SIGDIAL 2009

Proceedings of the SIGDIAL 2009 Conference

The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue

Edited by
Patrick Healey, Roberto Pieraccini,
Donna Byron, Steve Young, Matthew Purver

11–12 September 2009
London, UK

Queen Mary
University of London

Production and Manufacturing by
Tribun EU s.r.o.
Gorkého 41
602 00 Brno
Czech Republic

We thank our sponsors:

Microsoft Research

AMIDS

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 978-1-932432-64-0

Introduction

We are happy to present the Proceedings of the SIGDIAL 2009 Conference, the 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue. This year the SIGDIAL meeting has been elevated from Workshop to Conference by ACL, its main sponsoring organization. That is an unmistakable recognition of the role that dialogue and discourse research play in the fields of computational linguistics, human-machine communication, and language technology in general.

Our thanks go to the program committee who have performed an excellent job in reviewing the submitted papers and providing scores and comments that have helped to maintain a high standard of quality. We received a record number of 103 submissions; 24 of them were accepted as lectures, 24 as poster presentations, and 3 as demos. As it is a tradition of SIGDIAL, we have attempted to preserve a balance among the different topics, especially between the more theoretical and empirically oriented studies and the technological and engineering challenges.

We are also grateful to our two keynote speakers: Professor Yorick Wilks and Professor Janet Bavelas for providing stimulating talks on the state-of-the-art in dialogue research.

Many thanks go to Donna Byron and Steve Young who graciously agreed to serve as technical program chairs and coordinated the whole review process by selecting the program committee, assigning papers to reviewers, resolving conflicts, and making sure that all reviews were submitted on time. They also selected the candidates for the best paper awards, and created the final program of the conference. Many thanks to Matthew Purver, local Chair, who has done an outstanding job; always available and ready to help at any step of the process, from the local arrangements, to the conference Web site, to the publication of these proceedings.

We are grateful to ACL and Priscilla Rasmussen for providing financial support and budgetary information, and the SIGDIAL board for their constant support in all matters, in particular Tim Paek, SIGDIAL President, Amanda Stent, David Traum, and Kristiina Jokinen.

And finally, thanks to all the authors that submitted the papers and all the participants to the meeting for their continuous support of this exciting and interesting conference.

Pat Healey & Roberto Pieraccini

SIGDIAL 2009 Co-Chairs

Conference Organization

General Co-Chairs:

Patrick G. T. Healey, Queen Mary University of London, UK
Roberto Pieraccini, SpeechCycle, USA

Technical Program Co-Chairs:

Donna Byron, Northeastern University, USA
Steve Young, University of Cambridge, UK

Local Chair:

Matthew Purver, Queen Mary University of London, UK

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:

Gregory Aist, Arizona State University, USA
Jan Alexandersson, DFKI GmbH, Germany
Srinivas Bangalore, AT&T Labs - Research, USA
Dan Bohus, Microsoft Research, USA
Johan Bos, Università di Roma "La Sapienza", Italy
Charles Calloway, University of Edinburgh, UK
Rolf Carlson, Royal Institute of Technology (KTH), 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
Stephanie Elzer, Millersville University, USA
Mary Ellen Foster, Technical University Munich, Germany
Kallirroi Georgila, University of Southern California, USA
Jonathan Ginzburg, King's College London, UK
Genevieve Gorrell, Sheffield University, UK
Alexander Gruenstein, Massachussetts Institute of Technology, USA
Pat Healey, Queen Mary University of London, UK
Mattias Heldner, Royal Institute of Technology (KTH), Sweden
Beth Ann Hockey, University of California at Santa Cruz, USA
Kristiina Jokinen, University of Helsinki, Finland
Arne Jonsson, University of Linköping, Sweden
Simon Keizer, University of Cambridge, UK
John Kelleher, Dublin Institute of Technology, Ireland
Alexander Koller, University of Edinburgh, UK
Ivana Kruijff-Korbayová, Universität des Saarlandes, Germany
Staffan Larsson, Göteborg University, Sweden

Gary Geunbae Lee, Pohang University of Science and Technology, Korea
Fabrice Lefèvre, University of Avignon, France
Oliver Lemon, University of Edinburgh, 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 University Berlin, Germany
Vincent Ng, University of Texas at Dallas, USA
Tim Paek, Microsoft Research, USA
Patrick Paroubek, LIMSI-CNRS, France
Roberto Pieraccini, SpeechCycle, USA
Paul Piwek, Open University, UK
Rashmi Prasad, University of Pennsylvania, USA
Matthew Purver, Queen Mary University of London, UK
Alex Rudnicky, Carnegie Mellon University, USA
Yoshinori Sagisaka, Waseda University, Japan
Ruhi Sarikaya, IBM Research, USA
Candy Sidner, BAE Systems AIT, USA
Ronnie Smith, East Carolina University, USA
Amanda Stent, AT&T Labs - Research, USA
Matthew Stone, Rutgers University, USA
Matthew Stuttle, Toshiba Research, UK
Joel Tetreault, Educational Testing Service, USA
Michael White, Ohio State University, USA
Jason Williams, AT&T Labs - Research, USA

Invited Speakers:

Janet Bavelas, University of Victoria, Canada
Yorick Wilks, University of Sheffield, UK

Table of Contents

Conference Program

Friday, 11th September, 2009

8:45–9:00 Introduction

9:00–10:00 Papers 1-3:

Evaluating the Effectiveness of Information Presentation in a Full End-To-End Dialogue System
Taghi Paksima, Kallirroi Georgila and Johanna Moore

Can I Finish? Learning When to Respond to Incremental Interpretation Results in Interactive Dialogue
David DeVault, Kenji Sagae and David Traum

Are You Being Addressed? - Real-Time Addressee Detection to Support Remote Participants in Hybrid Meetings
Harm op den Akker and Rieks op den Akker

10:00–11:00 Invited Talk 1:

What's Unique About Dialogue?
Janet Bavelas

11:00–11:15 Coffee

11:15–12:15 Papers 4-6:

Incremental Reference Resolution: The Task, Metrics for Evaluation, and a Bayesian Filtering Model that is Sensitive to Disfluencies
David Schlangen, Timo Baumann and Michaela Atterer

Dealing with Interpretation Errors in Tutorial Dialogue
Myroslava Dzikovska, Charles Callaway, Elaine Farrow, Johanna Moore, Natalie Steinhauser and Gwendolyn Campbell

Towards the Interpretation of Utterance Sequences in a Dialogue System
Ingrid Zukerman, Patrick Ye, Kapil Kumar Gupta and Enes Makalic

12:15–1:30 Lunch

Friday, 11th September, 2009 (continued)

1:30–2:30 Papers 7-9:

Participant Subjectivity and Involvement as a Basis for Discourse Segmentation
John Niekrasz and Johanna Moore

Genre-Based Paragraph Classification for Sentiment Analysis
Maite Taboada, Julian