) errors by developing ASR confidence measures based on utterance-level (Komatani and Kawahara, 2000) or dialogue-level information (Litman et al., 1999; Walker et al., 2000; Hazen et al., 2000). Especially in systems deployed for the general public such as those of (Komatani et al., 2005; Raux et al., 2006), the systems need to correctly detect interpretation errors caused by various utterances made by various users, including novices. Error detection using individual user models would be a promising way of improving performance in such systems because users often access them repeatedly (Komatani et al., 2007).

We choose to detect interpretation errors of barge-in utterances, mostly caused by ASR errors, as a task for showing the effectiveness of the user's utterance histories. We try to improve the accuracy of classifying barge-in utterances into correctly and erroneously interpreted ones without any manual labeling. By classifying utterances accurately, the system can reduce erroneous responses caused by the errors and unnecessary confirmations. Here, a "barge-in utterance" is a user utterance that interrupts the system's prompt. In this situation, the system stops its prompt and starts recognizing the user utterance.

In this study, we combine the ASR confidence measure with features obtained from the user's utterance history, i.e., the estimated ASR accuracy and the barge-in rate, to detect interpretation errors of barge-in utterances. We show that the features are still effective when they are used together with the ASR confidence measure, which is usually used to detect erroneous ASR results. The characteristics of our method are summarized as follows:

1. The user's utterance history used as his/her profile: The user's current barge-in rate and ASR accuracy are used for error detection.

2. Online user modeling: We try to obtain the user profiles listed above without any manual labeling after the dialogue has been completed. This means that the system can improve its performance while it is deployed.

In our earlier report (Komatani and Rudnicky, 2009), we defined the estimated ASR accuracy and showed that it is helpful in improving the accuracy of classifying barge-in utterances into correctly and erroneously interpreted ones, by using it in conjunction with the user's barge-in rate. In this

*Currently with Graduate School of Engineering, Nagoya University, Furo-cho, Chikusa-ku, Nagoya 464-8603, Japan. komatani@nuee.nagoya-u.ac.jp

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 289–296,
The University of Tokyo, September 24-25, 2010.

Table 1: ASR accuracy per barge-in

	Correct	Incorrect	Total	Accuracy
w/o barge-in	16,694	3,612	20,306	(82.2%)
w/ barge-in	3,281	3,912	7,193	(45.6%)
Total	19,975	7,524	27,499	(72.6%)

report, we verify our approach when the ASR confidence measure is also incorporated into it. Thus, we show the individual user's utterance history is helpful as a user profile and works as prior information for the ASR confidence.

2 Barge-in Utterance and its Errors

Barge-in utterances were often incorrectly interpreted mainly because of ASR errors in our data as shown in Table 1. The table lists the ASR accuracy per utterance for two cases: when the system prompts were played to the end (denoted as "w/o barge-in") and when the system prompts were barged in ("w/ barge-in"). Here, an utterance is assumed to be correct only when all content words in the utterance are correctly recognized; one is counted as an error if any word in it is misrecognized. Table 1 shows that barge-in utterances amounted to 26.2% (7,193/27,499) of all utterances, and half of those utterances contained ASR errors in their content words.

This result implies that many false barge-ins occurred despite the user's intention. Specifically, the false barge-ins included instances when background noises were incorrectly regarded as barge-ins and the system's prompt stopped. Such instances often occur when the user accesses the system using mobile phones in crowded places. Breathing and whispering were also prone to be incorrectly regarded as barge-ins. Moreover, disfluency in one utterance may be unintentionally divided into two portions, which causes further misrecognitions and unexpected system actions. The abovementioned phenomena, except background noises, are caused by the user's unfamiliarity with the system. That is, some novice users are not unaware of the timing at which to utter, and this causes the system to misrecognize the utterance. On the other hand, users who have already become accustomed to the system often use the barge-in functions intentionally and, accordingly, make their dialogues more efficient.

The results in Table 2 show the relationship between barge-in rate per user and the corresponding ASR accuracies of barge-in utterances. We

Table 2: ASR accuracy of barge-in utterances for different barge-in rates

Barge-in rate	Correct	Incorrect	Acc. (%)
0.0 - 0.2	407	1,750	18.9
0.2 - 0.4	205	842	19.6
0.4 - 0.6	1,602	880	64.5
0.6 - 0.8	1,065	388	73.3
0.8 - 1.0	2	36	5.3
1.0	0	16	0.0
Total	3,281	3,912	45.6

here ignore a small number of users whose barge-in rates were greater than 0.8, which means almost all utterances were barge-ins, because most of their utterances were misrecognized because of severe background noises and accordingly they gave up using the system. We thus focus on users whose barge-in rates were less than 0.8. The ASR accuracy of barge-in utterances was high for users who frequently barged-in. This suggests that the barge-ins were intentional. On the other hand, the ASR accuracies of barge-in utterances were less than 20% for users whose barge-in rates were less than 0.4. This suggests that the barge-ins of these users were unintentional.

A user study conducted by Rose and Kim (2003) revealed that there are many more disfluencies when users barge in compared with when users wait until the system prompt ends. Because such disfluencies and resulting utterance fragments are parts of human speech, it is difficult to select erroneous utterances to be rejected by using a classifier that distinguishes speech from noise on the basis of the Gaussian Mixture Model (Lee et al., 2004). These errors cannot be detected by using only bottom-up information obtained from single utterances such as acoustic features and ASR results.

To cope with the problem, we use individual users' utterance histories as their profiles. More specifically, we use each user's average barge-in rate and ASR accuracy from the time the user started using the system until the current utterance. The barge-in rate intuitively corresponds to the degree to which the user is accustomed to using the system, especially to using its barge-in function. That is, this reflects the tendency shown in Table 2; that is, the ASR accuracy of barge-in utterances is higher for users whose barge-in rates are higher. Each user's ASR accuracy also indicates the user's habituation. This corresponds to an empirical tendency that ASR accuracies of more accustomed

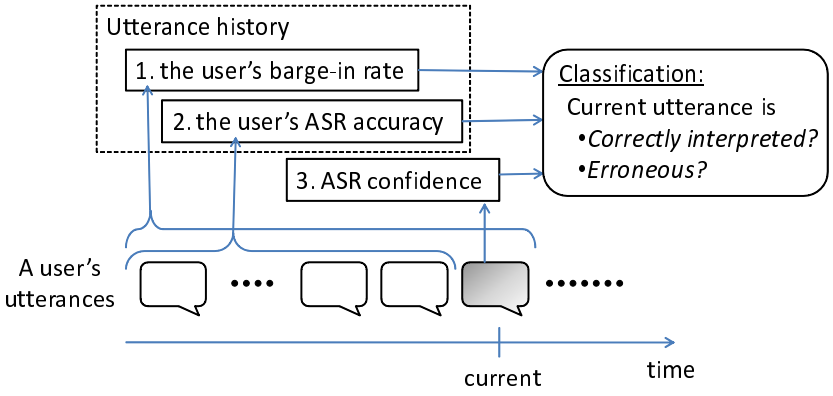


Figure 1: Overview of detecting interpretation errors

users are higher (Komatani et al., 2007; Levow, 2003). To account for another fact that some expert users have low barge-in rates, and, accordingly, not all expert users barge in frequently (Komatani et al., 2007), we use both the user's barge-in rate and ASR accuracy to represent degree of habituation, and verify their effectiveness as prior information for detecting erroneous interpretation results when they are used together with an ASR confidence measure.

To obtain the user's ASR accuracy without any manual labeling, we exploit certain dialogue patterns indicating that ASR results at certain positions are reliable. For example, Sudoh and Nakano (2005) proposed a "post-dialogue confidence scoring" in which ASR results corresponding to the user's intention upon dialogue completion are assumed to be correct and are used for confidence scoring. Bohus and Rudnicky (2007) proposed "implicitly supervised learning" in which user responses following the system's explicit confirmations are used for confidence scoring. If the ASR results can be regarded as reliable after the dialogue, machine learning algorithms can use them as teacher signals. This approach does not need any manual labeling or transcription, a task which requires much time and labor when spoken dialogue systems are being developed. We focus on users' affirmative and negative responses to the system's explicit confirmations, and estimated the user's ASR accuracy on the basis of his or her history of responses (Komatani and Rudnicky, 2009). This estimated ASR accuracy can be also used as an online feature representing a user's utterance history.

3 Detecting Errors by using the User's Utterance History

We detect interpretation errors of barge-in utterances by using the following three information sources:

1. the current user's barge-in rate,
2. the current user's ASR accuracy, and
3. ASR confidence of the current utterance.

The error detection method is depicted in Figure 1. Barge-in rate and ASR accuracy are accumulated and averaged from the beginning until the current utterance and are used as each user's utterance history. Then, at every point a user makes an utterance, the barge-in utterances are classified into correctly or erroneously interpreted ones by using a logistic regression function:

$$P = \frac{1}{1 + \exp(-(a_1 x_1 + a_2 x_2 + a_3 x_3 + b))}, \quad (1)$$

where x_1, x_2 and x_3 denote the barge-in rate, the ASR accuracy until the current utterance, and the ASR confidence measure of the current utterance, respectively. Coefficients a_i and b are determined by 10-fold cross validation on evaluation data. In the following subsections, we describe how to obtain these features.

3.1 Barge-In Rate

The barge-in rate is defined as the ratio of the number of barge-in utterances to all the user's utterances until the current utterance. Note that the current utterance itself is included in this calculation. We confirmed that the barge-in rate changes as the user becomes accustomed to the system

U1:	205. (Number 100)
S1:	Will you use bus number 100?
U2:	No. (No)
S2:	Please tell me your bus stop or bus route number.
U3:	Nishioji Matsu... *[disfluency]* (*Rejected*)
S3:	Please tell me your bus stop or bus route number.
U4:	From Nishioji Matsubara. (From Nishioji Matsubara)
S4:	Do you get on a bus at Nishioji Matsubara?
U5:	Yes. (Yes)

Initial characters 'U' and 'S' denote the user and system utterance.
A string in parentheses denotes the ASR result of the utterance.

Figure 2: Example dialogue

(Komatani et al., 2007). To take these temporal changes into consideration, we set a window when calculating the rate (Komatani et al., 2008). That is, when the window width is N, the rate is calculated on the basis of only the last N utterances, and utterances before those ones are discarded. When the window width exceeds the total number of utterances by the user, the barge-in rate is calculated on the basis of all the user's utterances. Thus, when the width exceeds 2,838, the maximum number of utterances made by one user in our data, the barge-in rates equal the average rates of all utterances by the user.

3.2 ASR Accuracy

ASR accuracy is calculated per utterance. It is defined as the ratio of the number of correctly recognized utterances to all the user's utterances until the previous utterance. Note that the current utterance is not included in this calculation. The "correctly recognized" utterance denotes a case when every content word in the ASR result of the utterance was correctly recognized and no content word was incorrectly inserted. The ASR accuracy of the user's initial utterance is regarded as 0, because there is no utterance before it. We do not set any window when calculating the ASR accuracies, because classification accuracy did not improve as a result of setting one (Komatani and Rudnicky, 2009). This is because each users' ASR accuracies tend to converge faster than the barge-in rates do (Komatani et al., 2007), and the changes in the ASR accuracies are relatively small in comparison with those of the barge-in rates.

We use two kinds of ASR accuracies:

1. actual ASR accuracy and
2. estimated ASR accuracy (Komatani and Rudnicky, 2009).

The actual ASR accuracy is calculated from manual transcriptions for investigating the upper limit of improvement of the classification accuracy when ASR accuracy is used. Thus, it cannot be obtained online because manual transcriptions are required.

The estimated ASR accuracy is calculated on the basis of the user's utterance history. This is obtainable online, that is, without the need for manual transcriptions after collecting the utterances. We focus on users' affirmative or negative responses following the system's explicit confirmations, such as "Leaving from Kyoto Station. Is that correct?" To estimate the accuracy, we make three assumptions as follows:

1. The ASR results of the users' affirmative or negative responses are correctly recognized. This assumption will be verified in Section 4.2.

2. A user utterance corresponding to the content of the affirmative responses is also correctly recognized, because the user affirms the system's explicit confirmation for it.

3. The remaining utterances are not correctly recognized. This corresponds to when users do not just say "no" in response to explicit confirmations with incorrect content and instead use other expressions.

To summarize the above, we assume that the ASR results of the following utterances are correct: an affirmative response, its corresponding utterance which is immediately preceded by it, and

Table 3: Distribution of ASR confidence measures for barge-in utterances

Confidence measure	Correct	Incorrect	(%)
0.0 - 0.1	0	1491	0.0
0.1 - 0.2	0	69	0.0
0.2 - 0.3	0	265	0.0
0.3 - 0.4	0	708	0.0
0.4 - 0.5	241	958	20.1
0.5 - 0.6	639	333	65.7
0.6 - 0.7	1038	68	93.9
0.7 - 0.8	1079	20	98.2
0.8 - 0.9	284	0	100.0
0.9 - 1.0	0	0	–
Total	3281	3912	45.6

a negative response. All other utterances are assumed to be incorrect. We thus calculate the user's estimated ASR accuracy as follows:

$$(\text{Estimated ASR accuracy}) = \frac{2 \times (\#\text{affirmatives}) + (\#\text{negatives})}{(\#\text{all utterances})} \quad (2)$$

Here is an example of the calculation for the example dialogue shown in Figure 2. U2 is a negative response, and U5 is an affirmative response. When the dialogue reaches the point of U5, U2 and U5 are regarded as correctly recognized on the basis of the first assumption. Next, U4 is regarded as correct on the basis of the second assumption, because the explicit confirmation for it (S4) was affirmed by the user as U5. Then, the remaining U1 and U3 are regarded as misrecognized on the basis of the third assumption. As a result, the estimated ASR accuracy at U5 is 60%.

The estimated ASR accuracy is updated for every affirmative or negative response by the user. For a neither affirmative nor negative response, the latest estimated accuracy before it was used instead.

3.3 ASR Confidence Measure

We use an ASR confidence measure calculated per utterance. Specifically, we use the one derived from the ASR engine in the Voice Web Server, a product of Nuance Communications, Inc.[1]

Table 3 shows the distribution of ASR confidence measures for barge-in utterances. By using this ASR confidence, even a naive method can have high classification accuracy (90.8%) in which just one threshold ($\theta = 0.516$) is set and utterances whose confidence measure is greater than the threshold are accepted. This accuracy is regarded as the baseline.

[1] http://www.nuance.com/

Table 4: ASR accuracy by user response type

	Correct	Incorrect	Total	(Acc.)
Affirmative	9,055	243	9,298	(97.4%)
Negative	2,006	286	2,292	(87.5%)
Other	8,914	6,995	15,909	(56.0%)
Total	19,975	7,524	27,499	(72.6%)

4 Experimental Evaluation

4.1 Data

We used data collected by the Kyoto City Bus Information System (Komatani et al., 2005). This system locates a bus that a user wants to ride and tells the user how long it will be before the bus arrives. The system was accessible to the public by telephone. It adopted the safest strategy to prevent erroneous responses; that is, it makes explicit confirmations for every user utterance except for affirmative or negative responses such as "Yes" or "No".

We used 27,499 utterances that did not involve calls whose phone numbers were not recorded or those the system developer used for debugging. The data contained 7,988 valid calls from 671 users. Out of these, there were 7,193 barge-in utterances (Table 1). All the utterances were manually transcribed for evaluation; human annotators decided whether every content word in the ASR results was correctly recognized or not.

The phone numbers of most of the calls were recorded, and we assumed that each number corresponded to one individual. Most of the numbers were those of mobile phones, which are usually not shared; thus, the assumption seems reasonable.

4.2 Verifying Assumption in Calculating Estimated ASR Accuracy

We confirmed our assumption that the ASR results of affirmative or negative responses following explicit confirmations are correct. We classified the user utterances into affirmatives, negatives, and other, and calculated the ASR accuracies (precision rates) per utterance as shown in Table 4. Affirmatives include *hai* ('yes'), *soudesu* ('that's right'), OK, etc; and negatives include *iie* ('no'), *chigaimasu* ('I don't agree'), *dame* ('No good'), etc. The table indicates that the ASR accuracies of affirmatives and negatives were high. One of the reasons for the high accuracy was that

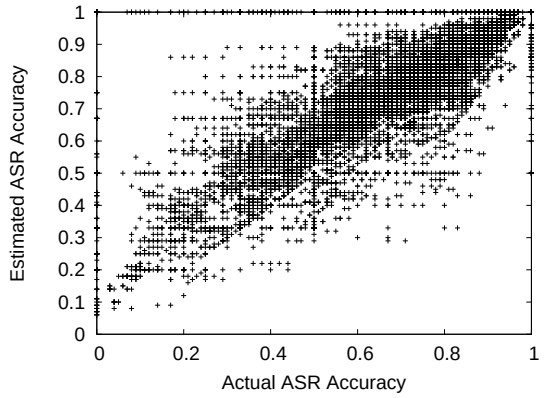


Figure 3: Correlation between actual and estimated ASR accuracy

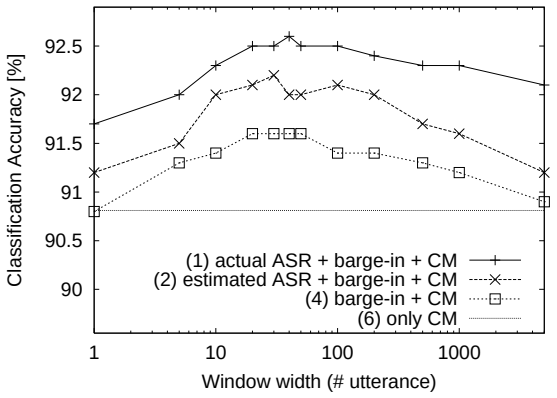


Figure 4: Classification accuracy when window width varies used to calculate barge-in rate

these utterances are much shorter than other content words, so they were less confused with other content words. Another reason was that the system often gave help messages such as "Please answer *yes* or *no*."

We then analyzed the correlation between the actual ASR accuracy and the estimated ASR accuracy based on Equation 2. We plotted the two ASR accuracies (Figure 3) for 26,231 utterances made after at least one affirmative/negative response by the user. The correlation coefficient between them was 0.806. Although the assumption that all ASR results of affirmative/negative responses are correct might be rather strong, the estimated ASR accuracy had a high correlation with the actual ASR accuracy.

4.3 Comparing Classification Accuracies When the Used Features Vary

We investigated the classification accuracy of the 7,193 barge-in utterances. The classification accuracies are shown in Table 5 in descending order for various sets of features x_i used as input into Equation 1. The conditions for when barge-in rates are used also show the window width w for the highest classification accuracy. The mean average error (MAE) is also listed, which is the average of the differences between an output of the logistic regression function X_j and a reference label manually given $\hat{X}_j$ (0 or 1):

$$MAE = \frac{1}{m}\sum_{j}^{m} |\hat{X}_j - X_j|, \qquad (3)$$

where m denotes the total number of barge-in utterances. This indicates how well the output of the logistic regression function (Equation 1) distributes. Regarding Condition (12) in Table 5 (majority baseline), the MAE was calculated by assuming $X_j = 0.456$, which is the average ASR accuracy, for all j. Its classification accuracy is the majority baseline; that is, all interpretation results are regarded as incorrect.

4.4 Experimental Results

The results are shown in Table 5. First, we can see that the classification accuracies for Conditions (1) to (6) are high because the ASR confidence measure (CM) works well (Table 3). The MAEs are also small, which means the outputs of the logistic regression functions are good indicators of the reliability of the interpretation result.

Upon comparing Condition (6) with Conditions (1) to (5), we can see that the classification accuracies improve as a result of incorporating the user's utterance histories such as barge-in rates and ASR accuracies. Table 6 lists p-values of the differences when the barge-in rate and the estimated ASR accuracy were used in addition to the CM. The significance test was based on the McNemar test. As shown in the table, all the differences were statistically significant ($p < 0.01$). That is, it was experimentally shown that these utterance histories of users are different information sources from those of single utterances and that they contribute to improving the classification accuracy even when used together with ASR confidence measures. The relative improvement in the error reduction rate was 15.2% between Conditions (2) and (6), that is, by adding the barge-in rate and the estimated ASR accuracy, both of which can be obtained without manual labeling.

Table 5: Best classification accuracy for each condition and optimal window width

Conditions (features used)	Window width	Classification accuracy (%)	MAE
(1) CM + barge-in rate + actual ASR acc.	w=40	92.6	0.112
(2) CM + barge-in rate + estimated ASR acc	w=30	92.2	0.119
(3) CM + actual ASR acc.	-	91.7	0.121
(4) CM + barge-in rate	w=30	91.6	0.126
(5) CM + estimated ASR acc.	-	91.2	0.128
(6) CM	-	90.8	0.134
(7) barge-in rate + actual ASR acc.	w=50	80.0	0.312
(8) barge-in rate + estimated ASR acc.	w=50	77.7	0.338
(9) actual ASR acc.	-	72.8	0.402
(10) barge-in rate	w=30	71.8	0.404
(11) estimated ASR acc.	-	57.6	0.431
(12) majority baseline	-	54.4	0.496

CM: confidence measure
MAE: mean absolute error

Table 6: Results of significance test

Condition pair	p-value
(2) vs (4)	0.00066
(2) vs (5)	0.00003
(4) vs (6)	0.00017
(5) vs (6)	0.00876

Figure 4 shows the results in more detail; the classification accuracies for Conditions (1), (2), (4), and (6) are shown for various window widths. Under Condition (6), the classification accuracy does not depend on the window width because the barge-in rate is not used. Under Conditions (1), (2), and (4), the accuracies depend on the window width for the barge-in rate and are highest when the width is 30 or 40. These results show the effectiveness of the window, which indicates that temporal changes in user behaviors should be taken into consideration, and match those of our earlier reports (Komatani et al., 2008; Komatani and Rudnicky, 2009): the user's utterance history becomes effective after he/she uses the system about ten times because the average number of utterances per dialogue is around five.

By comparing Conditions (2) and (4), we can see that the classification accuracy improves after adding the estimated ASR accuracy to Condition (4). This shows that the estimated ASR accuracy also contributes to improving the classification accuracy. By comparing Conditions (1) and (2), we can see that Condition (1), in which the actual ASR accuracy is used, outperforms Condition (2), in which the estimated one is used. This suggests that the classification accuracy, whose upper limit is Condition (1), can be improved by making the ASR accuracy estimation shown in Section 3.2 more accurate.

5 Conclusion

We described a method of detecting interpretation errors of barge-in utterances by exploiting the utterance histories of individual users, such as their barge-in rate and ASR accuracy. The estimated ASR accuracy as well as the barge-in rate and the ASR confidence measure is obtainable online. Thus, the detection method does not require manual labeling. We showed through experiments that the utterance history of each user is helpful for detecting interpretation errors even when the ASR confidence measure is used.

The proposed method is effective in systems that are repeatedly used by the same user over 10 times, as indicated by the results of Figure 4. It is also assumed that the user's ID is known (we used their telephone number). The part of our method that estimates the user's ASR accuracy assumes that the system's dialogue strategy is to make explicit confirmations about every utterance by the user and that all affirmative and negative responses followed by explicit confirmations are correctly recognized. Our future work will attempt to reduce or remove these assumptions and to enhance the generality of our method. The experimental

result was shown only in the Kyoto City Bus domain, in which dialogues were rather well structured. Experimental evaluations in other domains will assure the generality.

Acknowledgments

We are grateful to Prof. Tatsuya Kawahara of Kyoto University who led the Kyoto City Bus Information System project. The evaluation data used in this study was collected during the project. This research was partly supported by Grants-in-Aid for Scientific Research (KAKENHI).

References

Dan Bohus and Alexander Rudnicky. 2007. Implicitly-supervised learning in spoken language interfaces: an application to the confidence annotation problem. In *Proc. SIGdial Workshop on Discourse and Dialogue*, pages 256–264.

Timothy J. Hazen, Theresa Burianek, Joseph Polifroni, and Stephanie Seneff. 2000. Integrating recognition confidence scoring with language understanding and dialogue modeling. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 1042–1045, Beijing, China.

Kazunori Komatani and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. Int'l Conf. Computational Linguistics (COLING)*, pages 467–473.

Kazunori Komatani and Alexander I. Rudnicky. 2009. Predicting barge-in utterance errors by using impricitly-supervised asr accuracy and barge-in rate per user. In *Proc. ACL-IJCNLP*, pages 89–92.

Kazunori Komatani, Shinichi Ueno, Tatsuya Kawahara, and Hiroshi G. Okuno. 2005. User modeling in spoken dialogue systems to generate flexible guidance. *User Modeling and User-Adapted Interaction*, 15(1):169–183.

Kazunori Komatani, Tatuya Kawahara, and Hiroshi G. Okuno. 2007. Analyzing temporal transition of real user's behaviors in a spoken dialogue system. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 142–145.

Kazunori Komatani, Tatuya Kawahara, and Hiroshi G. Okuno. 2008. Predicting asr errors by exploiting barge-in rate of individual users for spoken dialogue systems. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pages 183–186.

Akinobu Lee, Keisuke Nakamura, Ryuichi Nisimura, Hiroshi Saruwatari, and Kiyohiro Shikano. 2004. Noice robust real world spoken dialogue system using GMM based rejection of unintended inputs. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 173–176.

Gina-Anne Levow. 2003. Learning to speak to a spoken language system: Vocabulary convergence in novice users. In *Proc. 4th SIGdial Workshop on Discourse and Dialogue*, pages 149–153.

Diane J. Litman, Marilyn A. Walker, and Michael S. Kearns. 1999. Automatic detection of poor speech recognition at the dialogue level. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 309–316.

Antoine Raux, Dan Bohus, Brian Langner, Alan W. Black, and Maxine Eskenazi. 2006. Doing research on a deployed spoken dialogue system: One year of Let's Go! experience. In *Proc. Int'l Conf. Spoken Language Processing (INTERSPEECH)*.

Richard C. Rose and Hong Kook Kim. 2003. A hybrid barge-in procedure for more reliable turn-taking in human-machine dialog systems. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 198–203.

Katsuhito Sudoh and Mikio Nakano. 2005. Post-dialogue confidence scoring for unsupervised statistical language model training. *Speech Communication*, 45:387–400.

Marilyn Walker, Irene Langkilde, Jerry Wright, Allen Gorin, and Diane Litman. 2000. Learning to predict problematic situations in a spoken dialogue system: Experiments with How May I Help You? In *Proc. North American Chapter of Association for Computational Linguistics (NAACL)*, pages 210–217.

Dialogue Act Modeling in a Complex Task-Oriented Domain

Kristy Elizabeth Boyer | **Eun Young Ha** | **Robert Phillips*** | **Michael D. Wallis*** | **Mladen A. Vouk** | **James C. Lester**

Department of Computer Science, North Carolina State University
Raleigh, North Carolina, USA

*Dual affiliation with Applied Research Associates, Inc.
Raleigh, North Carolina, USA

{keboyer, eha, rphilli, mdwallis, vouk, lester}@ncsu.edu

Abstract

Classifying the dialogue act of a user utterance is a key functionality of a dialogue management system. This paper presents a data-driven dialogue act classifier that is learned from a corpus of human textual dialogue. The task-oriented domain involves tutoring in computer programming exercises. While engaging in the task, students generate a task event stream that is separate from and in parallel with the dialogue. To deal with this complex task-oriented dialogue, we propose a vector-based representation that encodes features from both the dialogue and the hierarchically structured task for training a maximum likelihood classifier. This classifier also leverages knowledge of the hidden dialogue state as learned separately by an HMM, which in previous work has increased the accuracy of models for predicting tutorial moves and is hypothesized to improve the accuracy for classifying student utterances. This work constitutes a step toward learning a fully data-driven dialogue management model that leverages knowledge of the user-generated task event stream.

1 Introduction

Two central challenges for dialogue systems are interpreting user utterances and selecting system dialogue moves. Recent years have seen an increased focus on data-driven techniques for addressing these challenging tasks (Bangalore et al., 2008; Frampton & Lemon, 2009; Hardy et al., 2006; Sridar et al., 2009; Young et al., 2009). Much of this work utilizes dialogue acts, built on the notion of speech acts (Austin, 1962), which provide a valuable intermediate representation that can be used for dialogue management.

Data-driven approaches to dialogue act interpretation have included models that take into account a variety of lexical, syntactic, acoustic, and prosodic features for dialogue act tagging (Sridhar et al., 2009; Stolcke et al., 2000). In task-oriented domains, recent work has approached dialogue act classification by learning dialogue management models entirely from human-human corpora (Bangalore et al., 2008; Chotimongkol, 2008; Hardy et al., 2006). Our work adopts this approach for a corpus of human-human dialogue in a task-oriented tutoring domain. Unlike the majority of task-oriented domains that have been studied to date, our domain involves the separate creation of a persistent artifact, in our case a computer program, by the user during the course of the dialogue. Our corpus consists of human-human textual dialogue utterances and a separate, parallel stream of user-generated task actions. We utilize structural features including task/subtask, speaker, and hidden dialogue state along with lexical and syntactic features to interpret user (student) utterances.

This paper makes three contributions. First, it addresses representational issues in creating a dialogue model that integrates task actions with hierarchical task/subtask structure. The task is captured within a separate synchronous event stream that exists in parallel with the dialogue. Second, this paper explores the performance of dialogue act classifiers using different lexical/syntactic and structural feature sets. This comparison includes one model trained entirely on lexical/syntactic features, an important step toward robust unsupervised dialogue act tagging

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 297–305,
The University of Tokyo, September 24-25, 2010.

(Sridhar et al., 2009). Finally, it investigates whether the addition of HMM and task/subtask features improves the performance of the dialogue act classifiers. The findings support this hypothesis for three student dialogue moves, each with important implications for tutorial dialogue.

2 Related Work

A variety of modeling approaches have been investigated for statistical dialogue act classification, including sequential approaches and vector-based classifiers. Sequential approaches typically formulate dialogue as a Markov chain in which an observation depends on a finite number of preceding observations. HMM-based approaches make use of the Markov assumption in a doubly stochastic framework that allows fitting optimal dialogue act sequences using the Viterbi algorithm (Rabiner, 1989; Stolcke et al., 2000). Like this work, the approach reported here adopts a first-order Markov formulation to train an HMM on sequences of dialogue acts, but the prediction of this HMM is subsequently encoded in a feature vector for training a vector-based classifier.

Vector-based approaches, such as maximum entropy modeling, also frequently take into account both lexical/syntactic and structural features. Lexical and syntactic cues are extracted from local utterance context, while structural features involve longer dialogue act sequences and, in task-oriented domains, task/subtask history. Work by Bangalore et al. (2008) on learning the structure of human-human dialogue in a catalogue-ordering domain (also extended to the Maptask and Switchboard corpora) utilizes features including words, part of speech tags, supertags, and named entities, and structural features including dialogue acts and task/subtask labels. In order to perform incremental decoding of dialogue acts and task/subtask structure, they take a greedy approach that does not require the search of complete dialogue sequences. Our work also accomplishes left-to-right incremental interpretation with a greedy approach. Our feature vectors differ from the aforementioned work slightly with respect to lexical/syntactic features and notably in the addition of a set of structural features generated by a separately trained HMM, as described in Section 4.2.

Recent work has explored the use of lexical, syntactic, and prosodic features for online dialogue act tagging (Sridhar et al., 2009); that work explores the notion that structural (history) features could be omitted altogether from incremental left-to-right decoding, resulting in computationally inexpensive and robust dialogue act classification. Although our textual dialogue does not feature prosodic cues, we report on the use of lexical/syntactic features alone to perform dialogue act classification, a step toward a fully unsupervised approach.

Like Bangalore et al. (2008), we treat task structure as an integral part of the dialogue model. Other work that has taken this approach includes the Amitiés project, in which a dialogue manager for a financial domain was derived entirely from a human-human corpus (Hardy et al., 2006). The TRIPS dialogue system also closely integrated task and dialogue models, for example, by utilizing the task model to facilitate indirect speech act interpretation (Allen et al., 2001). Work on the Maptask corpus has modeled task structure in the form of conversational games (Wright Hastie et al., 2002). Recent work in task-oriented domains has focused on learning task structure with unsupervised approaches (Chotimongkol, 2008). Emerging unsupervised methods, such as for detecting actions in multi-party discourse, also implicitly capture a task structure (Purver et al., 2006).

Our domain differs from the task-oriented domains described above in that our dialogues center on the user creating a persistent artifact of intrinsic value through a separate, synchronous stream of task actions. To illustrate, consider a catalogue-ordering task in which one subtask is to obtain the customer's name. The fulfillment of this subtask occurs entirely through the dialogue, and the resulting artifact (a completed order) is produced by the system. In contrast, our task involves the user constructing a solution to a computer programming problem. The fulfillment of this task occurs partially in the dialogue through tutoring, and partially in a separate synchronous stream of user-driven task actions about which the tutor must reason. The stream of user-driven task actions produces an artifact of value in itself (a functioning computer program), and that artifact is the subject of much of the dialogue. We propose a representation that integrates task actions and dialogue acts from these streams into a shared vector-based representation, and we investigate the use of the resulting structural, lexical, and syntactic features for dialogue act classification.

3 Corpus and Annotation

The corpus was collected during a controlled human-human tutoring study in which tutors and students worked through textual dialogue to solve an introductory computer programming problem. The dialogues were effective: on average, students exhibited significant learning and self-confidence gains (Boyer et al., 2009).

The corpus contains 48 dialogues each with a separate, synchronous task event stream as depicted in Excerpt 1 of the appendix. There is exactly one dialogue (tutoring session) per student. The corpus captures approximately 48 hours of dialogue and contains 1,468 student utterances and 3,338 tutor utterances. Because the dialogue was textual, utterance segmentation consisted of splitting at existing sentence boundaries when more than one dialogue act was present in the utterance. This segmentation was conducted manually by the principal dialogue act annotator.[1]

The corpus was manually annotated with dialogue act labels and task/subtask features. Lexical and syntactic features were extracted automatically. The remainder of this section describes the manual annotation.

3.1 Dialogue Act Annotation

The dialogue act annotation scheme was inspired by schemes for conversational speech (Stolcke et al., 2000) and task-oriented dialogue (Core & Allen, 1997). It was also influenced by tutoring-specific tagsets (Litman & Forbes-Riley, 2006). Inter-rater reliability for the dialogue act tagging on 10% of the corpus selected via stratified (by tutor) random sampling was κ=0.80. The dialogue act tags, their relative frequencies, and their individual kappa scores from manual annotation are displayed in Table 1.

3.2 Task Annotation

All task actions were generated by the student while implementing the solution to an introductory computer programming problem in Java. These task actions were recorded as a separate event stream in parallel with the dialogue corpus. This stream included 97,509 keystroke-level user task events, which were manually aggregated into task/subtask event clusters and annotated for subtask structure and then for correctness. A total of 3,793 aggregated student subtask actions were identified through manual annotation. The task annotation scheme is hierarchical, reflecting the nested nature of the subtasks. A subset of this task annotation scheme is depicted in Figure 1. In the models reported in this paper, the 66 leaves of the task/subtask hierarchy were encoded in the input feature vectors.

Table 1. Student dialogue acts

Student Dialogue Act	Rel. Freq.	Human κ
ACKNOWLEDGMENT (ACK)	.17	*.90*
REQUEST FOR FEEDBACK (RF)	.20	*.91*
EXTRA-DOMAIN (EX)	.08	*.79*
GREETING (GR)	.04	*.92*
UNCERTAIN FEEDBACK WITH ELABORATION (UE)	.01	*.53*
UNCERTAIN FEEDBACK (U)	.02	*.49*
NEGATIVE FEEDBACK WITH ELABORATION (NE)	.01	*.61*
NEGATIVE FEEDBACK (N)	.05	*.76*
POSITIVE FEEDBACK WITH ELABORATION (PE)	.02	*.43*
POSITIVE FEEDBACK (P)	.09	*.81*
QUESTION (Q)	.09	*.85*
STATEMENT (S)	.16	*.82*
THANKS (T)	.05	*1*

Each group of task events that occurred between dialogue utterances was tagged, possibly with many subtask labels, by a human judge. The judge aggregated the raw task keystrokes and tagged the task/subtask hierarchy for each cluster. (Please see Excerpt 1 in the appendix.) A second judge tagged 20% of the corpus in a reliability study for which one-to-one subtask identification was not enforced, an approach that was intended to give judges maximum flexibility to cluster task actions and subsequently apply the tags. All unmatched subtask tags were treated as disagreements. The resulting kappa statistic at the leaves was κ= 0.58. However, we also observe that the sequential nature of the subtasks within the larger task produces an ordinal relationship between subtasks. For example, in Figure 1, the "distance" between subtasks *1-a* and *1-b* can be thought of as "less than" the distance between subtasks *1-a* vs. *3-d* because those subtasks are farther from each other within the larger task. The weighted Kappa statistic (Artstein & Poesio, 2008) takes into account such an ordinal relationship and its implicit distance function. The weighted Kappa is

[1] Automatic segmentation is a challenging problem in itself and is left to future work.

$\kappa_{weighted}$=0.86, which indicates acceptable inter-rater reliability on the task/subtask annotation.

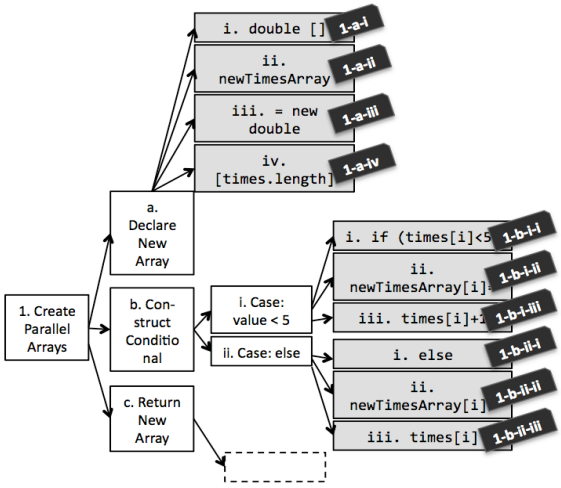


Figure 1. Portion of task annotation scheme

Along with its tag for hierarchical subtask structure, each task event was also judged for correctness according to the requirements of the task as depicted in Table 2. The agreement statistic for correctness was calculated for task events on which the two judges agreed on subtask tag. The resulting unweighted agreement statistic for correctness was κ=0.80.

Table 2. Task correctness labels

Label	Description
CORRECT	Fully satisfying the requirements of the learning task. Does not require tutorial remediation.
BUGGY	Violating the requirements of the learning task. Often requires tutorial remediation.
INCOMPLETE	Not violating, but not yet fully satisfying, the requirements of the learning task. May require tutorial remediation.
DISPREFERRED	Technically satisfying the requirements of the learning task, but not adhering to its pedagogical intentions. Usually requires tutorial remediation.

4 Features

The vector-based representation for training the dialogue act classifiers integrates several sources of features: lexical and syntactic features, and structural features that include dialogue act labels, task/subtask labels, and set of hidden dialogue state prediction features.

4.1 Lexical and Syntactic Features

Lexical and syntactic features were automatically extracted from the utterances using the Stanford Parser default tokenizer and part of speech (*pos*) tagger (De Marneffe et al., 2006). The parser created both phrase structure trees and typed dependencies for individual sentences. From the phrase structure trees, we extracted the top-most syntactic node and its first two children. In the case where an utterance consisted of more than one sentence, only the phrase structure tree of the first sentence was considered. Typed dependencies between pairs of words were extracted from each sentence. Individual word tokens in the utterances were further processed with the Porter Stemmer (Porter, 1980) in the NLTK package (Loper & Bird, 2004). The *pos* features were extracted in a similar way. Unigram and bigram word and *pos* tags were included for feature selection in the classifiers.

4.2 Structural Features

Structural features include the annotated dialogue acts, the annotated task/subtask labels, and attributes that represent the *hidden dialogue state*. Our previous work has found that a set of hidden dialogue states, which correspond to widely accepted notions of dialogue modes in tutoring, can be identified in an unsupervised fashion (without hand labeling of the modes) by HMMs trained on manually labeled dialogue acts and task/subtask features (Boyer et al., 2009). These HMMs performed significantly better than bigram models for predicting human *tutor* moves (Boyer et al., 2010), which indicates that the hidden dialogue state leveraged by the HMMs has predictive value even in the presence of "true" (manually annotated) dialogue act labels. Therefore, we hypothesized that an HMM could also improve the performance of models to classify student dialogue acts. To explore this hypothesis, we trained an HMM utilizing the methodology described in (Boyer et al., 2009) and used it to generate hidden dialogue state predictions in the form of a probability distribution over possible user utterances at each step in the dialogue. This set of stochastic features was subsequently passed to the classifier as part of the input vector (Figure 2).

4.3 Input Vectors

The features were combined into a shared vector-based representation for training the classifier. As depicted in Table 3, the components of the

feature vector include binary existence vectors for lexical and syntactic features for the current (target) utterance as well as for three utterances of left context (this left context may include both tutor and student utterances, which are distinguished by a separate indicator for the speaker). The task/subtask and correctness history features encode the separate stream of task events. There is no one-to-one correspondence between these history features and the left-hand dialogue context, because several task events could have occurred between a pair of dialogue events (or vice versa). This distinction is indicated in the table by the representation of dialogue time steps as [*t, t-1,…*] and task history steps as [*task(t), task(t-1),…*]. In total, the feature vectors included 11,432 attributes that were made available for feature selection.

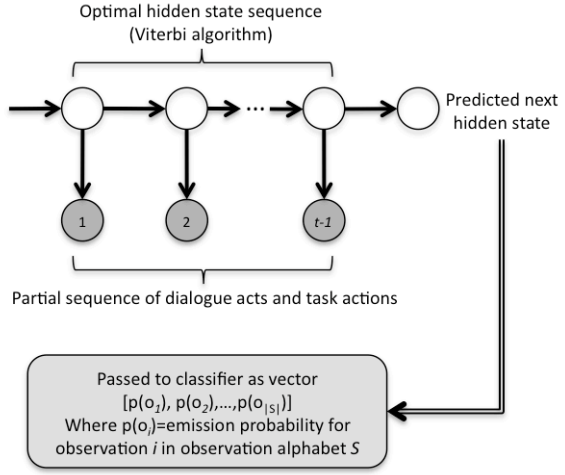


Figure 2. Generation of hidden dialogue state prediction features

5 Experiments

This section describes the learning of maximum likelihood vector-based models for classification of user dialogue acts. In addition to investigating the accuracy of the overall model, we also performed experiments regarding the utility of feature types for discriminating between particular dialogue acts of interest.

The classifiers are based on logistic regression, which learns a discriminant for each pair of dialogue acts by assigning weights in a maximum likelihood fashion.[2] The logistic regression models were learned using the Weka machine learning toolkit (Hall et al., 2009). For feature selection, we performed attribute subset evaluation with a best-first approach that greedily searched the space of possible features using a hill climbing approach with backtracking. The prediction accuracy of the classifiers was determined through ten-fold cross-validation on the corpus, and the results below are presented in terms of prediction accuracy (number of correct classifications divided by total number of classifications) as well as by the kappa statistic, which adjusts for expected agreement by chance.

Table 3. Feature vectors

Feature vector *f*	Description
$[w_{t,1},\ldots w_{t,\|w\|}, p_{t,1},\ldots,p_{t,\|p\|}, d_{t,1},\ldots,d_{t,\|d\|}, s_{t,1},\ldots,s_{t,\|s\|}]$	Binary existence vector for word unigrams & bigrams, *pos* unigrams & bigrams, dependency types, and syntactic nodes for current target utterance *t*
$[w_{t-k,1},\ldots w_{t-k,\|w\|}, p_{t-k,1},\ldots,p_{t-k,\|p\|}, d_{t-k,1},\ldots,d_{t-k,\|d\|}, s_{t-k,1},\ldots,s_{t-k,\|s\|}]$ where $k=1,\ldots,3$	Binary existence vector for word unigrams & bigrams, *pos* unigrams & bigrams, dependency types, and syntactic nodes for three utterances of left context
$[p(o_1),\ldots,p(o_{\|S\|})]$	Probability distribution for emission symbols in predicted next hidden state as generated by HMM
$[da_{t-1}, da_{t-2}, da_{t-3}]$	Dialogue act left context
$[sp_{t-1}, sp_{t-2}, sp_{t-3}]$	Speaker label left context
$[tk_{task(t-1)}, tk_{task(t-2)}, tk_{task(t-3)}]$	Three steps of subtask history (each level of hierarchy represented as a separate feature)
$[c_{task(t-1)}, c_{task(t-2)}, c_{task(t-3)}]$	Three steps of task correctness history
pt	Indicator for whether the target utterance was immediately preceded by a task event

5.1 Overall Classification Task

The overall dialogue act classification model was trained to classify each utterance with respect to the thirteen dialogue acts (Table 1). For this task, the feature selection algorithm selected 63 attributes including some syntax, dependency, *pos*, and word attributes as well as dialogue act, speaker, and task/subtask features. No hidden dialogue state features or task correctness attributes were selected. The overall classification accuracy was 62.8%. This accuracy constitutes a 369% improvement over baseline chance of 17% (the relative frequency of the most frequently occurring dialogue act, ACK). An alternate nontrivial baseline is a bigram model on true dialogue acts (including speaker tags); this model's accuracy was 36.8%. The

[2] In general, the model that maximizes likelihood also maximizes entropy under the same constraints (Berger et al., 1996).

overall kappa for the full classifier was κ=.57. The confusion matrix for this model is depicted in Figure 3.

In addition to the classifier described above, we experimented with classifiers that used only the lexical and syntactic features of each utterance. This approach is of interest in part because it avoids the error propagation that can happen when a model relies on a series of its own previous classifications as features. The classifier that used only the set of lexical and syntactic features achieved a prediction accuracy of 60.2% and κ=.53 using 85 attributes.

GR	N	P	S	RF	Q	T	ACK	Ex	NE	PE	L	LE	
50	0	0	1	0	0	0	1	0	0	0	0	0	GR
0	19	6	13	2	2	0	5	2	1	0	6	2	N
0	5	52	37	1	1	0	18	3	0	1	1	1	P
0	6	21	145	3	9	0	15	7	0	4	4	0	S
0	2	1	11	232	23	0	6	5	0	1	0	1	RF
1	2	2	13	60	46	0	1	5	0	1	0	0	Q
0	0	1	3	0	0	60	3	1	1	0	0	1	T
0	0	7	19	4	0	2	195	4	0	0	2	0	ACK
0	1	4	24	12	3	1	16	40	0	1	0	0	Ex
0	1	1	9	0	1	1	0	1	0	3	1	0	NE
0	2	4	13	0	2	0	1	1	1	4	2	2	PE
0	6	4	2	3	0	0	0	0	1	3	6	1	L
0	3	0	5	2	2	1	0	1	0	0	0	1	LE

Figure 3. Confusion matrix

5.2 Binary Dialogue Act Classification

In tutoring, some student dialogue acts are particularly important to identify because of their implications for the tutor's response or for the student model. For example, a student's REQUEST FOR FEEDBACK requires the tutor to assess the condition of the task, rather than to query the in-domain factual knowledge base. UNCERTAIN FEEDBACK is another dialogue act of high importance because identifying it allows the tutor to respond in an affectively advantageous way (Forbes-Riley & Litman, 2009).

To explore which features are useful for classifying particular dialogue acts, we constructed binary dialogue act classifiers, one for each dialogue act, by preprocessing the dialogue act labels from the set of thirteen down to TRUE or FALSE depending on whether the label of the utterance matched the target dialogue act for that specialized classifier. Table 4 displays the features that were selected for each binary classifier, along with the percent accuracy and kappa for each model. Note that for some dialogue acts the chance baseline is very high, and therefore even a model with high prediction accuracy achieves a low kappa.

As depicted in Table 4, for several dialogue act models, the feature selection algorithm retained subtask and HMM features.

Table 4. Binary DA classifiers

DA	# Features Selected		% Correct	Model κ
ACK	51	Lexical/syntax, HMM, DA history (preceding=S), speaker history (preceding=Tutor)	.933	*.75*
RF	42	Lexical/syntax, DA history, preceded by subtask	.905	*.72*
EX	57	Dependency, *pos*, word, HMM, DA history (preceding=EX), subtask	.939	*.45*
GR	11	Syntax, *pos*, word, DA (previous=EMPTY), speaker, subtask	.998	*.97*
UE	21	Dependency, *pos*, word, subtask	.991	*.33*
U	63	Syntax, dependency, *pos*, word, HMM, subtask	.979	*.21*
NE	44	Dependency, *pos*, word, HMM, DA history (2 ago=UNCERTAIN), subtask	.987	*0*
N	83	Lexical/syntax, DA history, subtask	.966	*.76*
PE	90	Dependency, *pos*, word, HMM, subtask	.976	*.10*
P	110	Dependency, *pos*, word, HMM, DA history (previous=REQUEST FEEDBACK)	.945	*.58*
Q	43	Syntax, dep, *pos*, word, HMM, subtask	.940	*.60*
S	92	Syntax, *pos*, word, HMM, DA history (previous=EMPTY or Q)	.901	*.57*
T	29	Syntax, *pos*, word, DA history (previous=POSITIVE) (3 ago=POSITIVE)	.992	*.92*

In an experiment to quantify the utility of these features, it was found that for many dialogue acts, a binary dialogue act classifier that was trained using only lexical and syntactic features achieved the same or better classification accuracy than the model that was given all features (Figure 4). For comparison, the modified baseline model used the last three true dialogue acts (with speaker tags); this model achieved better than chance for four dialogue acts and achieved nearly as well as the full model for GREETING (GR). The models that were given all possible features for selection outperformed the lexical/syntax-only model for seven of the thirteen dialogue acts (GREETING (GR), REQUEST FOR FEEDBACK (RF), POSITIVE FEEDBACK (P), POSITIVE ELABORATED FEEDBACK (PE), UNCERTAIN ELABORATED FEEDBACK (UE), NEGATIVE FEEDBACK (N), and EXTRA-DOMAIN (EX)); however, it should be noted that none of these differences in performance is statistically reliable at the p=0.05 level.

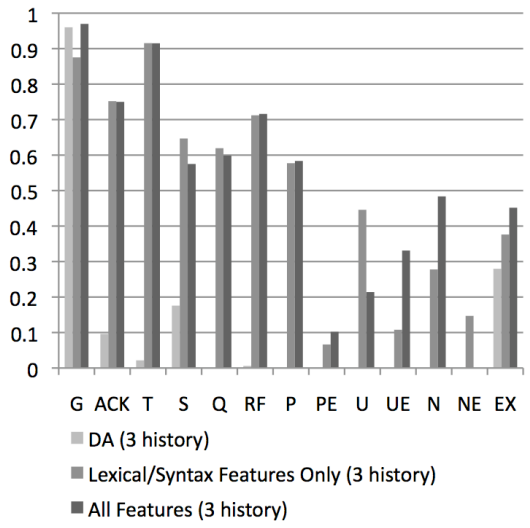


Figure 4. Kappa for binary DA classifiers by features available for selection

6 Discussion

We have presented a maximum likelihood classifier that assigns dialogue act labels to user utterances from a corpus of human-human tutorial dialogue given a set of lexical, syntactic, and structural features. Overall, this classifier achieved 62.8% accuracy in ten-fold cross-validation on the corpus. This performance is on par with other automatic dialogue act tagging models, both sequential and vector-based, in task-oriented domains that do not feature complex, user-driven parallel tasks.

In a catalogue ordering domain with an integrated task and dialogue model, Bangalore et al. (2009) report 75% classification accuracy for user utterances using a maximum entropy classifier, a 275% improvement over baseline. Poesio & Mikheev (1998) report 54% classification accuracy by utilizing conversational game structure and speaker changes in the Maptask corpus, an improvement of 170% over baseline. Recent work on Maptask reports a classification accuracy of 65.7% using local utterance (such as lexical/syntactic) features alone, with prosodic cues yielding further slight improvement (Sridhar et al., 2009). This classifier is analogous to our lexical/syntactic feature model, which achieved 60.2% accuracy.

The results of these models demonstrate that, consistent with the findings in other task-oriented domains, lexical/syntactic features are highly useful for classifying student dialogue moves in this complex task-oriented domain. Models trained using those lexical/syntactic features performed almost universally better (with the exception of the binary classifier for GREETING) than models that were given the same left context of true dialogue act tags.

It was hypothesized that leveraging both the hidden dialogue state and hierarchical subtask features would improve the performance of the classifiers. There is some evidence that the subtask structure was helpful for the overall classifier; however, no HMM features were kept during feature selection for the overall model. Of the binary models, approximately half performed better than the overall model in terms of true positive rate; of those, three did so by including HMM or task/subtask features among the selected attributes to differentiate different tones of student feedback. However, this difference in performance was not statistically reliable. This finding suggests that, given lexical and syntactic features which are strong predictors of dialogue acts, the hidden dialogue state as captured by an an HMM may not contribute significantly to the dialogue act classification task.

7 Conclusion and Future Work

Dialogue modeling for complex task-oriented domains poses significant challenges. An effective dialogue model allows systems to detect user dialogue acts so that they can respond in a manner that maximizes the chance of success. Experiments with the data-driven classifiers presented in this paper demonstrate that lexical/syntactic features can effectively classify student dialogue acts in the task-oriented tutoring domain. For POSITIVE, NEGATIVE, and UNCERTAIN ELABORATED student feedback acts, which play a key role in tutorial dialogue system, the addition of hidden dialogue state features (as learned by an HMM) and task/subtask features (annotated manually) improve classification accuracy, but not statistically reliably.

The overarching goal of this work is to create a data-driven tutorial dialogue system that learns its behavior from corpora of effective human tutoring. The dialogue act classification models reported here constitute an important step toward that goal, by integrating the dialogue stream with a parallel user-driven task event stream. The next generation of data-driven systems should leverage models that capture the rich interplay between dialogue and task. Future work will focus on data-driven approaches to task recognition and tutorial planning. Additionally, as dialogue system research addresses

increasingly complex task-oriented domains, it becomes increasingly important to investigate unsupervised approaches for dialogue act classification and task recognition.

Acknowledgements. This work is supported in part by the North Carolina State University Department of Computer Science and the National Science Foundation through a Graduate Research Fellowship and Grants CNS-0540523, REC-0632450 and IIS-0812291. Any opinions, findings, conclusions, or recommendations expressed in this report are those of the participants, and do not necessarily represent the official views, opinions, or policy of the National Science Foundation.

References

Allen, J., Ferguson, G., & Stent, A. (2001). An architecture for more realistic conversational systems. *Proceedings of the IUI,* 1-8.

Artstein, R., & Poesio, M. (2008). Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4), 555-596.

Austin, J. L. (1962). *How to do things with words*. Oxford: Oxford University Press.

Bangalore, S., Di Fabbrizio, G., & Stent, A. (2008). Learning the structure of task-driven human-human dialogs. *IEEE Transactions on Audio, Speech, and Language Processing, 16*(7), 1249-1259.

Berger, A. L., Pietra, V. J. D., & Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Comp. Ling., 22*(1), 71.

Boyer, K. E., Phillips, R., Ha, E. Y., Wallis, M. D., Vouk, M. A., & Lester, J. C. (2009). Modeling dialogue structure with adjacency pair analysis and hidden markov models. *Proceedings of NAACL-HLT, Short Papers, 49-52.*

Boyer, K. E., Phillips, R., Ha, E. Y., Wallis, M. D., Vouk, M. A., & Lester, J. C. (2010). Leveraging hidden dialogue state to select tutorial moves. *Proceedings of the 5th NAACL HLT Workshop on Innovative use of NLP for Building Educational Applications,* Los Angeles, California.

Chotimongkol, A. (2008). *Learning the structure of task-oriented conversations from the corpus of in-domain dialogs.* (Unpublished Ph.D. Dissertation). Carnegie Mellon University School of Computer Science.

Core, M., & Allen, J. (1997). Coding dialogs with the DAMSL annotation scheme. *AAAI Fall Symposium on Communicative Action in Humans and Machines,* 28–35.

De Marneffe, M. C., MacCartney, B., & Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. *Proceedings of LREC,* Genoa, Italy.

Forbes-Riley, K., & Litman, D. (2009). Adapting to student uncertainty improves tutoring dialogues. *Proceedings of AIED,* 33-40.

Frampton, M., & Lemon, O. (2009). Recent research advances in reinforcement learning in spoken dialogue systems. *The Knowledge Engineering Review, 24*(4), 375-408.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. (2009). The WEKA data mining software: An update. *SIGKDD Explorations, 11*(1)

Hardy, H., Biermann, A., Inouye, R. B., McKenzie, A., Strzalkowski, T., Ursu, C., Webb, N., & Wu, M. (2006). The Amitiés system: Data-driven techniques for automated dialogue. *Speech Comm., 48*(3-4), 354-373.

Litman, D., & Forbes-Riley, K. (2006). Correlations between dialogue acts and learning in spoken tutoring dialogues. *Natural Language Engineering, 12*(2), 161-176.

Loper, E., & Bird, S. (2004). NLTK: The natural language toolkit. *Proceedings of the ACL Demonstration Session,* Barcelona, Spain. 214-217.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130-137.

Purver, M., Kording, K. P., Griffiths, T. L., & Tenenbaum, J. B. (2006). Unsupervised topic modelling for multi-party spoken discourse. *Proceedings of the ACL,* Sydney, Australia. *, 44*(1) 17.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257-286.

Sridhar, V. K. R., Bangalore, S., & Narayanan, S. (2009). Combining lexical, syntactic and prosodic cues for improved online dialog act tagging. *Computer Speech & Language, 23*(4), 407-422.

Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C., & Meteer, M. (2000). Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comp. Ling., 26*(3), 339-373.

Wright Hastie, H., Poesio, M., & Isard, S. (2002). Automatically predicting dialogue structure using prosodic features. *Speech Communication, 36*(1-2), 63-79.

Young, S., Gasic, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., & Yu, K. (2009). The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language, 24*(2), 150-174.

Appendix

Time Stamp	Dialogue Stream	Task Stream	
2008-04-11 18:23:45	Student: so do i have to manipulate the array this time? [Q]		
2008-04-11 18:23:53	Tutor: this time, we need to do two things [S]		
2008-04-11 18:24:02	Tutor: first, we need to create a new array to hold the changed values [S]		
2008-04-11 18:24:28		i	
2008-04-11 18:24:28		n	**1-a-i**
2008-04-11 18:24:28		t	**BUGGY**
2008-04-11 18:24:28		\sp	
2008-04-11 18:24:35		\del	
2008-04-11 18:24:36		\sp	
2008-04-11 18:24:36		d	
2008-04-11 18:24:36		o	
2008-04-11 18:24:36		u	
2008-04-11 18:24:36		b	**1-a-i**
2008-04-11 18:24:37		l	**CORRECT**
2008-04-11 18:24:37		e	
2008-04-11 18:24:37		\sp	
2008-04-11 18:24:39		[]	
2008-04-11 18:24:40		\sp	
2008-04-11 18:24:42		n	
2008-04-11 18:24:42		e	
2008-04-11 18:24:42		w	
2008-04-11 18:24:43		\sp	
2008-04-11 18:24:44		\del	
2008-04-11 18:24:45		T	
2008-04-11 18:24:46		\del	**1-a-ii**
2008-04-11 18:24:54		T	**CORRECT**
2008-04-11 18:24:54		i	
2008-04-11 18:24:54		m	
2008-04-11 18:24:54		e	
2008-04-11 18:24:54		s	
2008-04-11 18:24:55		3	
2008-04-11 18:24:57		;	
2008-04-11 18:25:11	Student: good? [RF]		
2008-04-11 18:25:14	Tutor: good so far, yes [PF]		
2008-04-11 18:25:29	Student: so now i have to change parts of the times array right? [Q]		
2008-04-11 18:25:34	Tutor: not quite [LF]		
2008-04-11 18:25:57	Tutor: So, when you create a new object, like a String for example, you'd say something like String s = new String() [S]		
2008-04-11 18:25:59	Tutor: right? [AQ]		
2008-04-11 18:26:06	Student: right [P]		
2008-04-11 18:26:14	Tutor: arrays are similar [S]		

Excerpt 1. Parallel synchronous dialogue and task event streams with annotations. (Note tutor dialogue acts: AQ=ASSESSING QUESTION, LF=LUKEWARM FEEDBACK, PF=POSITIVE FEEDBACK)

Hand Gestures in Disambiguating Types of *You* Expressions in Multiparty Meetings

Tyler Baldwin
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
baldwin96@cse.msu.edu

Joyce Y. Chai
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
jchai@cse.msu.edu

Katrin Kirchhoff
Department of Electrical Engineering
University of Washington
Seattle, WA, USA
katrin@ee.washington.edu

Abstract

The second person pronoun *you* serves different functions in English. Each of these different types often corresponds to a different term when translated into another language. Correctly identifying different types of *you* can be beneficial to machine translation systems. To address this issue, we investigate disambiguation of different types of *you* occurrences in multiparty meetings with a new focus on the role of hand gesture. Our empirical results have shown that incorporation of gesture improves performance on differentiating between the generic use of *you* (e.g., refer to people in general) and the referential use of *you* (e.g., refer to a specific person or a group of people). Incorporation of gesture can also compensate for limitations in automated language processing (e.g., reliable recognition of dialogue acts) and achieve comparable results.

1 Introduction

The second person pronoun *you* is one of the most prevalent words in conversation and it serves several different functions (Meyers, 1990). For example, it can be used to refer to a single addressee (i.e., the *singular* case) or multiple addressees (i.e., the *plural* case). It can also be used to represent people in general (i.e., the *generic* case) or be used idiomatically in the phrase "you know".

For machine translation systems, these different types of *you* often correspond to different translations in another language. For example, in German, there are different second-person pronouns for singular vs. plural *you* (viz. *du* vs. *ihr*); in addition there are different forms for formal vs. informal forms of address (*du* vs. *Sie*) and for the generic use (*man*). The following examples demonstrate different translations of *you* from English (EN) into German (DE):

- Generic *you*
  EN: Sometimes **you** have meetings where the decision is already taken.
  DE: Manchmal hat **man** Meetings wo die Entscheidung schon gefallen ist.
- Singular *you*:
  EN: Do **you** want an extra piece of paper?
  DE: Möchtest **du** noch ein Blatt Papier?
- Plural *you:*
  EN: Hope **you** are all happy!
  DE: Ich hoffe, **ihr** seid alle zufrieden!

These examples show that correctly identifying different types of *You* plays an important role in the correct translation of *you* in different context.

To address this issue, this paper investigates the role of hand gestures in disambiguating different usages of *you* in multiparty meetings. Although identification of *you* type has been investigated before in the context of addressee identification (Gupta et al., 2007b; Gupta et al., 2007a; Frampton et al., 2009; Purver et al., 2009), our work here focuses on two new angles. First, because of our different application on machine translation, rather than processing *you* at an utterance level to identify addressee, our work here concerns each occurrence of *you* within each utterance. Second and more importantly, our work investigates the role of corresponding hand gestures in the disambiguation process. This aspect has not been examined in previous work.

When several speakers are conversing in a situated environment, they often overtly gesture at one another to help manage turn order or explicitly direct a statement toward a particular participant (McNeill, 1992). For example, consider the following snippet from a multiparty meeting:

A: "Why is that?"
B: "Because, um, based on what ev-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 306–313,
The University of Tokyo, September 24-25, 2010.

erybody's saying, right, [*gestures at Speaker D*] you want something simple. You [*gestures at Speaker C*]want basic stuff and [*gestures at Speaker A*] you want something that is easy to use. Speech recognition might not be the simplest thing."

The use of gesture in this example indicates that each instance of the pronoun *you* is intended to be referential, and gives some indication of the indented addressee. Without the aid of gesture, it would be difficult even for a human listener to be able to interpret each instance correctly.

Therefore, we conducted an empirical study on several meeting segments from the AMI meeting corpus. We formulated our problem as a classification problem for each occurrence of *you*, whether it is a *generic*, *singular*, or *plural* type. We combined gesture features with several linguistic and discourse features identified by previous work and evaluated the role of gesture in two different settings: (1) a two stage classification that first differentiates the *generic* type from the *referential* type and then within the *referential* type distinguishes *singular* and *plural* usages; (2) a three way classification between *generic*, *singular*, or *plural* types. Our empirical results have shown that incorporation of gesture improves performance on differentiating between the *generic* and the *referential* type. Incorporation of gesture can also compensate for limitations in automated language processing (e.g., reliable recognition of dialogue acts) and achieve comparable results. These findings have important implications for machine translation of *you* expressions from multiparty meetings.

2 Related Work

Psychological research on gesture usage in human-human dialogues has shown that speakers gesture for a variety of reasons. While speakers often gesture to highlight objects related to the core conversation topic (Kendon, 1980), they also gesture for dialogue management purposes (Bavelas et al., 1995). While not all of the gestures produced relate directly to the resolution of the word *you*, many of them give insight into which participant is being addressed, which has a close correlation with *you* resolution. Our investigation here is closely related to two areas of previous work: addressee identification based on *you* and the use of gestures in coreference resolution.

Addressee Identification. Disambiguation of *you* type in the context of addressee identification has been examined in several papers in recent years. Gupta et. al. (2007b) examined two-party dialogues from the Switchboard corpus. They modeled the problem as a binary classification problem of differentiating between generic and referential usages (referential usages include the singular and plural types). This work has identified several important linguistic and discourse features for this task (which was used and extended in later work and our work here). Later work by the same group (Gupta et al., 2007a) examined the same problem on multiparty dialogue data. They made adjustments to their previous methods by removing some oracle features from annotation and applying simpler and more realistic features. A recent work (Frampton et al., 2009) has examined both the generic vs. referential and singular vs. plural classification tasks. A main difference is that this work incorporated gaze feature information in both classification tasks (gaze features are commonly used in addressee identification). More recent work (Purver et al., 2009) discovered that large gains in performance can be achieved by including n-gram based features. However, they found that many of the most important n-gram features were topic specific, and thus required training data consisting of meetings about the same topic.

Gestures in Coreference Resolution. Eisenstein and Davis (2006; 2007) examined coreference resolution on a corpus of speaker-listener pairs in which the speaker had to describe the workings of a mechanical device to the listener, with the help of visual aids. In this gesture heavy dataset, they found gesture data to be helpful in resolving references. In our previous work (2009), we examined gestures for the identification of coreference on multparty meeting data. We found that gestures only provided limited help in the coreference identification task. Given the nature of the meetings under investigation, although gestures have not been shown to be effective in general, they are potentially helpful in recognizing whether two linguistic expressions refer to a same participant.

Compared to these two areas of earlier work, our investigation here has two unique aspects. First, as mentioned earlier, previous work on addressee identification focused the problem at the

utterance level. Because the goal was to find the addressee of an utterance, the assumption was that all instances of *you* in an utterance were of the same type. However, since several instances of *you* in the same utterance may translate differently, we instead examine the classification task at the instance level. Second, our work here specifically investigates the role of gestures in disambiguation of different types of *you*. This aspect has not been examined in previous work.

3 Data

The dataset used in our investigation was the AMI meeting corpus (Popescu-Belis and Estrella, 2007), the same corpus used in previous work (Gupta et al., 2007a; Frampton et al., 2009; Purver et al., 2009; Baldwin et al., 2009). The AMI meeting corpus is a large publicly available corpus of multiparty design meetings. AMI meeting annotations contain manual speech transcriptions, as well as annotations of several additional modalities, such as focus of attention and head and hand gesture.

For this work, six AMI meeting segments (IS1008a, IS1008b, IS1008c, IS1008d, ES2008a, TS3005a) were used. These instances were chosen because they contained manual annotations of hand gesture data, which was not available for all AMI meeting segments. These six meeting segments were from AMI "scenario" meetings, in which meeting participants had a specific task of designing a hypothetical remote control.

All instances of the word *you* and its variants were manually annotated as either generic, singular, or plural. This produced a small dataset of 533 instances. Agreement between two human annotators was high ($\kappa = 0.9$). The distribution of *you* types is shown in Figure 1. The most prevalent type in our data set was the generic type, which accounted for 47% of all instances of *you* present. Of the two referential types, the singular type accounted for about 60% of the referential instances.

A total of 508 gestures are present in our data set. Table 1 shows the distribution of gestures. As shown, "non-communicative gestures", make up nearly half (46%) of the gestures produced. These are gestures that are produced without an overt communicative intent, such as idly tapping on the table. The other main categorization of gestures is "communicative gestures", which accounts for 45% of all gestures produced and is made up of the "pointing at participants", "pointing at objects", "interact with object", and "other communicative" gesture types from Table 1. A total of 17% of the gestures produced were pointing gestures that pointed to people, a type of gesture that would likely be helpful for *you* type identification. A small percentage of the gestures produced were not recorded by the meeting recording cameras (i.e., off camera), and thus are of unknown type.

4 Methodology

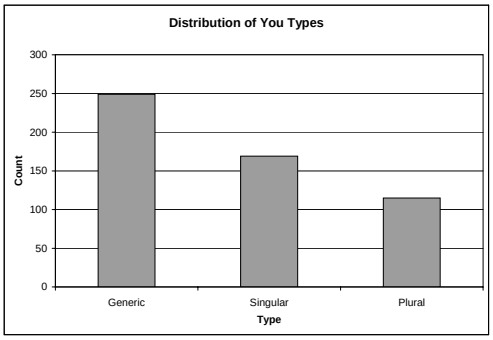
Our general methodology followed previous work and formulated this problem as a classification problem. We evaluated how gesture data may help *you* type identification using two different approaches: (1) two stage binary classification, and (2) a single three class classification problem. In two stage binary classification, we first attempt to distinguish between instances of *you* that are generic and those that are referential. We then take those cases that are referential and attempt to subdivide them into instances that are intended to refer to a single person and those that refer to several people.

Our feature set includes features used by Gupta et. al. (2007a) (Hereafter referred to as Gupta) and Frampton et. al. (2009) (Hereafter Frampton), as well as new features incorporating gestures. We summarize these features as follows.

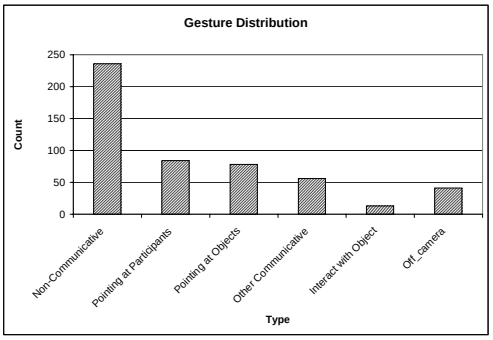
Sentential Features. We used several sentential features to capture important phrase patterns. Most of our sentential features were drawn from Gupta (2007a). These features captured the patterns "you guys", "you know", "do you" (and similar variants), "which you" (and variants), "if you", and "you hear" (and variants). Another sentential feature captured the number of times the word *you* appeared in the sentence. Additionally, other features captured sentence patterns not related to *you*, such as the presence of the words "I" and "we".

A few other sentential features were drawn from Frampton et. al. (2009). These include the pattern "<auxiliary> you" (a more general version of the "do you" feature) and a count of the number of total words in the utterances.

Part-of-Speech Features. Several features based on automatic part-of-speech tagging of the sentence containing *you* were used. Quality of automatic tagging was not assessed. From the tagged results, we extracted 5 features based on sentence



(a) Distribution of *You* types



(b) Distribution of gesture types

Figure 1: Data distributions

and tag patterns: whether or not the sentence that contained *you* also contained *I*, or *we* followed by a verb tag (3 separate features), and whether or not the sentence contains a comparative JJR (adjective) tag. All of these features were adapted from Gupta (2007a).

Dialog Act Features. We used the manually annotated dialogue act tags provided by the AMI corpus to produce our dialogue act features. Three dialogue act features were used: the dialogue act tag of the current sentence, the previous sentence, and the sentence prior to that. Dialog act tags were incorporated into the feature set in one of two different ways: 1) using the full tag set provided by the AMI corpus, and 2) using a binary feature recording if the dialogue act tag was of the elicit type. The latter way of dialogue act incorporation represents a simpler and more realistic treatment of dialogue acts.

Question Mark Feature. The question mark feature captures whether or not the current sentence ends in a question mark. This feature captures similar information to the elicit dialogue act tag and was used in Gupta as an automatically extractable replacement to the manually extracted dialogue act tags (2007a).

Backward Looking/Forward Looking Features. Several features adapted from Frampton et. al. (2009) used information about previous and next sentences and speakers. These features connected the current utterance with previous utterances by the other participants in the room. For each listener, a feature was recorded that indicated how many sentences elapsed between the current sentence and the last/next time the person spoke. Additionally, two features captured the number of speakers in the previous and next five sentences.

Gesture Features. Several different features were used to capture gesture information. Three types of gesture data were considered: all produced gestures, only those gestures that were manually annotated as being communicative, and only those gestures that were manually annotated as pointing towards another meeting participant. For each of these types, one gesture feature captures the total number of gestures that co-occur with the current sentence, while another feature records only whether or not a gesture co-occurs with the utterance of *you*. Since previous work (Kendon, 1980) has indicated that gesture production tends to precede the onset of the expression, gestures were considered to have co-occurred with instances if they directly overlapped with them or preceded them by a short window of 2.5 seconds.

Note that in this investigation, we used annotated gestures provided by the AMI corpus. Although automated extraction of reliable gesture features can be challenging and should be pursued in the future, the use of manual annotation allows us to focus on our current goal, which is to understand whether and to what degree hand gestures may help disambiguation of *you* Type.

It is also important to note that although previous work (Purver et al., 2009) showed that n-gram features produced large performance gains, these features were heavily topic dependent. The AMI meeting corpus provides several meetings on exactly the same topic, which allowed the n-gram features to learn topic-specific words such as *button, channel, and volume*. However, as real world

	Accuracy
Majority Class Baseline	53.3%
Gupta automatic	70.7%
Gupta manual	74.7%
Gupta + Frampton automatic	73.2%
Gupta + Frampton manual	74.3%
All (+ gesture)	**79.0**%

Table 1: Accuracy values for Generic vs. Referential Classification

	Accuracy
Majority Class Baseline	59.5%
Gupta automatic	72.2%
Gupta manual	73.6%
Gupta + Frampton automatic	73.2%
Gupta + Frampton manual	72.5%
All (+ gesture)	**74.6**%

Table 2: Accuracy values for Singular vs. Plural Classification

meetings occur with a wider range of goals and topics, we would like to build a topic and domain independent model that does not require a corpus of topic specific training data. As such, we have excluded n-gram features from our study.

Additionally, we have not implemented gaze features. Although previous work (Frampton et al., 2009) showed that these features were able to improve performance, we decided to focus solely on gesture to the exclusion of other non-speech modalities. However, we are currently in the process of evaluating the overlap between gesture and gaze feature coverage.

5 Results

Due to the small number of meeting segments in our data, leave-one-out cross validation was preformed for evaluation. Since a primary focus of this paper is to understand whether and to what degree gesture is able to aid in the *you* type identification task, experiments were run using a decision tree classifier due to its simplicity and transparency [1].

5.1 Two Stage Classification

We first evaluated the role of gesture via two stage binary classification. That is, we performed two binary classification tasks, first differentiating between generic and referential instances, and then further dividing the referential instances into the singular and plural types. This provides a more detailed analysis of where gesture may be helpful.

Results for the generic vs. referential and singular vs. plural binary classification tasks are shown in Table 1 and Table 2, respectively. Tables 1 and 2 present several different configurations. The "Gupta" feature configurations consist of all features used by Gupta et. al. (2007a). These include all part-of-speech features, all dialogue act features, the question mark feature, and all sentential features except the "<auxiliary> you" feature and the word count feature. Results from two types of processing are presented: automatic and manual.

- *Automatic feature extraction (automatic)* - The automatic configurations consist of only features that were automatically extracted from the text. This includes all of the features we examined except for the dialogue act and gesture features. These features are extracted from meeting transcriptions.
- *Manual feature extraction (manual)* - Manual configurations apply manual annotations of dialogue acts and gestures together with the automatically extracted features.

The Frampton configurations add the additional sentential features as well as the backward-looking and forward-looking features. As before, results are presented for a manual and an automatic run. The final configuration ("All") includes the entire feature set with the addition of gesture features. The *All* configuration is the only configuration that includes gesture features.

Although they are not directly comparable, the results for generic vs. referential classification shown in Table 1 appear consistent with those reported by Gupta (2007a). Adding additional features from Frampton et. al. did not produce an overall increase in performance when dialogue act features were present. Including gesture features leads to a significant increase in performance (McNemar Test, $p < 0.01$), an absolute increase of 4.3% over the best performing feature set that does not include gesture. This result seems to confirm our hypothesis that, because gestures are likely

[1] In order to get a more direct comparison to previous work (Gupta et al., 2007a; Frampton et al., 2009), we also experimented with classification via a bayesian network. We found that the overall results were comparable to those obtained with the decision tree.

	Accuracy
Majority Class Baseline	46.7%
Gupta automatic	61.5%
Gupta manual	66.2%
Gupta + Frampton automatic	63.6%
Gupta + Frampton manual	70.2%
All (+ gesture)	**70.4**%

Table 3: Accuracy values for several different feature configurations on the three class classification problem.

to accompany referential instances of *you* but not generic instances, gesture information is able to help differentiate between the two. Manual inspection of the decision tree produced indicates that gesture features were among the most discriminative features.

The results on the singular vs. plural task shown in Table 2 are less clear. Although (Gupta et al., 2007a) did not report results on singular vs. plural classification, their feature set produced reasonable classification accuracy of 73.6%. Including gesture and other features did not produce a statistically significant improvement in the overall accuracy. This suggests that while gesture is helpful for predicting referentiality, it does not appear to be a reliable predictor of whether an instance of *you* is singular or plural. Inspection on the decision tree confirms that gesture features were not seen to be highly discriminative.

5.2 Three Class Classification

The results presented for singular vs. plural classification are based on performance on the subset of *you* instances that are referential, which assumes that we are able to filter out generic references with 100% accuracy. While this gives us an evaluation of how well the singular vs. plural task can be performed without the generic references presenting a confounding factor, it presents unrealistic performance for a real system. To account for this, we present results on a three class problem of determining whether an instance of *you* is generic, singular, or plural. The results are shown in Table 3. A simple majority class classifier yields accuracy of 46.7% (In our data, the generic class was the majority class).

As we can see from Table 3, adding additional features gives improved performance over the original implementation by Gupta et. al., resulting in an overall accuracy of about 70%. We also observed that the dialogue act features were important; manual configurations produced absolute gains of about 7% accuracy over fully automatic configurations. The gesture feature, however, did not provide a significant increase in performance over the same feature set without gesture information.

Table 4 shows the precision, recall, and F-measure values for each *you* type for several different configurations. As shown, the generic class proved to be the easiest for the classifiers to identify. This is not suprising, as not only are generic instance our majority class, but many of the features used were originally tailored towards the two class problem of differentiating generic instances from the other classes. The performance on the plural and singular classes is comparable to one another when the basic feature set is used. However, as more features are added, the performance on the singular class increases while the performance on the plural class does not. This seems to suggest that future work should attempt to include more features that are indicative of plural instances.

When manual dialogue acts are applied, it appears incorporation of gestures does not lead to any overall performance improvement (as shown in Table 3). One possible explanation is that gesture features as they are incorporated here do provide some disambiguating information (as shown in the two stage classification), but this information is subsumed by other features, such as dialogue acts. To test this hypothesis, we ran an experiment with a feature set that contained all features except dialogue act features. That is, a feature set that contains all of the automatic features, as well as gesture features. Results are shown in Table 5.

Our "automatic + gesture" feature configuration produced accuracy of 66.2%. When compared to the same feature set without gesture features (the "Gupta + Frampton automatic" row in Table 3) we see a statistically significant ($p < 0.01$) absolute accuracy improvement of about 2.6%. This seems to suggest that gesture features are providing some small amount of relevant information that is not captured by our automatically extractable features.

Up until this point we have incorporated dialogue acts using the full set of dialogue act tags provided by the AMI corpus. As we have men-

		Precision	Recall	F-Measure
Gupta automatic	Plural	0.553	0.548	0.550
	Singular	0.657	0.408	0.504
	Generic	0.624	0.787	0.696
Gupta manual	Plural	0.536	0.513	0.524
	Singular	0.675	0.503	0.576
	Generic	0.704	0.839	0.766
All (+ gesture)	Plural	0.542	0.565	**0.553**
	Singular	0.745	0.604	**0.667**
	Generic	0.754	0.835	**0.792**

Table 4: Precision, recall, and F-measure results for each *you* type based on three class classification.

	Accuracy
Gupta + Frampton automatic	63.6%
Gupta + Frampton automatic + gesture	**66.2**%
Gupta + Frampton automatic + simple dialogue act	66.6%
Gupta + Frampton automatic + simple dialogue act + gesture	**69.0**%

Table 5: Accuracy for 3-way classification by combining gesture information with automatically extracted features based on the Decision Tree model

tioned, this level of granularity may not be practically extractable for use in a current state-of-the-art system. As a result, we implemented the simpler dialogue act incorporation method proposed by (Gupta et al., 2007a), in which only the presence or absence of the elicit dialogue act type is considered. Using this feature with the automatically extracted features yielded accuracy of 66.6%, a statistically significant improvement ($p < 0.01$) of an absolute 3% over a fully automatic run. Furthermore, if we incorporate gesture features with this configuration, the performance increases to 69.0% (statistically significantly, $p < 0.01$). This suggests that while gesture features may be redundant with information provided by the full set of dialogue act tags, it is largely complementary with the simpler dialogue act incorporation. The incorporation of gesture along with simpler and more reliable dialogue acts can potentially approach the performance gained by incorporation of more complex dialogue acts, which are often difficult to obtain. Of course, gesture features themselves are often difficult to obtain. However, redundancy in two potentially error-prone feature sources can be an asset, as data from one source may help to compensate for errors in the other. Although addressing a different problem of multimodal integration, previous work (Oviatt et al., 1997) appears to indicate that this is the case.

6 Conclusion

In this paper, we investigate the role of hand gestures in disambiguating types of *You* expressions in multiparty meetings for the purpose of machine translation.

Our results have shown that on the binary generic vs. referential classification problem, the inclusion of gesture data provides a statistically significant increase in performance over the same feature set without gesture. This result is consistent with our hypothesis that gesture data would be helpful because speakers are more likely to gesture when producing referential instances of *you*.

To produce results more akin to those that would be expected during incorporation in a real machine translation system, we experimented with the type identification problem as a three class classification problem. It was discovered that when a full set of dialogue act tags were used as features, the incorporation of gesture features does not provide an increase in performance. However, when simpler dialogue act tags are used, the incorporation of gestures helps to make up for lost performance. Since it remains a difficult problem to automatically predict complex dialog acts with high accuracy, the incorporation of gesture features may prove beneficial to current systems.

7 Acknowledgement

This work was supported by IIS-0855131 (to the first two authors) and IIS-0840461 (to the third author) from the National Science Foundation. The authors would like to thank anonymous reviewers for valuable comments and suggestions.

References

Tyler Baldwin, Joyce Y. Chai, and Katrin Kirchhoff. 2009. Communicative gestures in coreference identification in multiparty meetings. In *ICMI-MLMI '09: Proceedings of the 2009 international conference on Multimodal interfaces*, pages 211–218. ACM.

J. B. Bavelas, N. Chovil, L. Coates, and L. Roe. 1995. Gestures specialized for dialogue. *Personality and Social Psychology Bulletin*, 21:394–405.

Jacob Eisenstein and Randall Davis. 2006. Gesture improves coreference resolution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 37–40, New York City, USA, June. Association for Computational Linguistics.

Jacob Eisenstein and Randall Davis. 2007. Conditional modality fusion for coreference resolution. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 352–359, Prague, Czech Republic, June. Association for Computational Linguistics.

Matthew Frampton, Raquel Fernández, Patrick Ehlen, Mario Christoudias, Trevor Darrell, and Stanley Peters. 2009. Who is "you"?: combining linguistic and gaze features to resolve second-person references in dialogue. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 273–281, Morristown, NJ, USA. Association for Computational Linguistics.

Surabhi Gupta, John Niekrasz, Matthew Purver, and Dan Jurafsky. 2007a. Resolving you in multi-party dialog. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*.

Surabhi Gupta, Matthew Purver, and Dan Jurafsky. 2007b. Disambiguating between generic and referential you in dialog. In *Proceedings of the 42th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Adam Kendon. 1980. Gesticulation and speech: Two aspects of the process of utterance. In Mary Richie Key, editor, *The Relationship of Verbal and Nonverbal Communication*, pages 207–227.

D. McNeill. 1992. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press.

W. M. Meyers. 1990. Current generic pronoun usage. *American Speech*, 65(3):228–237.

Sharon Oviatt, Antonella DeAngeli, and Karen Kuhn. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, New York, NY, USA. ACM.

Andrei Popescu-Belis and Paula Estrella. 2007. Generating usable formats for metadata and annotations in a large meeting corpus. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 93–96, Prague, Czech Republic, June. Association for Computational Linguistics.

Matthew Purver, Raquel Fernández, Matthew Frampton, and Stanley Peters. 2009. Cascaded lexicalised classifiers for second-person reference resolution. In *SIGDIAL '09: Proceedings of the SIGDIAL 2009 Conference*, pages 306–309, Morristown, NJ, USA. Association for Computational Linguistics.

User-adaptive Coordination of Agent Communicative Behavior in Spoken Dialogue

Kohji Dohsaka
NTT Communication Science Laboratories
NTT Corporation
2-4, Hikaridai, Seika-cho,
Kyoto 619-0237, Japan

Atsushi Kanemoto
Graduate School of
Information Science and Technology
Osaka University, 1-1, Yamadaoka,
Suita, Osaka 565-0871, Japan

Ryuichiro Higashinaka
NTT Cyber Space Laboratories
NTT Corporation
1-1, Hikarinooka, Yokosuka,
Kanagawa 239-0847, Japan

Yasuhiro Minami and **Eisaku Maeda**
NTT Communication Science Laboratories
NTT Corporation
2-4, Hikaridai, Seika-cho,
Kyoto 619-0237, Japan

{dohsaka,minami,maeda}@cslab.kecl.ntt.co.jp
higashinaka.ryuichiro@lab.ntt.co.jp

Abstract

In this paper, which addresses smooth spoken interaction between human users and conversational agents, we present an experimental study that evaluates a method for user-adaptive coordination of agent communicative behavior. Our method adapts the pause duration preceding agent utterances and the agent gaze duration to reduce the discomfort perceived by individual users during interaction. The experimental results showed a statistically significant tendency: the duration of the agent pause and the gaze converged during interaction with the method. The method also significantly improved the perceived relevance of the agent communicative behavior.

1 Introduction

Conversational agents have been studied as an effective human-computer interface for such purposes as training decision-making in team activities (Traum and Rickel, 2002), learning support (Johnson et al., 2002), museum guides (Kopp et al., 2005), and community facilitators (Zheng et al., 2005; Fujie et al., 2009). They will play a crucial role in establishing a society where humans and robots collaborate through natural interaction. However, agents cannot produce their intended effects when the smooth flow of interaction is disturbed. To fully exploit the promise of agents, we need to achieve smooth interaction between human users and agents.

Although various types of modalities have been used in human-computer interfaces, speech has drawn a great deal of interest because it is one of the most pervasive communication methods in our daily lives and we usually perform it without any special effort (Nass and Brave, 2005). In this paper, we are interested in smooth spoken dialogues between users and agents.

A spoken dialogue is a joint activity among participants (Clark, 1996). For such a joint activity to be smooth and successful, participants need to coordinate their communicative behaviors in various ways. In human dialogues, participants agree on lexical choices to refer to objects (Brennan and Clark, 1996) and coordinate eye gaze (Richardson and Dale, 2005) and whose turn it is to speak (Sacks et al., 1974). They become more similar to their partners as the dialogue proceeds in many aspects such as pitch, speech rate, and pause structure (Burgoon et al., 1995; Hayashi et al., 2009). Such coordination serves to make conversation flow easily and intelligibly (Garrod and Pickering, 2004).

The coordination of communicative behaviors also plays a crucial role in smooth human-agent interaction. Previous work addressed human behavior adaptation to agents (Oviatt et al., 2004), agent behavior adaptation to human partners (Mitsunaga et al., 2005; Tapus and Matarić, 2007), and the mutual adaptation of human and agent behavior (Breazeal, 2003).

In this paper, which addresses smooth spoken interaction between human users and agents, we focus on the adaptation of agent communicative behavior to individual users in spoken dialogues

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 314–321,
The University of Tokyo, September 24-25, 2010.

with flexible turn-taking. We present a method for user-adaptive coordination of agent communicative behavior to reduce the discomfort perceived by individual users during the interaction and show experimental results that evaluate how the method influences agent communicative behavior and improves its relevance as perceived by users. For evaluation purposes, we used a quiz-style multi-party spoken dialogue system (Minami et al., 2007; Dohsaka et al., 2009). A quiz-style dialogue is a kind of thought-evoking dialogue that can stir user thinking and activate communication (Higashinaka et al., 2007a; Dohsaka et al., 2009). This characteristic is expected to be advantageous for evaluation experiments since it encourages involvement in the dialogue.

Our method adapts agent communicative behavior based on policy gradient reinforcement learning (Sutton et al., 2000; Kohl and Stone, 2004). The policy gradient method has been used for robot communicative behavior adaptation (Mitsunaga et al., 2005; Tapus and Matarić, 2007). However, both studies dealt with scenario-based interaction in which a user and a robot acted with predetermined timing. In contrast, we focus on spoken dialogues in which users and agents can speak with more flexible timing. In addition, we allow for two- and three-party interactions among a user and two agents. It remains unclear whether the policy gradient method can successfully adapt agent communicative behavior to a user in two- or three-party spoken dialogues with flexible turn-taking. Although this paper focuses on agent behavior adaptation to human users, we believe that our investigation of the agent behavior adaptation mechanism in flexible spoken interaction will contribute to conversational interfaces where human users and agents can mutually adapt their communicative behaviors.

As agent communicative behavior to be adapted, this paper focuses on the pause duration preceding agent utterances and the agent gaze duration. In conversation, the participant pause duration is influenced by partners, and the coordination of pause structure leads to smooth communication (Burgoon et al., 1995; Hayashi et al., 2009). Without pause structure coordination, undesired speech overlaps or utterance collisions are likely to occur between users and agents, which may disturb smooth communication. Funakoshi *et al.* proposed a method to prevent undesired speech overlaps in human-robot speech interactions by using a robot's subtle expressions produced by a blinking LED attached to its chest (Funakoshi et al., 2008). In their method, a blinking light notifies users about such internal states of the robot as processing or busy and helps users identify the robot pause structures; however we are concerned with the adaptation of robot pause structures to users.

Gaze coordination is causally related to the success of communication (Richardson and Dale, 2005), and the amount of gaze influences conversational turn-taking (Vertegaal and Ding, 2002). The relevant control of agent gaze duration is thus essential to the smooth flow of conversation. Moreover, since the amount of gaze is related to specific interpersonal attitudes among participants and is also subject to such individual differences as personalities (Argyle and Cook, 1976), agent gaze duration must be adapted to individual users.

In the following, Section 2 describes our quiz-style multi-party spoken dialogue system. Section 3 shows our method for the user-adaptive coordination of agent communicative behavior. Section 4 explains the experiment, and Section 5 describes its results. Section 6 concludes our paper.

2 Quiz-Style Spoken Dialogue System

To evaluate a method for agent communicative behavior adaptation, we used a quiz-style multi-party spoken dialogue system based on a quiz-style two-party spoken dialogue system (Minami et al., 2007) and extended it to perform multi-party interaction (Dohsaka et al., 2009).

In this system, a human user and one or two agents interact. The two agents include a quiz-master and a peer. The quizmaster agent creates a "Who is this?" quiz about a famous person and presents hints one by one to the user and the peer agent, who participates in the interaction and guesses the correct answer in the same way that the user does.

The hints are automatically created from the biographical facts of people in Wikipedia[1] and ranked based on the difficulty of solving the quizzes experienced by users (Higashinaka et al., 2007b). Since users must consider the hints to offer reasonable answers, the system can stimulate their thinking and encourage them to engage in the interaction (Higashinaka et al., 2007a). In addition, the peer agent's presence and the agent's empathic expressions improve user satisfaction and

[1] http://ja.wikipedia.org/

Figure 1: User interacting with two agents using the quiz-style spoken dialogue system

increase user utterances (Dohsaka et al., 2009).

Figure 1 shows a human user interacting with the two agents, both of whom are physically embodied robots. The system utilizes an extremely large vocabulary with continuous speech recognition (Hori et al., 2007). Agent utterances are produced by speech synthesis. The agents can gaze at other participants by directing their faces to them. At each point of the dialogue, the system chooses the next speaker and its utterance based on the dialogue state that the system maintains, the preconditions of the individual utterances, and a few turn-taking rules (Dohsaka et al., 2009). The agent pause and gaze durations are controlled based on the adaptation method described in Section 3.

A sample dialogue among a user and two agents is depicted in Figure 2. *Master* is the quizmaster agent, and *Peer* is the peer agent. The agent utterances are classified as either spontaneous or responsive. Spontaneous utterances are those made after an agent takes his turn in an unforced manner, and responsive utterances are responses to the other's utterances. In the sample dialogue, *spon* identifies spontaneous and *res* identifies responsive utterances.

Quizmaster agent *Master* makes spontaneous utterances such as presenting hints (lines 1 and 5), indicating the quiz difficulty, and addressing listeners. It also makes such responsive utterances as evaluating the other's answers (lines 3, 9, and 11). Peer agent *Peer* makes such spontaneous utterances as showing its own difficulty (line 4), giving an answer (line 8), giving feedback when its own or the other's answer is right (line 12), and addressing listeners. It also makes such responsive utterances as showing empathy to the user (line 7).

3 Method for Agent Communicative Behavior Adaptation

We apply policy gradient reinforcement learning (Sutton et al., 2000; Kohl and Stone, 2004)

1	*Master*	Who is this? First hint. He graduated from the University of Tokyo. *(hint/spon)*
2	User	Yoshida Shigeru? *(answer/spon)*
3	*Master*	No, not even close! He's not a politician. *(evaluation/res)*
4	*Peer*	I don't know. Very difficult. *(show difficulty/spon)*
5	*Master*	It's time for the second hint: He's a novelist and a scholar of British literature. *(hint/spon)*
6	User	Oh, I think I know it but I can't remember his name. That's so frustrating. *(show difficulty/spon)*
7	*Peer*	Difficult for me, too. *(show empathy/res)*
8	*Peer*	Haruki Murakami? *(answer/spon)*
9	*Master*	Close! You are half right, because he is a novelist. *(evaluation/res)*
10	User	Natsume Soseki? *(answer/spon)*
11	*Master*	That's right. Wonderful. *(evaluation/res)*
12	*Peer*	Good job. *(feedback/spon)*

Figure 2: Sample dialogue between user and two agents: quizmaster *Master* and peer *Peer*. *Spon* identifies spontaneous and *res* identifies responsive utterances.

to the user-adaptive coordination of agent communicative behavior. A policy gradient method is a reinforcement learning (RL) approach that directly optimizes a parameterized policy by gradient ascent based on the gradient of the expected reward with respect to the policy parameters. Although RL methods have recently been applied to optimizing dialogue management in spoken dialogue systems (Williams and Young, 2007; Minami et al., 2009), these previous studies utilized RL methods based on the value-function estimation. The policy gradient method is an alternative approach to RL that has the following merits. It can handle continuous and large action spaces (Kimura and Kobayashi, 1998) and is usually assured to converge to a locally optimal policy in such action spaces (Sutton et al., 2000). Moreover, it does not need to explicitly estimate the value function, and it is incremental, requiring only a constant amount of computation per learning step (Kimura and Kobayashi, 1998).

Due to these advantages, the policy gradient method is suitable for adapting agent communicative behavior to a user during interaction, because

(1) $\boldsymbol{\Theta} = [\theta_j] \leftarrow$ initial policy (policy parameter vector of size n)
(2) $\boldsymbol{\epsilon} = [\epsilon_j] \leftarrow$ step size vector of size n
(3) $\eta \leftarrow$ overall scalar step size
(4) $maxA \leftarrow 0$ (greatest absolute value of reward ever observed in adaptation process)
(5) while dialogue continues
(6) for i = 1 to T
(7) for j = 1 to n
(8) $r_j \leftarrow$ random choice from $\{-1, 0, 1\}$
(9) $R^i_j \leftarrow \theta_j + \epsilon_j * r_j$
($\boldsymbol{R^i}$ is T random perturbations of $\boldsymbol{\Theta}$)
(10) for i = 1 to T
(11) Perform a hint dialogue based on policy $\boldsymbol{R^i}$, and evaluate reward
(12) for j = 1 to n
(13) $Avg_{+\epsilon,j} \leftarrow$ average reward for all $\boldsymbol{R^i}$ with positive perturbation in parameter ϵ_j
(14) $Avg_{0,j} \leftarrow$ average reward for all $\boldsymbol{R^i}$ with zero perturbation in parameter ϵ_j
(15) $Avg_{-\epsilon,j} \leftarrow$ average reward for all $\boldsymbol{R^i}$ with negative perturbation in parameter ϵ_j
(16) if ($Avg_{0,j} > Avg_{+\epsilon,j}$ and $Avg_{0,j} > Avg_{-\epsilon,j}$)
(17) $a_j \leftarrow 0$
(18) else
(19) $a_j \leftarrow Avg_{+\epsilon,j} - Avg_{-\epsilon,j}$
(20) $\forall j (a_j \leftarrow \frac{a_j}{|\boldsymbol{A}|} * \epsilon_j * \eta)$
(21) $maxC \leftarrow$ maximum of absolute value of reward in current adaptation cycle
(22) if ($maxC > maxA$)
(23) $maxA \leftarrow maxC$ (update $maxA$)
(24) else
(25) $\boldsymbol{A} \leftarrow \boldsymbol{A} * \frac{maxC}{maxA}$
(26) $\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} + \boldsymbol{A}$

Figure 3: Pseudocode for user-adaptive coordination of agent communicative behavior

it can naturally incorporate such continuous parameters as pause and gaze duration and incrementally adapt agent behavior. In fact, the policy gradient method has been successfully used for robot behavior adaptation (Mitsunaga et al., 2005; Tapus and Matarić, 2007). In this paper, we apply this method to agent communicative behavior adaptation in spoken dialogues with flexible turn-taking.

Figure 3 shows our method for the user-adaptive coordination of agent communicative behavior. This method is a modification of an algorithm presented by Kohl and Stone (2004) in that the gradient is adjusted based on the maximum absolute value of the reward obtained during each adaptation cycle.

The agent communicative behaviors are determined based on a policy that is represented as vector $\boldsymbol{\Theta}(= [\theta_j])$ of n policy parameters. In the quiz-style dialogues, the behavior of both the quizmaster and peer agents is controlled based on the same policy parameters. The method adapts the behavior of both agents to individual users by adapting the policy parameters. In this experiment, we used the following four parameters ($n = 4$):

- pre-spontaneous-utterance pause duration σ_{spon}: duration of pauses preceding agent spontaneous utterances
- pre-responsive-utterance pause duration σ_{res}: duration of pauses preceding agent responsive utterances
- gaze duration σ_{gaze}: duration of agent's directing its face to the other while it is speaking or listening
- hint interval σ_{hint}: interval of presenting quiz hints

As shown above, we used two types of pause duration since the relevant pause duration can be dependent on dialogue acts (Itoh et al., 2009). Although our main concern is the pause and gaze duration, we examined the hint interval as a parameter particular to quiz-style dialogues.

To adapt the policy parameters to individual users, we first generate T random perturbations $[\boldsymbol{R^1}, \ldots, \boldsymbol{R^T}]$ of current policy $\boldsymbol{\Theta}$ by randomly adding $\epsilon_j, 0, -\epsilon_j$ to each parameter θ_j of $\boldsymbol{\Theta}$ in lines 6 to 9, where ϵ_j is a step size set for each parameter. In the experiment, we set T to 10. The step sizes of the parameters used in the experiment will be shown later in Table 1.

Dialogue per hint (a hint dialogue) is then performed based on each perturbation policy $\boldsymbol{R^i}$, and the reward for each hint dialogue is obtained in lines 10 to 11. All agent behaviors in a hint dialogue are determined based on the same perturbation policy. As we will explain in Section 4, in the experiment, we regarded the magnitude of discomfort perceived by users during a hint dialogue as a negative reward. Users signified discomfort by pressing buttons on the controller held in their hands. After performing hint dialogues for all T perturbations $\boldsymbol{R^i}$, gradient $\boldsymbol{A}(= [a_j])$ is computed in lines 12 to 19. The gradient is normalized by

Parameters	σ_{spon} (sec.)	σ_{res} (sec.)	σ_{gaze} (sec.)	σ_{hint} (sec.)
Initial value	4.96	0.53	3.04	27.7
Step size	0.50	0.20	0.30	2.5

Table 1: Initial values and step sizes of policy parameters: σ_{spon} (pre-spontaneous-utterance pause duration), σ_{res} (pre-responsive-utterance pause duration), σ_{gaze} (gaze duration), and σ_{hint} (hint interval)

overall scalar step size η and individual step size ϵ_j for each parameter in line 20. Overall scalar step size η is used to adjust the adaptation speed, which we set to 1.0.

Next we get the maximum $maxC$ of the absolute value of the reward in the current adaptation cycle. As in lines 21 to 25, the gradient is adjusted based on the ratio of $maxC$ to the greatest absolute value $maxA$ of reward ever observed in the overall adaptation process. Finally, the current policy parameters are updated using the gradient in line 26.

This is an adaptation cycle. By iterating it, the agent communicative behavior is adapted to reduce the discomfort perceived by each user.

4 Experiment

We recruited and paid 32 Japanese adults (16 males and 16 females) for their participation. The mean ages of the male and female groups were 33.2 and 36.8, respectively. They were divided into two groups: two-party dialogues (user and quizmaster) and three-party dialogues (user, quizmaster, and peer). In each group, the numbers of males and females were identical.

For this experiment, we used a quiz-style spoken dialogue system. We chose the quiz subjects in advance and divided them into sets of five so that the difficulty level was approximately the same in all sets. For this purpose, we made several sets of five people of approximately identical PageRank ™ scores based on Wikipedia's hyperlink structure.

The users first rehearsed the dialogues for a set of five quizzes to familiarize themselves with the system. After practicing, they performed the dialogues to evaluate the adaptation method and took a break per five-quiz set. The presentation order of the quiz sets was permutated to prevent order effect. For each user, the dialogues continued until the user received 150 hints. The adaptation method was applied during the interaction, and the policy parameters were updated per 10 hint dialogues. As a result, the parameters were updated 15 times through the dialogues. It took about two hours for each user to complete all dialogues.

The policy parameters were updated based on the magnitude of discomfort perceived by users. In this experiment, users were told to concentrate on the discomfort caused by agent pause and gaze duration and signified it by pressing buttons on the controller held in their hands at three levels of magnitude: '3', '2', and '1'. The sum of discomfort obtained during a hint dialogue was normalized with respect to the hint dialogue length, and the normalized values were regarded as negative rewards. Ideally we should estimate user discomfort from such user behaviors as pause structure and eye gaze. However, as the first step toward that goal, in this experiment we adopted this setting in which users directly signified their discomfort by pressing buttons.

Table 1 shows the initial values and the step sizes of the policy parameters used in the experiment. To obtain the relevant initial values, we conducted a preparatory experiment in which ten other participants performed quiz-style dialogues under the same conditions as this experiment. The initial values in this experiment were set to the averaged final values of the policy parameters in the preparatory experiment. The step sizes were determined as approximately one-tenth of the initial values except for the pre-responsive-utterance pause, for which the step size was set to 200 msec based on the limits of human perception.

Before and after the adaptation, the users filled out the following questionnaire items (7-point Likert scale) to evaluate the relevance of agent pause and gaze duration:

- Did you feel that the pause duration preceding the agent utterances was relevant?
- Did you feel that the agent gaze duration was relevant while the agents were speaking or listening to you?

5 Results

5.1 Convergence of policy parameters

The policy parameters were updated based on the adaptation method during the user-agent interaction. Figure 4 exemplifies how the policy parameter values changed during the adaptation cycles with a user engaged in the two-party dialogue.

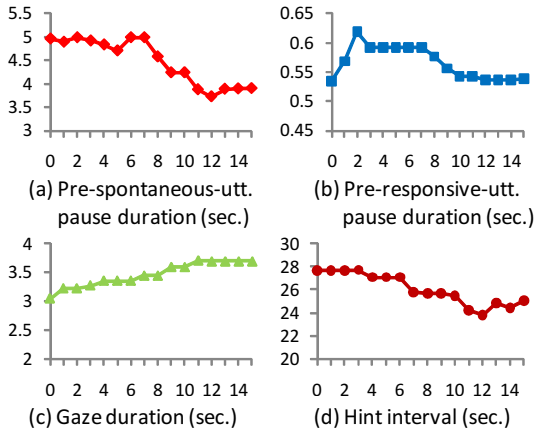


Figure 4: Change of policy parameter values during adaptation cycles with a user engaged in two-party dialogue. Horizontal axis shows adaptation cycles and vertical axis shows parameter values.

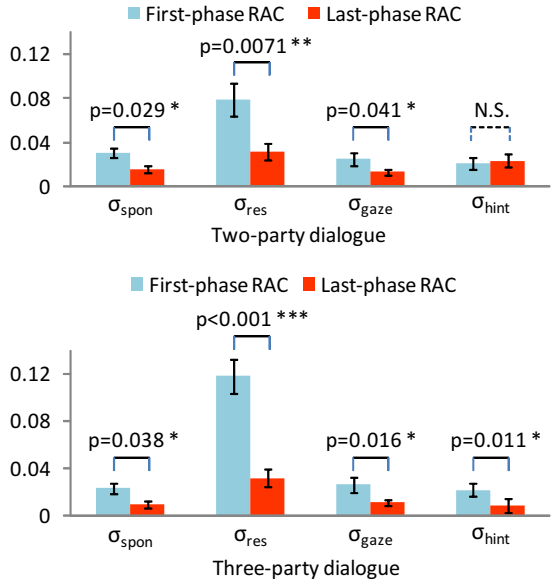


Figure 5: For each policy parameter, average and standard error of first- and last-phase RACs (relative amount of change in parameter values)

Table 2 shows the statistics of the final values of the policy parameters at the end of the adaptation process. Since the initial values were appropriately determined based on the preparatory experiment, the final value averages were not greatly different from the initial values. However, judging from the maximum, minimum, and standard deviations, the final values reflected individual users.

If the adaptation method works successfully, the policy parameter values should converge during the user-agent interaction. From this viewpoint, we examined the relative amount of change in the policy parameters (RAC). Given parameter value p_{k-1} at $(k-1)$-th adaptation cycle and parameter value p_k at k-th cycle, RAC is defined as $\frac{|p_k - p_{k-1}|}{p_{k-1}}$.

Two-party dialogues				
Parameters	σ_{spon} (sec.)	σ_{res} (sec.)	σ_{gaze} (sec.)	σ_{hint} (sec.)
Average	5.04	0.62	3.10	25.8
Min	3.90	0.39	2.40	19.5
Max	6.17	1.18	3.69	31.2
Sd.	0.72	0.21	0.36	2.7
Three-party dialogues				
Parameters	σ_{spon} (sec.)	σ_{res} (sec.)	σ_{gaze} (sec.)	σ_{hint} (sec.)
Average	4.86	0.62	3.15	27.4
Min	4.07	0.35	2.52	22.0
Max	5.54	0.90	3.58	32.7
Sd.	0.44	0.18	0.27	2.5

Table 2: Statistics of final values of policy parameters: σ_{spon} (pre-spontaneous-utterance pause duration), σ_{res} (pre-responsive-utterance pause duration), σ_{gaze} (gaze duration), and σ_{hint} (hint interval)

For each policy parameter, we compared the RAC averages in the first and in the last three adaptation cycles: the first-phase RAC and the last-phase RAC. As shown in Figure 5, the last-phase RAC tends to be smaller than the first-phase RAC. The Kolmogorov-Smirnov test showed that the assumption of normality ($p > 0.2$) was met for each group. By applying the paired Welch's t-test, as shown in Figure 5, we found that the last-phase RAC is significantly smaller than the first-phase RAC except for the hint interval in the two-party dialogues. This shows that the agent pause and gaze duration converged during the interaction in both the two- and three-party dialogues.

The hint interval is unlikely to converge, probably because it is a longer period than the pause and gaze duration and is subject to various factors. Moreover, it greatly depends on user interest.

5.2 User evaluations

Figure 6 shows the subjective user evaluations of the relevance of agent pause and gaze duration before and after the adaptation. Each user evaluation was measured by a Likert question. The rating of a single Likert question is an ordinal measure, and we generally cannot apply a parametric statistical test to an ordinal measure. Therefore we used a nonparametric test, the Wilcoxon signed-rank test, to compare user evaluations before and after the

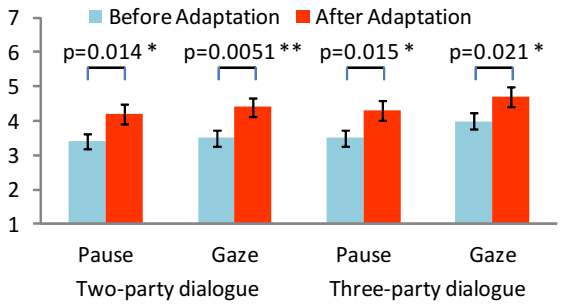


Figure 6: Average and standard error of user evaluations of relevance of agent pause and gaze duration before and after adaptation

adaptation. The F-test for the homogeneity of variances ($p > 0.1$) showed that the data satisfied the statistical test assumption.

We found that in both the two- and three-party dialogues, the relevance of the agent pause and gaze duration significantly improved during the two-hour adaptation process ($p < 0.01$ for gaze duration in the two-party dialogues, $p < 0.05$ for other cases). The p-values are shown in Figure 6. No significant differences between gender were found.

These results on the convergence of policy parameters and user evaluations show that the policy-gradient-based method can adapt agent communicative behavior to individual users in spoken dialogues with flexible turn-taking.

6 Conclusion

In this paper, addressing smooth spoken interaction between human users and conversational agents, we presented a method for user-adaptive coordination of agent communicative behavior and experimentally evaluated how it can adapt agent behavior to individual users in spoken dialogues with flexible turn-taking. The method coordinates agent pause and gaze duration based on policy gradient reinforcement learning to reduce the discomfort perceived by individual users during interaction. We experimentally evaluated the method in a setting where the users performed two- and three-party quiz-style dialogues and signified their discomfort by pressing buttons held in their hands. Our experimental results showed a statistically significant tendency: the agent pause and gaze duration converged during interaction with the method in both two- or three-party dialogues. The method also significantly improved the perceived relevance of the agent communicative behavior in both two- and three-party dialogues. These results indicate that in spoken dialogues with flexible turn-taking, the policy-gradient-based method can adapt agent communicative behavior to individual users.

Many directions for future work remain. First, we will analyze how users adapt their communicative behaviors with our method. Second, we need to automatically estimate user discomfort or satisfaction based on such user behaviors as pause structure, prosody, eye gaze, and body posture. Third, we will extend the adaptation method to regulate agent behavior based on dialogue states, since one limitation of the current method is its inability to recognize them. Fourth, we are interested in the adaptation of additional higher-level actions like the relevant choice of dialogue topics based on the level of user interest.

Acknowledgments

This work was partially supported by a Grant-in-Aid for Scientific Research on Innovative Areas, "Founding a creative society via collaboration between humans and robots" (21118004), from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

References

Michael Argyle and Mark Cook. 1976. *Gaze and Mutual Gaze*. Cambridge University Press.

Cynthia Breazeal. 2003. Regulation and entrainment for human-robot interaction. *International Journal of Experimental Robotics*, 21(10-11):883–902.

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.

Judee K. Burgoon, Lesa A. Stern, and Leesa Dillman. 1995. *Interpersonal Adaptation: Dyadic Interaction Patterns*. Cambridge University Press.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press.

Kohji Dohsaka, Ryota Asai, Ryuichiro Higashinaka, Yasuhiro Minami, and Eisaku Maeda. 2009. Effects of conversational agents on human communication in thought-evoking multi-party dialogues. In *Proc. of SIGDIAL 2009*, pages 217–224.

Shinya Fujie, Yoichi Matsuyama, Hikaru Taniyama, and Tetsunori Kobayashi. 2009. Conversation robot participating in and activating a group communication. In *Proc. of Interspeech 2009*, pages 264–267.

Kotaro Funakoshi, Kazuki Kobayashi, Mikio Nakano, Seiji Yamada, Yasuhiko Kitamura, and Hiroshi Tsujino. 2008. Smoothing human-robot speech interactions by using a blinking-light as subtle expression. In *Proc. of ICMI 2008*, pages 293–296.

Simon Garrod and Martin J. Pickering. 2004. Why is conversation so easy? *Trends in Cognitive Sciences*, 8:8–11.

Takanori Hayashi, Shohei Kato, and Hidenori Itoh. 2009. A synchronous model of mental rhythm using paralanguage for communication robots. In *Lecture Notes in Computer Science (PRIMA 2009)*, volume 5925, pages 376–388.

Ryuichiro Higashinaka, Kohji Dohsaka, Shigeaki Amano, and Hideki Isozaki. 2007a. Effects of quiz-style information presentation on user understanding. In *Proc. of Interspeech 2007*, pages 2725–2728.

Ryuichiro Higashinaka, Kohji Dohsaka, and Hideki Isozaki. 2007b. Learning to rank definitions to generate quizzes for interactive information presentation. In *Proc. of ACL 2007 (Poster Presentation)*, pages 117–120.

Takaaki Hori, Chiori Hori, Yasuhiro Minami, and Atsushi Nakamura. 2007. Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15:1352–1365.

Toshihiko Itoh, Norihide Kitaoka, and Ryota Nishimura. 2009. Subjective experiments on influence of response timing in spoken dialogues. In *Proc. of Interspeech 2009*, pages 1835–1838.

W. Lewis Johnson, Jeff W. Rickel, and James C. Lester. 2002. Animated pedagogical aqgents: face-to-face interaction in interactive learning environments. *International Journal of Artificial Intelligence in Education*, 11:47–78.

Hajime Kimura and Shigenobu Kobayashi. 1998. Reinforcement learning for continuous action using stochastic gradient ascent. In *Proc. of the 5th International Conference on Intelligent Autonomous Systems*, pages 288–295.

Nate Kohl and Peter Stone. 2004. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proc. of ICRA 2004*, volume 3, pages 2619–2624.

Stefan Kopp, Lars Gesellensetter, Nicole C. Krämer, and Ipke Wachsmuth. 2005. A conversational agent as museum guide: design and evaluation of a real-world application. In *Lecture Notes in Computer Science (IVA 2009)*, volume 3661, pages 329–343.

Yasuhiro Minami, Minako Sawaki, Kohji Dohsaka, Ryuichiro Higashinaka, Kentaro Ishizuka, Hideki Isozaki, Tatsushi Matsubayashi, Masato Miyoshi, Atsushi Nakamura, Takanobu Oba, Hiroshi Sawada, Takeshi Yamada, and Eisaku Maeda. 2007. The World of Mushrooms: human-computer interaction prototype systems for ambient intelligence. In *Proc. of ICMI 2007*, pages 366–373.

Yasuhiro Minami, Akira Mori, Ryuichiro Higashinaka, Kohji Dohsaka, and Eisaku Maeda. 2009. Dialogue control algorithm for ambient intelligence based on partially observable Markov decision processes. In *Proc. of IWSDS 2009*.

Noriaki Mitsunaga, Christian Smith, Takayuki Kanda, Hiroshi Isiguro, and Norihiro Hagita. 2005. Human-robot interaction based on policy gradient reinforcement learning. In *Proc. of IROS 2005*, pages 1594–1601.

Clifford Nass and Scott Brave. 2005. *Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship*. The MIT Press.

Sharon Oviatt, Courtney Darves, and Rachel Coulston. 2004. Toward adaptive conversational interfaces: modeling speech convergence with animated personas. *ACM Transactions on Computer-Human Interaction*, 11(3):300–328.

Daniel C. Richardson and Rick Dale. 2005. Looking to understand: the coupling between speakers' and listeners' eye movements and its relationship to discourse comprehension. *Cognitive Science*, 29:1045–1060.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking in conversation. *Language*, 50:696–735.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pages 1057–1063.

Adriana Tapus and Maja J. Matarić. 2007. Hands-off therapist robot behavior adaptation to user personality for post-stroke rehabilitation therapy. In *Proc. of 2007 IEEE International Conference on Robotics and Automation*, pages 1547–1553.

David Traum and Jeff Rickel. 2002. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proc. of AAMAS 2002*, pages 766–773.

Roel Vertegaal and Yaping Ding. 2002. Explaining effects of eye gaze on mediated group conversations: amount or synchronization. In *Proc. of CSCW 2002*, pages 41–48.

Jason D. Williams and Steve Young. 2007. Partially observable Markov decision processes for spoken dialog systems. *Computer & Speech Language*, 21(2):393–422.

Jun Zheng, Xiang Yuan, and Yam San Chee. 2005. Designing multiparty interaction support in Elva, an embodied tour guide. In *Proc. of AAMAS 2005*, pages 929–936.

Towards an Empirically Motivated Typology of Follow-Up Questions: The Role of Dialogue Context

Manuel Kirschner and **Raffaella Bernardi**
KRDB Centre, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
{kirschner,bernardi}@inf.unibz.it

Abstract

A central problem in Interactive Question Answering (IQA) is how to answer Follow-Up Questions (FU Qs), possibly by taking advantage of information from the dialogue context. We assume that FU Qs can be classified into specific types which determine if and how the correct answer relates to the preceding dialogue. The main goal of this paper is to propose an empirically motivated typology of FU Qs, which we then apply in a practical IQA setting. We adopt a supervised machine learning framework that ranks answer candidates to FU Qs. Both the answer ranking and the classification of FU Qs is done in this framework, based on a host of measures that include shallow and deep inter-utterance relations, automatically collected dialogue management meta information, and human annotation. We use Principal Component Analysis (PCA) to integrate these measures. As a result, we confirm earlier findings about the benefit of distinguishing between topic shift and topic continuation FU Qs. We then present a typology of FU Qs that is more fine-grained, extracted from the PCA and based on real dialogue data. Since all our measures are automatically computable, our results are relevant for IQA systems dealing with naturally occurring FU Qs.

1 Introduction

When real users engage in written conversations with an Interactive Question Answering (IQA) system, they typically do so in a sort of dialogue rather than asking single shot questions. The questions' context, i.e., the preceding interactions, should be useful for understanding Follow-Up Questions (FU Qs) and helping the system pinpoint the correct answer. In previous work (Kirschner et al., 2009; Bernardi et al., 2010; Kirschner, 2010), we studied how dialogue context should be considered to answer FU Qs. We have used Logistic Regression Models (LRMs), both for learning which aspects of dialogue structure are relevant to answering FU Qs, and for comparing the accuracy with which the resulting IQA systems can correctly answer these questions. Unlike much of the related research in IQA, which used artificial collections of user questions, our work has been based on real user-system dialogues we collected via a chatbot-inspired help-desk IQA system deployed on the web site of our University Library.

Previously, our experiments used a selection of shallow (Kirschner et al., 2009) and deep (Bernardi et al., 2010) features, all of which describe specific relations holding between two utterances (i.e., user questions or system answers). In this paper we present additional features derived from automatically collected dialogue meta-data from our chatbot's dialogue management component. We use Principal Component Analysis (PCA) to combine the benefits of all these information sources, as opposed to using only certain hand-selected features as in our previous work.

The main goal of this paper is to learn from data a new typology of FU Qs; we then compare it to an existing typology based on hand-annotated FU Q types, as proposed in the literature. We show how this new typology is effective for finding the correct answer to a FU Q. We produce this typology by analyzing the main components of the PCA.

This paper presents two main results. A new, empirically motivated typology of FU Qs confirms earlier results about the practical benefit of distinguishing between topic continuation and topic shift FU Qs, which are typically based on hand annotation. We then show that we can do without such hand annotations, in that our fully automatic,

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 322–331,
The University of Tokyo, September 24-25, 2010.

on-line measures – which include automatically collected dialogue meta-data from our chatbot's dialogue manager – lead to better performance in identifying correct answers to FU Qs.

In the remainder of this paper, we first review relevant previous work concerning FU Q typologies in IQA. Section 3 then introduces our collection of realistic IQA dialogues which we will use in all our experiments; the section includes descriptions of meta information in the form of dialogue management features and post-hoc human annotations. In Section 4 we introduce our experimental framework, based on inter-utterance features and LRMs. Our experimental results are presented in Section 5, which is followed by our conclusions.

2 Related work

Much of previous work on dialogue processing in the domain of contextual or interactive Question Answering (QA) (Bertomeu, 2008; van Schooten et al., 2009; Chai and Jin, 2004; Yang et al., 2006) has been based on (semi-)artificially devised sets of context questions. However, the importance of evaluating IQA against *real* user questions and the need to consider preceding system answers has already been emphasized (Bernardi and Kirschner, 2010). The corpus of dialogues we deal with consists of real logs in which actual library users were conversing (by typing) with a chat-bot to obtain information in a help-desk scenario.

(Yang et al., 2006) showed that shallow similarity features between a FU Q and the preceding utterances are useful to determine whether the FU Q is a continuation of the on-going topic ("topic continuation"), or it is a "topic shift". The authors showed that recognizing these two basic types of FU Qs is important for deciding which context fusion strategies to employ for retrieving the answer to the FU Q. (Kirschner et al., 2009) showed how shallow measures of lexical similarity between questions and answers in IQA dialogues are as effective as manual annotations for distinguishing between these basic FU Q types. However, that earlier work was based on a much smaller set of dialogue data than we use in this paper, making for statistically weaker results. (Bernardi et al., 2010) improved on this approach by increasing the data set, and adding "deep" features that quantify text coherence based on different theories of dialogue and discourse structure. However, FU Q classification was performed using either single, hand-selected shallow or deep features, or a hand-selected combination of one shallow and one deep feature. In this paper, we adopt the most promising measures of similarity and coherence from the two aforementioned papers, add new features based on automatically collected dialogue management meta-data, and combine all this information via Principal Component Analysis (PCA). By using PCA, we circumvent the theoretical problem that potentially multicollinear features pose to our statistical models, and at the same time we have a convenient means for inducing a new typology of FU Qs from our data, by analyzing the composition of the principal components of the PCA.

More fine-grained typologies of FU Qs have been suggested, and different processing strategies have been proposed for the identified types. In this paper, we start from our own manual annotation of FU Qs into four basic classes, as suggested by the aforementioned literature (Bertomeu, 2008; van Schooten et al., 2009; Sun and Chai, 2007). We then compare it to our new PCA-based FU Q typology.

3 Data

We now introduce the set of IQA dialogue data which we will use in our experiments. For the purpose of calculating inter-utterance features within these user-system interactions – as described in Section 4.4 – we propose to represent utterances in terms of *dialogue snippets*. A dialogue snippet, or *snippet* for short, contains a FU Q, along with a 2-utterance window of the preceding dialogue context. In this paper we use a supervised machine learning approach for evaluating the correctness of a particular answer to a FU Q; we thus represent also the answer candidate as part of the snippet. Introducing the naming convention we use throughout this paper, a snippet consists of the following four successive utterances: Q_1, A_1, Q_2, and A_2. The FU Q is thus referred to as Q_2.

The data consists of 1,522 snippets of 4-turn human-machine interactions in English: users ask questions and the system answers them. The data set was collected via the Bolzano Bot (BoB) web application that has been working as an on-line virtual help desk for the users of our University Library since October 2008.[1] The snippets were

[1] `www.unibz.it/library`. More information on the BoB dialogue corpus: `bob.iqa-dialogues.net`.

extracted from 916 users' interactions.

Table 3 shows three example dialogue snippets with correct A_1 and A_2; these examples are meant to give an idea of the general shape of the BoB dialogue data. In the third example snippet, A_1 and A_2 actually contain clickable hyperlinks that open an external web-site. We represent them here as dots in parentheses.

Our library domain experts manually checked that each FU Q was either correctly answered in the first place by BoB, or they corrected BoB's answer by hand, by assigning to it the correct answer from BoB's answer repository. In this way, the dialogue data contain 1,522 FU Qs, along with their respective contexts (Q_1 and A_1) and their *correct* answers (A_2). The resulting set of correct A_2s contains 306 unique answers.[2]

The BoB dialogue data also contain two levels of meta information that we will use in this paper. On the one hand, we have automatically collected dialogue meta-data from BoB's dialogue manager that describe the internal state of the BoB system when a FU Q was asked; this information is described in Section 4.2. On the other hand, 417 of the 1,522 FU Qs were hand-annotated regarding FU Q type, as described in Section 4.3.

4 Model

Our goal is, given a FU Q (Q_2 in our dialogue snippets), to pick the best answer from the fixed candidate set of 306 A_2s, by assigning a score to each candidate, and ranking them by this score. Different FU Q types might require different answer picking strategies. Thus, we specify both *A_2 (identification) features*, aiming at selecting the correct A_2 among candidates, and *context (identification) features*, that aim at characterizing the context. The A_2 identification features measure the similarity or coherence between an utterance in the context (e.g., Q_2) and a candidate A_2. Context features measure the similarity or coherence between pairs of utterances in the context (e.g., Q_1 and Q_2). They do not provide direct information about A_2, but might cue a special context (say, an instance of topic shift) where we should pay more attention to different A_2 identification features (say, less attention to the relation between Q_2 and A_2, and more to the one between A_1 and A_2).

We implement these ideas by estimating a generalized linear model from training data to predict the probability that a certain A_2 is correct given the context. In this model, we enter A_2 features as main effects, and context features in interactions with the former, allowing for differential weight assignment to the same A_2 features depending on the values of the context features.

4.1 Logistic Regression

Logistic regression models (LRMs) are generalized linear models that describe the relationship between features (independent variables) and a binary outcome (Agresti, 2002). LRMs are closely related to Maximum Entropy models, which have performed well in many NLP tasks. A major advantage of using logistic regression as a supervised machine learning framework (as opposed to other, possibly better performing approaches) is that the learned coefficients are easy to interpret and assess in terms of their statistical significance. The logistic regression equations specify the probability for a particular answer candidate A_2 being correct, depending on the β coefficients (representing the contribution of each feature to the total answer correctness score), and the feature values $x_1, \ldots, x_k$. In our setting, we are only interested in the *rank* of each A_2 among all answer candidates, which can be easily and efficiently calculated through the linear part of the LRM: score $= \beta_1 x_1 + \ldots + \beta_k x_k$.

FU Q typology is implicitly modeled by *interaction* terms, given by the product of an A_2 feature and a context feature. An interaction term provides an extra β to assign a differential weight to an A_2 feature depending on the value(s) of a context feature. In the simplest case of interaction with a binary 0-1 feature, the interaction β weight is only added when the binary feature has the 1-value.

As described in (Kirschner, 2010), we estimate the model parameters (the beta coefficients $\beta_1, \ldots, \beta_k$) using maximum likelihood estimation. Moreover, we put each model we construct under trial by using an iterative backward elimination procedure that keeps removing the least significant predictor from the model until a specific stopping criterion that takes into account the statistical goodness of fit is satisfied. All the results

[2] Many of the 306 answer candidates overlap semantically. This is problematic, given that our evaluation approach assumes exactly *one* candidate to be correct, while all other 305 answers to be wrong. In this paper, we shall accept this fact, for the merit of simplicity.

we report below are obtained with models that underwent this trimming procedure.

There is a potential pitfall when using multiple regression models such as LRMs with multicollinear predictors, i.e., predictors that are intercorrelated, such as our alternative implementations of inter-utterance string similarity. In such situations, the model may not give valid results about the importance of the individual predictors. In this paper, we use PCA to circumvent the problem by combining potentially multicollinear predictors to completely uncorrelated PC-based predictors.

In the following three sections, we describe the different types of information that are the basis for our features.

4.2 BoB dialogue management meta-data

When BoB interacts with a user, it keeps log files of the IQA dialogue. First of all, these logs include a timed protocol of user input and BoB's responses: the user and system utterances are the literal part of the information. On the other hand, BoB also logs two dimensions of meta information, both of which are based on BoB's internal status of its *dialogue management* routine. This routine is based on a main *initiative-response* loop, mapping user input to some canned-text answer, where the user input should be matched by (at least) one of a set of hand-devised regular expression question patterns.

Sub-dialogues Whenever BoB asks a system-initiated question, the main loop is suspended, and the system goes into a *sub-dialogue* state, where it waits for a specific response from the user – typically a short answer indicating the user's choice about one of the options suggested by BoB. The next user input is then matched against a small number of regular expression patterns specifically designed for the particular system-intiative question at hand. Depending on this user input, the sub-dialogue can:

`Continue:` the user input matched one of the regular expression patterns intended to capture possible user choices

`Break:` the user broke the sub-dialogue by entering something unforeseen, e.g., a new question

The first two parts of Table 4 give an overview of the statistics of BoB's dialogue management-based meta information concerned with sub-dialogue status. Besides *continue* and *break*, for Q_1 we consider also a third, very common case that a user question was not uttered in a sub-dialogue setting at all. Note that we excluded from our data collection all those cases where Q_2 continues a sub-dialogue from our collection of IQA dialogues, since we do not consider such Q_2s as FU Qs, as they are highly constrained by the previous dialogue.

Apology responses The third part of Table 4 gives statistics of whether a particular system response A_1 was an apology message stating that BoB did not understand the user's input, i.e., none of BoB's question patterns matched the user question.

4.3 Manual dialogue annotation

We now turn to the meta information in BoB dialogue data that stems from post-hoc human annotation. For a portion of BoB's log files, we added up to two additional levels of meta information, by annotating the log files after they were collected.[3]

The following paragraphs explain the individual levels of annotation by giving the corresponding annotator instructions; Table 5 contains an overview of the corresponding features. First of all, we annotated FU Qs with their FU Q type. Our choice of the particular four levels of the `FUQtype` feature was influenced by the following literature literature: from (De Boni and Manandhar, 2005) and (Yang et al., 2006) we adopted the distinction between topic shift and topic continuation, while from (Bertomeu et al., 2006) we took the notions of rephrases and context dependency. Our annotation scheme is described in Figure 1; note that topic continuations have three sub-types, which are spelled out below.

`FUQtype = isTopicShift:` marks a FU Q as a *topic shift* based on an intuitive notion of whether the FU Q "switches to something completely different".

`FUQtype = isRephrase:` marks whether the FU Q is an attempt to re-formulate the same question. The FU Q could be a literal repetition of the previous question, or it could be a rephrasing.

`FUQtype = isContextDepentFUQ:` marks whether the FU Q needs to be considered along with some information provided by

[3] All annotations were performed by either one of the authors.

the dialogue context in order to be correctly understood.

`FUQtype = isFullySpecifiedFUQ`: marks whether the FU Q does not need any information from the dialogue context in order to be correctly understood.

The second level of hand-annotation concerns a manual check of the correctness of A_1. It is available for 1,179 of our 1,522 snippets.

`A1.isAnswer.correct`: marks whether the system response is correct for the given question.

`A1.isApology.correct`: marks whether BoB's apology message is correct for the given question.

4.4 Shallow/deep inter-utterance relations

We exploit shallow features, which measure the similarity between two utterances within a snippet, and deep features, which encode coherence between two utterances based on linguistic theory. For each feature we will use names encoding the utterances involved; e.g., `distsim.A1.Q2` stands for the Distributional Similarity feature calculated between A_1 and Q_2.

Shallow features The detailed description of all the shallow features we used in our experiments can be found in (Kirschner et al., 2009). The intuition is that a high similarity between Q and A tends to indicate a correct answer, while in the case of high similarity between the dialogue context and the FU Q, it indicates a "topic continuation" FU Q (as opposed to a "topic shift" FU Q), and thus helps discriminating these two classes of FU Qs.

`Lexical Similarity (lexsim)`: If two utterances share some terms, they are similar; the more *discriminative* the terms they share, the more similar the utterances. Implements a TF-IDF-based similarity metric. `Distributional Similarity (distsim.svd)`: Two utterances are similar not only if they share the same terms, but also if they share similar terms (e.g., *book* and *journal*). Term similarity is estimated on a corpus, by representing each content word (noun, verb, adjective) as a vector that records its corpus co-occurrence with other content words within a 5-word span. `Action sequence (action)`: Based on the notion that in our helpdesk setting we are dealing with task-based dialogues, which revolve around library-related actions (e.g., "borrow", "search"). The action feature indicates whether two utterances contain the same action.

Deep features These features encode different theories of discourse and dialogue coherence. Refer to (Bernardi et al., 2010) for a full description of all deep features we used experimentally, along with more details on the underlying linguistic theories, and our implementation choices for these features.

We introduce a four-level feature, `center`, that encodes the four transitions holding between adjacent utterances that Centering Theory describes (Brennan et al., 1987; Grosz et al., 1995). Somewhat differently from that classic theory, (Sun and Chai, 2007) define the transitions depending on whether both the head and the modifier of the Noun Phrases (NP) representing the *preferred centers*[4] are continued (`cont`) or switched (rough shift: `roughSh`) between Q_1 and Q_2. The remaining two transitions are defined in similar terms.

4.5 PCA-based context classification features

Principal Component Analysis (PCA) (Manly, 2004) is a statistical technique for finding patterns in high-dimensional data, or for reducing their dimensionality. Intuitively, PCA rotates the axes of the original data dimensions in such a way that few of the new axes already cover a large portion of the variation in the data. These few new axes are represented by the so-called principal components (PCs). We employ this technique as a tool for combining a multitude of potentially multicollinear predictors for context classification, i.e., all predictors that involve Q_2 and some preceding utterance. In our experiments we will also want to look at the correlations of each of the top PCs with the original context classification features; these correlations are called *loadings* in PCA. We experiment with the following three versions of PCA:

PCA$_A$: without BoB dialogue management meta-data features PCA performed over all context classification features of the shallow and deep types described in Section 4.4.

[4]Centers are noun phrases. The syntactic structure of a noun phrase comprises a *head noun*, and possibly a *modifier*, e.g., an adjective. We use a related approach, described in (Ratkovic, 2009), to identify the *preferred center* of each question.

PCA$_B$: with BoB dialogue management meta-data features PCA$_A$ plus BoB's dialogue-management meta-data features (Section 4.2).

PCA$_C$: with BoB dialogue management meta-data features and manual A_1 correctness check PCA$_B$ plus additional manual annotation of A_1 correctness (Section 4.3).

5 Evaluation

We employ a standard 10-fold cross-validation scheme for splitting training and prediction data. We assess our LRMs by comparing the ranks that the models assign to the gold-standard correct A_2 candidate (i.e., the single A_2 that our library domain experts had marked as correct for each of the 1,522 FU Qs). To determine whether differences in A_2 ranking performance are significant, we consult both the paired t-test and the Wilcoxon signed rank test about the difference of the 1,522 ranks.

5.1 Approximating hand-annotated FU Q types with PCA-based features

We begin the evaluation of our approach by exploring the value of the hand-annotation-based FU Q type as cues for expressing the relevance and topical relatedness of that particular FU Q's dialogue context.

For this purpose, we use the subset of 417 dialogue snippets which we annotated with the `FUQtype` feature described in the first half of Table 5. Figure 1 depicts our FU Q type taxonomy, and the distribution of the four types in our data.

First of all, for this hand-annotated subset of dialogue snippets, we try to improve the A_2 ranking results of a "main effects only" **baseline** LRM, i.e., a model which does not distinguish between different FU Q types. This baseline model was proposed in earlier work (Kirschner et al., 2009). We tried the following features as interaction term(s) in our models, one after the other: whether the hand-annotated `FUQType` feature indicates a topic shift or not; the full four levels of `FUQType`; a linear combination of the top five PCs of each of the three PCA feature sets introduced in Section 4.5. After applying our automatic predictor elimination routine described in Section 4.1 and evaluating the A_2 ranking results of each of these models, none of the interactive models significantly outperform our baseline. PCA-based context classification using only fully automatic BoB meta information features (PCA$_B$ in Section 4.5) results in the largest improvement over baseline; however, this improvement does not reach statistical significance, most likely due to the small data set of only 417 cases. Still, using the hand-annotated FU Q type feature `FUQType`, we can visualize how the top PCs cluster the 417 FU Qs, and how this clustering mirrors some of the distinctions of manually assigned FU Q types: see Figure 2. E.g., plotting the FU Qs along their PC1 and PC2 values seems to mimic the annotator's distinction between topic shift FU Qs and the other three FU Q types. The other pairs of PCs also appear to show certain clusters. Overall, the automatic context classification features that served as input to the PCA are useful for describing different context-related behaviors of different FU Qs.

5.2 Optimizing A_2 ranking scores using PCA-based features

Having shown the usefulness (in terms of assigning high ranks to the gold-standard correct A_2) of FU Q classification via a PCA-based combination of purely automatic context classification features, we can now consider the full sample of 1,522 dialogue snippets described in Section 3, for which we do not in general possess manual FU Q type annotations.

The first row of Table 1 shows the A_2 ranking results of our baseline LRM. In the remainder of the table, we compare this baseline model to three different models which use a linear combination of different versions of the top five PCs as interaction terms. The three versions (A, B and C) were introduced in Section 4.5.

5.3 Analysis of PC-based context features

The main goal of this paper is to devise an empirically motivated typology of FU Qs, under consideration of automatically collected dialogue management meta information. We then want to show how this new typology is effective for finding the correct answer to a specific FU Q, in that for the given FU Q it indicates the relevance and topical relatedness of the question's particular dialogue context. In Section 5.2 we have seen how all PCA-based context classification features perform clearly better than a non-interactive baseline model; more specifically, the top five PCs from the PCA$_B$ scheme yield significantly better A_2 ranking results than the PCA$_A$ scheme which does not consider BoB dialogue management meta-data features. Based on these results, we now look in

Model ID	Interaction terms	Mean rank correct A_2	Median rank correct A_2	Standard dev.	p (Paired t-test)	p (Wilcoxon signed rank)
baseline	none	48.72	14	69.35		
PCA_A	$PC1 + \ldots + PC5$	44.25	12	64.58	< 0.0001	< 0.0001
PCA_B	$PC1 + \ldots + PC5$	**42.72**	**12**	62.53	0.0006	0.0087
PCA_C	$PC1 + \ldots + PC5$	42.87	12	62.94	not sig.	not sig.

Table 1: Improving ranking of correct A_2 (out of 306 answer candidates) with different PCA-based interaction terms. Significance tests of rank differences wrt. result in preceding row.

more detail at the relevance of the top five PC features in PCA_B, and at their most important *loadings*, i.e., the original context classification features that are most highly correlated with the value of each particular PC. After running our predictor elimination routine, the corresponding LRM has kept three of these five top PCs as interaction terms: PC1, PC2 and PC5. Table 2 describes the top three positive and top three negative loadings of these PCs. The table also shows how in model PCA_B, each of the interaction terms corresponding to the three PCs influences the score that is calculated for every A_2 candidate, either positively or negatively.

Interpreting the results of Table 2 on a high, dialogue-specific level, we draw the following conclusions:

PC1 seems to capture a rather general distinction of topic shift versus topic continuation. A FU Q with high lexical similarity to the preceding utterances (i.e., a "topic continuation") should preferably get an A_2 with higher lexical similarity with respect to both A_1 and Q_2. In this context, "topic shift" is partly described by a feature from Centering Theory, and two of BoB's dialogue management meta-data features.

PC2 shows relatively weak *positive* correlations with any context classification features. On the negative end, PC2 seems to describe a class of FU Qs that are uttered after a Q_1 that did neither continue nor exit a sub-dialogue. Also, A_1 was a regular system answer (as opposed to an apology message by BoB). Such FU Qs can thus be interpreted as "single shot" questions that a user poses after their previous question was already dealt with in A_1. Because of the negative loadings, the value of PC2 becomes negative, resulting in the *avoidance* of any A_2 that is highly similar to the preceding A_1.

PC5 distinguishes FU Qs that are mostly related to the previous answer from those that are more related to the previous question. Depending on whether PC5 turns positive or negative, A_2s are preferred that are more similar to A_1 or Q_2, respectively. $Q_1.Q_2$ similarity is determined by both lexical similarity and Centering Theory features.

6 Conclusion

In this paper we have experimentally explored the problem of FU Q types and their corresponding answer identification strategies. The first result is that our hand-annotated FU Q types did not significantly improve Q_2 answering performance (for the annotated sub-set of 417 snippets). We attribute this negative result in part to the difficulty of the 4-level FU Q type annotation task. On the other hand, we believe it is encouraging that with purely automatic features for context classification, combined through PCA, we significantly outperformed our baseline. Adding BoB's dialogue management meta information – which is also automatically available when using our dialogue collection scheme – for context classification helped improve the scores even further. We analyzed the top loadings of three PCs that our best-performing LRM uses for FU Q type classification. We used PCA both for circumventing the problem of multicollinear predictors in LRM, and as a diagnostic tool to analyze the most important components of automatically combined FU Q classification features. Finally, a potentially difficult and cumbersome manual annotation of the correctness of the previous system answer A_1 did not improve A_2 ranking performance.

References

Alan Agresti. 2002. *Categorical Data Analysis*. Wiley-Interscience, New York.

Raffaella Bernardi and Manuel Kirschner. 2010. From

LOADINGS

PC1		PC2		PC5	
0.33	distsim.Q1.Q2	0.05	Q1.bob.contSubdial	0.45	distsim.A1.Q2
0.26	distsim.A1.Q2	0.04	Q2.center.roughSh	0.31	A1.bob.isApology
0.26	action.Q1.Q2	0.02	Q2.bob.breakSubdial	0.29	lexsim.A1.Q2
⋮		⋮		⋮	
−0.13	A1.bob.isApology	−0.22	A1.bob.isAnswer	−0.18	lexsim.Q1.Q2
−0.15	Q2.bob.noSubdial	−0.30	Q2.bob.noSubdial	−0.23	Q2.center.cont
−0.22	Q2.center.roughSh	−0.31	Q1.bob.noSubdial	−0.26	A1.bob.isAnswer

INFLUENCE ON A_2 SELECTION IN MODEL PCA_B

pos for each A_2 similar to Q_2	**pos** for each A_2 similar to A_1	**pos** for each A_2 similar to A_1
pos for each A_2 similar to A_1		**neg** for each A_2 similar to $Q2$

Table 2: Strongest loadings for the three PCs retained as interaction terms in Model PCA_B, and indication of each PC's positive/negative influence on lexical similarity-based A_2 selection features

artificial questions to real user interaction logs: Real challenges for interactive question answering systems. In *Proc. of Workshop on Web Logs and Question Answering (WLQA'10)*, Valletta, Malta.

Raffaella Bernardi, Manuel Kirschner, and Zorana Ratkovic. 2010. Context fusion: The role of discourse structure and centering theory. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Núria Bertomeu, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. 2006. Contextual phenomena and thematic relations in database QA dialogues. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8, New York, NY.

Nuria Bertomeu. 2008. *A Memory and Attention-Based Approach to Fragment Resolution and its Application in a Question Answering System*. Ph.D. thesis, Department of Computational Linguistics, Saarland University.

Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 155–162, Stanford, California.

Joyce Y. Chai and Rong Jin. 2004. Discourse structure for context question answering. In *Proc. of the HLT-NAACL 2004 Workshop on Pragmatics in Question Answering*, Boston, MA.

Marco De Boni and Suresh Manandhar. 2005. Implementing clarification dialogues in open domain question answering. *Journal of Natural Language Engineering*, 11(4):343–361.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Manuel Kirschner, Raffaella Bernardi, Marco Baroni, and Le Thanh Dinh. 2009. Analyzing Interactive QA dialogues using Logistic Regression Models. In *Proc. of XIth International Conference of the Italian Association for Artificial Intelligence (AI*IA)*, Reggio Emilia, Italy.

Manuel Kirschner. 2010. *The Structure of Real User-System Dialogues in Interactive Question Answering*. Ph.D. thesis, Free University of Bozen-Bolzano, Italy.

Bryan F. J. Manly. 2004. *Multivariate Statistical Methods: A Primer*. Chapman & Hall/CRC, 3rd edition.

Zorana Ratkovic. 2009. Deep analysis in iqa: evaluation on real users' dialogues. Master's thesis, European Masters Program in Language and Communication Technologies.

Mingyu Sun and Joyce Y. Chai. 2007. Discourse processing for context question answering based on linguistic knowledge. *Know.-Based Syst.*, 20(6):511–526.

Boris van Schooten, R. op den Akker, R. Rosset, O. Galibert, A. Max, and G. Illouz. 2009. Follow-up question handling in the IMIX and Ritel systems: A comparative study. *Journal of Natural Language Engineering*, 15(1):97–118.

Fan Yang, Junlan Feng, and Giuseppe Di Fabbrizio. 2006. A data driven approach to relevancy recognition for contextual question answering. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 33–40, New York City, NY.

Q_1	where in the library can i find books
A_1	Most of our holdings are accessible in the open-shelves area. In the repository we only keep some older volumes of journals or the outsize stuff.
Q_2	shelf mark
A_2	Shelf marks are a combination of letters and numbers, which exactly define the location of a book in a Library. If you find a book or a journal in the catalogue, you also see the shelf mark. The shelf marks of the University Library consist of an alphanumeric code (e.g. ES 560 K19). Would you like to know how you search them?
Q_1	When can I return a book?
A_1	Books and other media can be consigned at the information desk during the regular opening hours of the Library. You can also leave them at the book drop located in front of the main entrance of the university building - 24 hours a day.
Q_2	I want to read the newspaper on the computer
A_2	If you haven't got a Library Card you can visit the Library to read books, journals or newspapers. If you want to borrow something or use the other services we offer you should get a Library Card. Do you want to know how it works?
Q_1	i need the opac
A_1	Here (...) is the page with our OPAC for you. You can search the complete holdings of the Library.
Q_2	where is the advisory service
A_2	The University's Advisory Service provides an information service for prospective students who would like to know more about which degree course to study. It also provides a support service for enrolled students during their entire time at the University. You can get further information from this (...) site.

Table 3: Example dialogue snippets with correctly identified A_2

Feature name	Freq.	Description
`Q1.bob.contSubdial`	7.6%	Q_1 continues system-initiated sub-dialogue
`Q1.bob.breakSubdial`	9.6%	Q_1 breaks out of system-initiated sub-dialogue
`Q1.bob.noSubdial`	82.9%	BoB not in sub-dialogue mode when Q_1 was uttered
`Q2.bob.breakSubdial`	13.6%	Q_2 breaks out of system-initiated sub-dialogue
`Q2.bob.noSubdial`	86.4%	BoB not in sub-dialogue mode when Q_2 was uttered
`A1.bob.isAnswer`	75.6%	A_1 is regular answer retrieved by BoB
`A1.bob.isApology`	24.4%	A_1 is apology message: BoB did not understand

Table 4: BoB dialogue management meta information. Proportions out of those 1,441 of total 1,522 snippets for which this information was logged.

Feature name	Freq.	Description
`FUQtype=isTopicShift`	40.0% (of 417)	Q_2 is topic shift
`FUQtype=isRephrase`	19.2% (of 417)	Q_2 is rephrasing of Q_1
`FUQtype=isContextDepentFUQ`	6.5% (of 417)	Q_2 is context dependent
`FUQtype=isFullySpecifiedFUQ`	34.3% (of 417)	Q_2 is not context dependent
`A1.isAnswer.correct`	66.5% (of 1,179)	BoB's regular answer A_1 is correct
`A1.isAnswer.false`	19.0% (of 1,179)	BoB's regular answer A_1 is false
`A1.isApology.correct`	1.3% (of 1,179)	BoB's apology message A_1 is correct
`A1.isApology.false`	13.2% (of 1,179)	BoB's apology message A_1 is false

Table 5: Manual annotation meta information. Proportions out of those sub-sets of total 1,522 snippets with available annotation.

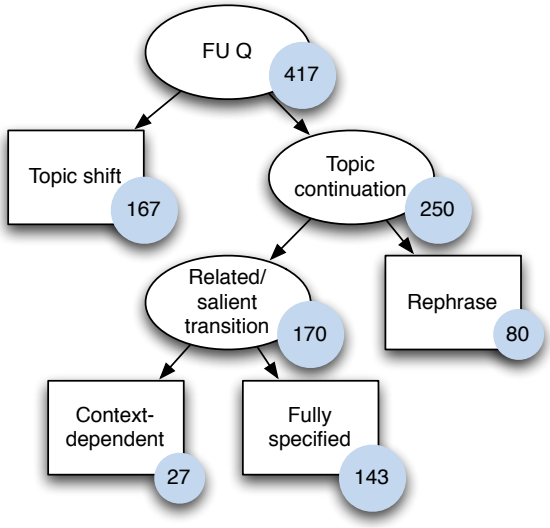


Figure 1: Manual FU Q type annotation scheme, with counts of FU Q types

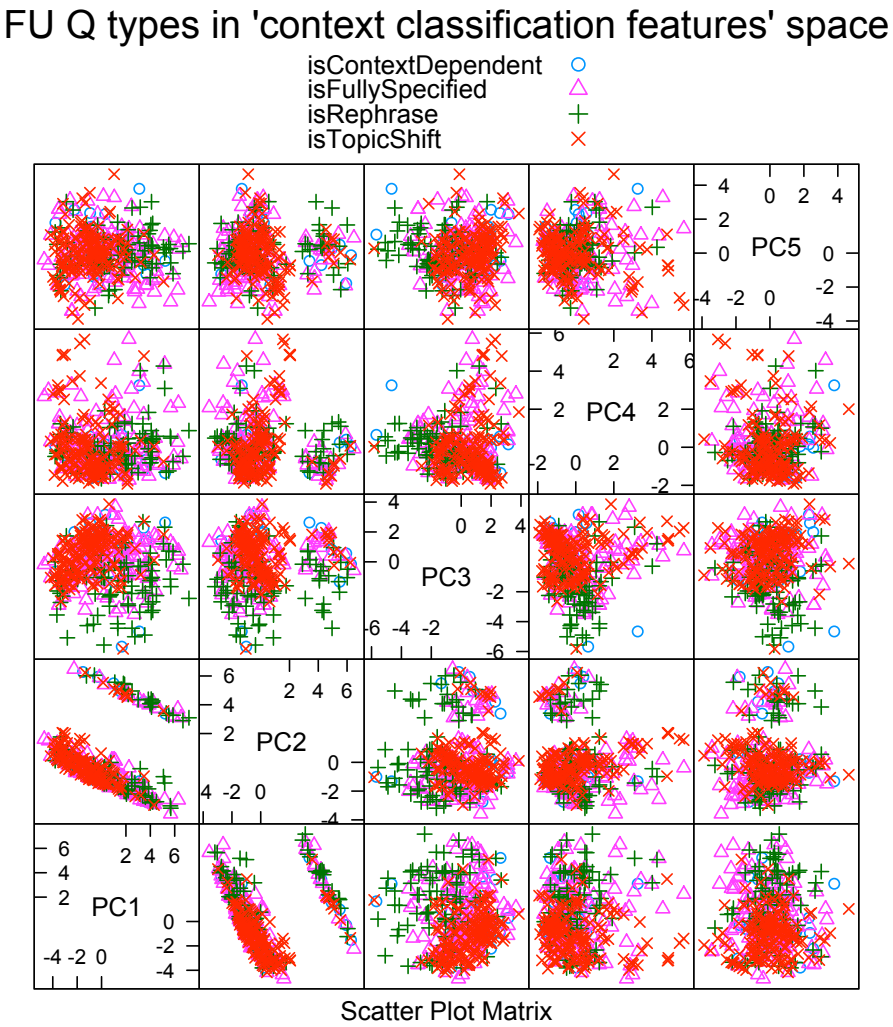


Figure 2: Distribution of hand-annotated FU Q types in PC-based feature space (PCA_B)

Assessing the effectiveness of conversational features for dialogue segmentation in medical team meetings and in the AMI corpus

Saturnino Luz
Department of Computer Science
Trinity College Dublinm Ireland
luzs@cs.tcd.ie

Jing Su
School of Computer Science and Statistics
Trinity College Dublin, Ireland
sujing@scss.tcd.ie

Abstract

This paper presents a comparison of two similar dialogue analysis tasks: segmenting real-life medical team meetings into patient case discussions, and segmenting scenario-based meetings into topics. In contrast to other methods which use transcribed content and prosodic features (such as pitch, loudness etc), the method used in this comparison employs only the duration of the prosodic units themselves as the basis for dialogue representation. A concept of Vocalisation Horizon (VH) allows us to treat segmentation as a classification task where each instance to be classified is represented by the duration of a talk spurt, pause or speech overlap event in the dialogue. We report on the results this method yielded in segmentation of medical meetings, and on the implications of the results of further experiments on a larger corpus, the Augmented Multiparty Meeting corpus, to our ongoing efforts to support data collection and information retrieval in medical team meetings.

1 Introduction

As computer mediated communication becomes more widespread, and data gathering devices start to make their way into the meeting rooms and the workplace in general, the need arises for modelling and analysis of dialogue and human communicative behaviour in general (Banerjee et al., 2005). The focus of our interest in this area is the study of multi-party interaction at Multidisciplinary Medical Team Meeting (MDTMs), and the eventual recording of such meetings followed by indexing and structuring for integration into electronic health records. MDTMs share a number of characteristics with more conventional business meetings, and with the meeting scenarios targeted in recent research projects (Renals et al., 2007). However, MDTMs are better structured than these meetings, and more strongly influenced by the time pressures placed upon the medical professionals who take part in them (Kane and Luz, 2006).

In order for meeting support and review systems to be truly effective, they must allow users to efficiently browse and retrieve information of interest from the recorded data. Browsing in these media can be tedious and time consuming because continuous media such as audio and video are difficult to access since they lack natural reference points. A good deal of research has been conducted on indexing recorded meetings. From a user's point of view, an important aspect of indexing continuous media, and audio in particular, is the task of structuring the recorded content. Banerjee et al. (2005), for instance, showed that users took significantly less time to retrieve answers when they had access to discourse structure annotation than in a control condition in which they had access only to unannotated recordings.

The most salient discourse structure in a meeting is the topic of conversation. The content within a given topic is cohesive and should therefore be viewed as a whole. In MDTMs, the meeting consists basically of successive patient case discussions (PCDs) in which the patient's condition is discussed among different medical specialists with the objective of agreeing diagnoses, making patient management decisions etc. Thus, the individual PCDs can be regarded as the different "topics" which make up an MDTM (Luz, 2009).

In this paper we explore the use of a content-free approach to the representation of vocalisation events for segmentation of MDTM dialogues. We start by extending the work of Luz (2009) on a small corpus of MDTM recordings, and then test our approach on a larger dataset, the AMI (Aug-

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 332–339,
The University of Tokyo, September 24-25, 2010.

mented Multi-Party Interaction) corpus (Carletta, 2007). Our ultimate goal is to analyse and apply the insights gained on the AMI corpus to our work on data gathering and representation in MDTMs.

2 Related work

Topic segmentation and detection, as an aid to meeting information retrieval and meeting indexing, has attracted the interest of many researchers in recent years. The objective of topic segmentation is to locate the beginning and end time of a cohesive segment of dialogue which can be singled out as a "topic". Meeting topic segmentation has been strongly influenced by techniques developed for topic segmentation in text (Hearst, 1997), and more recently in broadcast news audio, even though it is generally acknowledged that dialogue segmentation differs from text and scripted speech in important respects (Gruenstein et al., 2005).

In early work (Galley et al., 2003), meeting annotation focused on changes that produce high inter-annotator agreement, with no further specification of topic label or discourse structure. Current work has paid greater attention to discourse structure, as reflected in two major meeting corpus gathering and analysis projects: the AMI project (Renals et al., 2007) and the ICSI meeting project (Morgan et al., 2001). The AMI corpus distinguishes top-level and functional topics such as "presentation", "discussion", "opening", "closing", "agenda" which are further specified into sub-topics (Hsueh et al., 2006). Gruenstein et al. (2005) sought to annotated the ICSI corpus hierarchically according to topic, identifying, in addition, action items and decision points. In contrast to these more general types of meetings, MDTMs are segmented into better defined units (i.e. PCDs) so that inter-annotator agreement on topic (patient case discussion) boundaries is less of an issue, since PCDs are collectively agreed parts of the formal structure of the meetings.

Meeting transcripts (either done manually or automatically) have formed the basis for a number of approaches to topic segmentation (Galley et al., 2003; Hsueh et al., 2006; Sherman and Liu, 2008). The transcript-based meeting segmentation described in (Galley et al., 2003) adapted the unsupervised lexical cohesion method developed for written text segmentation (Hearst, 1997). Other approaches have employed supervised machine learning methods with textual features (Hsueh et al., 2006). Prosodic and conversational features have also been integrated into text-based representations, often improving segmentation accuracy (Galley et al., 2003; Hsueh and Moore, 2007).

However, approaches that rely on transcription, and sometimes higher-level annotation on transcripts, as is the case of (Sherman and Liu, 2008), have two shortcomings which limit their applicability to MDTM indexing. First, manual transcription is unfeasible in a busy hospital setting, and automatic speech recognition of unconstrained, noisy dialogues falls short of the levels of accuracy required for good segmentation. Secondly, the contents of MDTMs are subject to stringent privacy and confidentiality constraints which limit access to training data. Regardless of such application constraints, some authors (Malioutov et al., 2007; Shriberg et al., 2000) argue for the use of prosodic features and other acoustic patterns directly from the audio signal for segmentation. The approach investigated in this paper goes a step further by representing the data solely through what is, arguably, the simplest form of content-free representation, namely: duration of talk spurts, silences and speech overlaps, optionally complemented with speaker role information (e.g. medical speciality).

3 Content-free representations

There is more to the structure (and even the semantics) of a dialogue than the textual content of the words exchanged by its participants. The role of prosody in shaping the illocutionary force of vocalisations, for instance, is well documented (Holmes, 1984), and prosodic features have been used for automatic segmentation of broadcast news data into sentences and topics (Shriberg et al., 2000). Similarly, recurring audio patterns have been employed in segmentation of recorded lectures (Malioutov et al., 2007). Works in the area of social psychology have used the simple conversational features of duration of vocalisations, pauses and overlaps to study the dynamics of group interaction. Jaffe and Feldstein (1970) characterise dialogues as Markov processes, and Dabbs and Ruback (1987) suggest that a "content-free" method based on the amount and structure of vocal interactions could complement group interaction frameworks such as the one proposed by Bales (1950). Pauses and overlap statistics alone can be used, for instance, to characterise

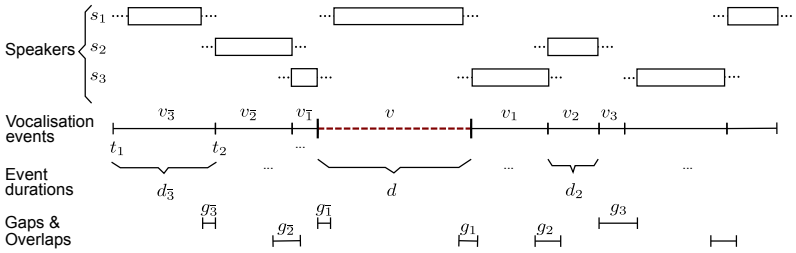


Figure 1: Vocalisation Horizon for event $\boldsymbol{v}$.

the differences between face-to-face and telephone dialogue (ten Bosch et al., 2005), and a correlation between the duration of pauses and topic boundaries has been demonstrated for recordings of spontaneous narratives (Oliveira, 2002).

These works provided the initial motivation for our content-free representation scheme and the topic segmentation method proposed in this paper. It is easy to verify by inspection of both the corpus of medical team meetings described in Section 4 and the AMI corpus that pauses and vocalisations vary significantly in duration and position on and around topic boundaries. Table 1 shows the mean durations of vocalisations that initiate new topics or PCDs in MDTMs and the scenario-based AMI meetings, as well as the durations of pauses and overlaps that surround it (within one vocalisation event to the left and right). In all cases the differences were statistically significant. These results agree with those obtained by Oliveira (2002) for discourse topics, and suggest that an approach based on representing the duration of vocalisations, pauses and overlaps in the immediate context of a vocalisation might be effective for automatic segmentation of meeting dialogues into topics or PCDs.

Table 1: Mean durations in seconds (and standard deviations) of vocalisation and pauses on and near topic boundaries in MDTM and AMI meetings.

	Boundary	Non-boundary	t-test
AMI vocal.	5.3 (8.2)	1.6 (3.5)	$p < .01$
AMI pauses	2.6 (4.9)	1.2 (2.8)	$p < .01$
AMI overlaps	0.4 (0.4)	0.3 (0.6)	$p < .01$
MDTM vocal.	12.0 (15.5)	8.1 (14.7)	$p < .05$
MDTM pauses	9.7 (12.7)	8.2 (14.8)	$p < .05$

We thus conceptualise meeting topic segmentation as a classification task approachable through supervised machine learning. A meeting can be pre-segmented into separate *vocalisations* (i.e. talk spurts uttered by meeting participants) and silences, and such basic units (henceforth referred to as *vocalisation events*) can then be classified as to whether they signal a topic transition. The basic defining features of a vocalisation event are the identity of the speaker who uttered the vocalisation (or speakers, for events containing speech overlap) and its duration, or the duration of a pause, for silence events. However, identity labels and interval durations by themselves are not enough to enable segmentation. As we have seen above, some approaches to meeting segmentation complement these basic data with text (keywords or full transcription) uttered during vocalisation events, and with prosodic features. Our proposal is to retain the content-free character of the basic representation by complementing the speaker and duration information for an event with data describing its preceding and succeeding events. We thus aim to capture an aspect of the dynamics of the dialogue by representing snapshots of vocalisation sequences. We call this general representation strategy *Vocalisation Horizon* (VH).

Figure 1 illustrates the basic idea. Vocalisation events are placed on a time line and combine utterances produced by the speakers who took part in the meeting. These events can be labelled with nominal attributes $(s_1, s_2, \ldots)$ denoting the speaker (or some other symbolic attribute, such as the speaker's role in the meeting). Silences (gaps) and group talk (overlap) can either be assigned reserved descriptors (such as "Floor" and "Group") or regarded as separate annotation layers. The general data representation scheme for, say, segment v would involve a data from its left context $(v_{\bar{1}}, v_{\bar{2}}, v_{\bar{3}}, \ldots)$ and its right context $(v_1, v_2, v_3, \ldots)$ in addition to the data for v itself. These can be a combination of symbolic labels (in Figure 1, for instance, s_1 for the current speaker, $s_3, s_2, s_1, \ldots$ for the preceding events and $s_3, s_2, s_3, \ldots$ for the following events), durations (d, $d_{\bar{1}}, d_{\bar{2}}, d_{\bar{3}}, \ldots$ etc)

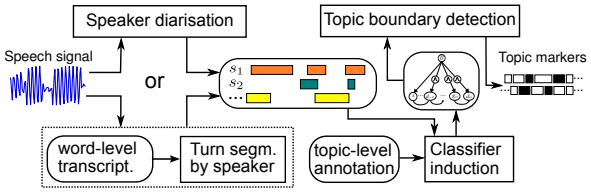


Figure 2: Meeting segmentation processing architecture.

and gaps or overlaps $g_{\bar{1}}, g_{\bar{2}}, g_{\bar{3}}, \ldots, g_1, g_2, g_3, \ldots$ etc). Specific representations depend on the type of annotation available on the speech data and on the nature of the meeting. Sections 4 and 5 present and assess various representation schemes.

The general processing architecture for meeting segmentation assumed in this paper is shown in Figure 2. The system will received the speech signal, possibly on a single channel, and pre-segment it into separate channels (one per speaker) with intervals of speech activity and silence labelled for each stream. Depending on the quality of the recording and the characteristics of the environment, this initial processing stage can be accomplished automatically through existing speaker diarisation methods — e.g. (Ajmera and Wooters, 2003). In the experiments reported below manual annotation was employed. In the AMI corpus, speaker and speech activity annotation is done on the word level and include transcription (Carletta, 2007). We parsed these word-level labels, ignoring the transcriptions, in order to build the content-free representation described above. Once the data representation has been created it is then used, along with topic boundary annotations, to train a probabilistic classifier. Finally, the topic detection module uses the models generated in the training phase to hypothesise boundaries in unannotated vocalisation event sequences and, optionally, performs post-processing of these sequences before returning the final hypothesis. These modules are described in more detail below.

4 MDTM Segmentation

The MDTM corpus was collected over a period of three years as part of a detailed ethnographic study of medical teams (Kane and Luz, 2006). The corpus consists in 28 hours or meetings recorded in a dedicated teleconferencing room at a major primary care hospital. The audio sources included a pressure-zone microphone attached to the teleconferencing system and a highly sensitive directional microphone. Video was gathered through two separate sources: the teleconferencing system, which showed the participants and, at times, the medical images (pathology slides, radiology) relevant to the case under discussion, and a high-end camcorder mounted on a tripod. All data were imported into a multimedia annotation tool and synchronised. Of these, two meetings encompassing 54 PCDs were chosen an annotated for vocalisations (including speaker identity and duration) and PCD boundaries.

Vocalisation events were encoded as vectors $v = (s, d, s_{\bar{1}}, d_{\bar{1}}, \ldots, s_{\bar{n}}, d_{\bar{n}}, s_1, d_1, \ldots, s_n, d_n)$, where the variables are as explained in Section 3. The speaker labels s, $s_{\bar{i}}$ and s_i are replaced, for the sake of generality, by "role" labels denoting medical specialties, such as "radiologist", "surgeon", "clinical oncologist", "pathologist" etc. In addition to these roles, we reserved the special labels "Pause" (a period of silence between two vocalisations by the same speaker), "SwitchingPause" (pause between vocalisations by different speakers), and "Group" (vocalisations containing overlaps, i.e. speech by more than one speaker). We set a minimum duration of 1s for a talk spurt to count as a speech vocalisation event and a 0.9s minimum duration for silence period to be a pause. Shorter intervals (depicted in Figure 1 as the fuzzy ends of the speech lines on the top of the chart) are incorporated into an adjacent vocalisation event.

The segmentation process can be defined as the process of mapping the set of vocalisation events V to $\{0, 1\}$ where 1 represents a topic boundary and 0 represents a non-boundary vocalisation event. In order to implement this mapping we employ a Naive Bayes classifier. The conditional probabilities for the nominal variables (speaker roles) are estimated on the training set by maximum likelihood and combined into multinomial models, while the continuous variables are log transformed and modelled through Gaussian kernels (John and Langley, 1995).

These models are used to estimate the probability, given by equation (1), of a vocalisation being marked as a topic boundary given the above described data representation, and the usual conditional independence assumptions applies.

$$P(B = b|V = v) = P(B = b|S_{\bar{n}} = s_{\bar{n}}, D_{\bar{n}} = d_{\bar{n}}, \ldots, S = s, \ldots, D_n = d_n) \quad (1)$$

The model can therefore be represented as a simple Bayesian network where the only depen-

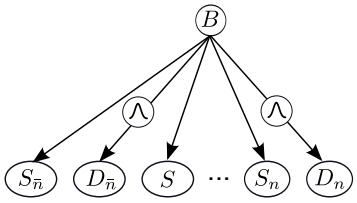


Figure 3: Bayesian model employed for dialogue segmentation.

dencies are between the boundary variable and each feature of the vocalisation event, as shown in Figure 3.

Luz (2009) reports that, for a similar data representation, horizons of length $2 < n < 6$ produced the best segmentation results. Following this finding, we adopt $n = 3$ for all our experiments. We tested two variants of the representation: V_{pd} that discriminated between pause types (pauses, switching pauses, group pauses, and group switching pauses), as in (Dabbs and Ruback, 1987), and V_{sp} which labelled all pauses equally. The evaluation metrics employed include the standard classification metrics of *accuracy* (A), the proportion of correctly classified segments, boundary precision (P), the proportion of correctly assigned boundaries among all events marked as topic boundaries, boundary recall (R), the proportion of target boundaries correctly assigned, and the F_1 score, the harmonic mean of P and R.

Although these standard metrics provide an initial approximation to segmentation effectiveness, they have been criticised as tools for evaluating segmentation because they are hard to interpret and are not sensitive to near misses (Pevzner and Hearst, 2002). Furthermore, due to the highly unbalanced nature of the classification task (boundary vocalisation events are only 3.3% of all instances), accuracy scores tend to produce overoptimistic results. Therefore, to give a fairer picture of the effectiveness of our method, we also report values for two error metrics proposed specifically for segmentation: P_k (Beeferman et al., 1999) and WindowDiff, or *WD*, (Pevzner and Hearst, 2002).

The P_k metric gives the probability that two vocalisation events occurring k vocalisations apart and picked otherwise randomly from the dataset are incorrectly identified by the algorithm as belonging to the same or to different topics. P_k is computed by sliding two pairs of pointers over the reference and the hypothesis sequences and observing whether each pair of pointers rests in the same or in different segments. An error is counted if the pairs disagree (i.e. if they point to the same segment in one sequence and to different segments in the other).

WD is as an estimate of inconsistencies between reference and hypothesis, obtained by sliding a window of length equal k segments over the time line and counting disagreements between true and hypothesised boundaries. Like the standard IR metrics, P_k and *WD* range over the $[0, 1]$ interval. Since they are error metrics, the greater the value, the worse the segmentation.

Table 2: PCD segmentation results for 5-fold cross validation, horizon $\boldsymbol{n = 3}$ (mean values).

Threshold	Filter	Data	A	P	R	F_1	P_k	*WD*
MAP	no	V_{sp}	0.94	0.20	0.21	0.18	0.33	0.44
		V_{pd}	0.95	0.17	0.20	0.16	0.30	0.38
	yes	V_{sp}	0.95	0.20	0.16	0.16	0.32	0.38
		V_{pd}	**0.95**	0.16	0.12	0.13	0.29	0.34
Proport.	no	V_{sp}	0.95	0.28	**0.28**	**0.28**	0.26	0.36
		V_{pd}	0.95	0.26	0.27	0.26	0.27	0.42
	yes	V_{sp}	0.95	**0.30**	0.22	0.25	0.25	**0.31**
		V_{pd}	0.95	0.22	0.14	0.17	0.27	0.33

Table 2 shows the results for segmentation of MDTMs into PCDs under the representational variants mentioned above and two different thresholding strategies: *maximum a posteriori* hypothesis (MAP) and proportional threshold. The latter is a strategy that varies the threshold probability above which an event is marked as a boundary according to the generality of boundaries found in the training set. The motivation for testing proportional thresholds is illustrated by Figure 4, which shows a step plot of MAP hypothesis (h) superimposed on the true segmentation (peaks represent boundaries) and the corresponding values for $p(b|v)$. It is clear that a number of false positives would be removed if the threshold were set above the MAP level[1] with no effect on the number of false negatives.

Another possible improvement suggested by Figure 4 is the *filtering* of adjacent boundary hypotheses. Wider peaks, such as the ones on instances 14 and 172 indicate that two or more boundaries were hypothesised in immediate succession. Since this is clearly impossible, a straightforward improvement of the segmentation

[1]I.e. $p(b|v) > 0.5$; above the horizontal line in the centre.

hypothesis can be achieved by choosing a single boundary marker among a cluster of adjacent ones. This has been done as a post-processing step by choosing a single event with maximal estimated probability within a cluster of adjacent boundary hypotheses as the new hypothesis.

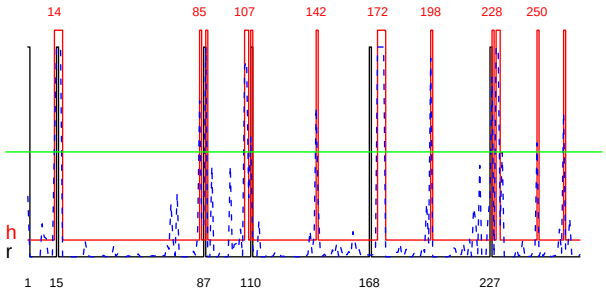


Figure 4: Segmentation profile showing true boundaries (r), boundaries hypothesised by a MAP classifier (h) and probabilities (dotted line).

The results suggest that both proportional thresholding and filtering improve segmentation. As expected, accuracy figures were generally high (an uninformative) reflecting the great imbalance in favour of negative instances and the conservative nature of the classifier. Precision, recall and F_1 (for positive instances only) were also predictably low, with V_{sp} under a proportional threshold attaining the best results. However, in meeting browsing marking the topic boundary precisely is far less important than retrieving the right text is in information retrieval or text categorisation, since the user can easily scan the neighbouring intervals with a slider (Banerjee et al., 2005). Therefore, P_k and *WD* are the most appropriate measures of success in this task. Here our results seem quite encouraging, given that they all represent great improvements over the (rather reasonable) baselines of $P_k = .46$ and $WD = .51$ estimated by Monte Carlo simulation as in (Hsueh et al., 2006) by hypothesising the same proportion of boundaries found in the training set. Our results also compare favourably with some of the best results reported in the meeting segmentation literature to date, namely $P_k = 0.32$ and $WD = 0.36$, for a lexical cohesion algorithm on the ICSI corpus (Galley et al., 2003), and $P_k = 0.34$ and $WD = 0.36$, for a maximum entropy approach combining lexical, conversational and video features on the AMI corpus (Hsueh et al., 2006).

Although these results are promising, they pose a question as regards data representation. While V_{pd} yielded the best results under MAP, V_{sp} worked best overall under a proportional threshold. What is the effect of encoding more detailed pause and overlap information? Unfortunately, the MDTM corpus has not been annotated to the level of detail required to allow in-depth investigation of this question. We therefore turn to the far larger and more detailed AMI corpus for our next experiments. In addition to helping clarify the representation issue, testing our method on this corpus will give us a better idea of how our method performs in a more standard topic segmentation task.

5 AMI Segmentation

The AMI corpus is a collection of meetings recorded under controlled conditions, many of which have a fixed scenario, goals and assigned participant roles. The corpus is manually transcribed, and annotated with word-level timings and a variety of metadata, including topics and sub-topics (Carletta, 2007). Transcriptions in the AMI corpus are extracted from redundant recording channels (lapel, headset and array microphones), and stored separately for each participant. Because timing information in AMI is so detailed, we were able to create much richer VH representations, including finer grained pause and overlap information.

The original XML-encoded AMI data were parsed and collated to produce our variants of the VH scheme. We tested four types of VH: V_v, which includes only vocalisation events; V_g, which includes only pause and speech overlap events; V_a, which includes all vocalisations, pauses and overlaps; and V_r, which is similar to V_{pd} in that it includes speaker roles in addition to vocalisations. Pauses and overlaps were encoded by the same variable g_i, where $g_i > 0$ indicates a pause $g_i < 0$ an overlap, as shown in Figure 1. Unlike MDTM, no arbitrary threshold was imposed on the identification of pause and overlap events. As before, we tested on a horizon $n = 3$, in order to allow comparison with MDTM results.

The training and boundary inference process also remained as in the MDTM experiment, except that the larger amount of meeting data available enabled us to increase the number of folds for cross validation so that the results could be tested for statistical significance.

The error scores and the number of boundaries predicted for the different representational vari-

ants, filtering an thresholding strategies are shown in Table 3. Although all methods significantly outperformed the baseline scores of $P_k = 0.473$ and *WD* $= 0.542$ (paired t-tests, $p < 0.01$, for all conditions), there were hardly any differences in P_k scores across the different representations, even when conservative boundary filtering is performed. Filtering, however, caused a significant improvement for *WD* in all cases, though the combined effects of proportional thresholding and filtering caused the classifier to err on the side of underprediction. A 3-way analysis of variance including non-filtered scores for proportional threshold resulted in $F[4, 235] = 31.82$, $p < 0.01$ for *WD* scores. These outcomes agree with the results of the smaller-scale MDTM segmentation experiment, showing that categorisation based on conversational features tend to mark clusters of segments around the true topic boundary. In addition, the trend for better performance of proportional thresholding exhibited in the MDTM data was not as clearly observed in the AMI data, where only *WD* scores were significantly better than MAP ($p < 0.01$, Tukey HSD).

Table 3: Segmentation results for 16-fold cross validation on AMI corpus, horizon $\boldsymbol{n = 3}$. Correct number of boundaries in reference is 724.

Threshold	Filter	Data	P_k	*WD*	# bound.
MAP	no	V_a	0.270	0.462	3322
		V_g	0.278	0.433	1875
		V_v	0.273	0.449	3075
		V_r	0.271	0.448	3073
	yes	V_a	0.272	0.362	574
		V_g	0.277	0.391	851
		V_v	0.275	0.358	468
		V_r	0.274	0.357	469
Proport.	no	V_a	0.289	0.398	1233
		V_g	0.290	0.382	735
		V_v	0.293	0.387	1002
		V_r	0.293	0.387	1002
	yes	V_a	0.293	0.353	241
		V_g	0.290	0.362	383
		V_v	0.297	0.350	183
		V_r	0.297	0.350	182

It is noteworthy that the finer-grained representations from which speaker roles were excluded (V_v, V_g, and V_a) yielded segmentation performance comparable to the MDTM segmentation performance under V_{sp} and V_{pd}. In fact, adding speaker role information in V_r did not result in improvement for AMI segmentation. Also interesting is the fact that representations based solely on pause and overlap information also produced good performance, thus confirming our initial intuition.

5.1 MDTM revisited

Since V_v, V_g and V_a seem to perform well without including speaker role information (except for the current vocalisation's speaker role) we would like to see how a similar representation might affect segmentation performance for MDTM. We therefore tested whether excluding preceding and following speaker role information from V_{sp} and V_{pd} had a positive impact on PCD segmentation performance. However, contrary to our expectations neither of the modified representations yielded better scores. The best results, achieved for the modified V_{pd} under proportional thresholding ($P_K = 0.27$ and *WD* $= 0.34$), failed to match the results obtained with the original representation. It seems that the various and more specialised speaker roles found in medical meetings can be good predictors of PCD boundaries. For example: a typical pattern at the start of a PCD is the recounting of the patient's initial symptoms and clinical findings by the registrar in a narrative style. In AMI, on other hand, the roles are much fewer, being only acted out by the participants as part of the given scenario, which might explain the irrelevance of these roles for segmentation.

5.2 Conclusion

MDTM segmentation differs from topic segmentation of the AMI meetings in that PCDs are more regular in their occurrence than meeting topics proper. Speaker role information was also found to help MDTM segmentation, which was expected since there are many more very distinct active speaker roles in MDTM (10 specialties, in total). Furthermore, V_{sp} and V_{pd} represent pauses and overlaps as reserved roles, so that the information encoded in V_g and V_a as separate variables appear in the speaker role horizon of V_{sp} and V_{pd}. It is possible that the finer-grained timing annotation of the AMI corpus (including detailed overlap and pause information unavailable in the MDTM data) contributed to the relatively good segmentation performance achieved on AMI even in the absence of speaker role cues. It would be interesting to investigate whether finer pause and overlap timings can also improve MDTM segmentation. This suggests some requirements for MDTM data collection and pre-processing, such as the use of individual close-talking and the use of a speech recogniser to derive word-level timings. We plan on conducting further experiments in that regard.

Acknowledgements

This research was funded by Science Foundation Ireland under the Research Frontiers program. The presentation was funded by Grant 07/CE/1142, Centre for Next Generation Localisation (CNGL).

References

J. Ajmera and C. Wooters. 2003. A robust speaker clustering algorithm. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU'03*, pages 411–416. IEEE Press.

R. F. Bales. 1950. *Interaction Process Analysis: A Method for the Study of Small Groups*. Addison-Wesley, Cambridge, Mass.

S. Banerjee, C. Rose, and A. I. Rudnicky. 2005. The necessity of a meeting recording and playback system, and the benefit of topic-level annotations to meeting browsing. In *Proceedings of the 10th International Conference on Human-Computer Interaction (INTERACT'05)*, pages 643–656.

D. Beeferman, A. Berger, and J. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34:177–210, Feb. 10.1023/A:1007506220214.

J. Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41(2):181–190.

J. M. J. Dabbs and B. Ruback. 1987. Dimensions of group process: Amount and structure of vocal interaction. *Advances in Experimental Social Psychology*, 20(123–169).

M. Galley, K. R. McKeown, E. Fosler-Lussier, and H. Jing. 2003. Discourse segmentation of multiparty conversation. In *Proceedings of the 41st Annual Meeting of the ACL*, pages 562–569.

A. Gruenstein, J. Niekrasz, and M. Purver. 2005. Meeting structure annotation: Data and tools. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 117–127, Lisbon, Portugal, September.

M. A. Hearst. 1997. Texttiling: segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64.

J. Holmes. 1984. Modifying illocutionary force. *Journal of Pragmatics*, 8(3):345 – 365.

P. Hsueh and J. D. Moore. 2007. Combining multiple knowledge sources for dialogue segmentation in multimedia archives. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. ACL Press.

P. Hsueh, J. D. Moore, and S. Renals. 2006. Automatic segmentation of multiparty dialogue. In *Proceedings of the 11th Conference of the European Chapter of the ACL (EACL)*, pages 273–277. ACL Press.

J. Jaffe and S. Feldstein. 1970. *Rhythms of dialogue*. Academic Press, New York.

G. H. John and P. Langley. 1995. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI'95)*, pages 338–345, San Francisco, CA, USA, August. Morgan Kaufmann Publishers.

B. Kane and S. Luz. 2006. Multidisciplinary medical team meetings: An analysis of collaborative working with special attention to timing and teleconferencing. *Computer Supported Cooperative Work (CSCW)*, 15(5):501–535.

S. Luz. 2009. Locating case discussion segments in recorded medical team meetings. In *Proceedings of the ACM Multimedia Workshop on Searching Spontaneous Conversational Speech (SSCS'09)*, pages 21–30, Beijing, China, October. ACM Press.

I. Malioutov, A. Park, R. Barzilay, and J. Glass. 2007. Making sense of sound: Unsupervised topic segmentation over acoustic input. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 504–511, Prague, Czech Republic, June. Association for Computational Linguistics.

N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke. 2001. The meeting project at ICSI. In *Procs. of Human Language Technologies Conference*, San Diego.

M. Oliveira, 2002. *The role of pause occurrence and pause duration in the signaling of narrative structure*, volume 2389 of *LNAI*, pages 43–51. Springer.

L. Pevzner and M. A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28:19–36, Mar.

S. Renals, T. Hain, and H. Bourlard. 2007. Recognition and interpretation of meetings: The AMI and AMIDA projects. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '07)*.

M. Sherman and Y. Liu. 2008. Using hidden Markov models for topic segmentation of meeting transcripts. In *Proceedings of the IEEE Spoken Language Technology Workshop*, pages 185–188.

E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür. 2000. Prosody-based automatic segmentation of speech into sentences and topics. *Speech communication*, 32(1-2):127–154.

L. ten Bosch, N. Oostdijk, and L. Boves. 2005. On temporal aspects of turn taking in conversational dialogues. *Speech Communication*, 47:80–86.

Author Index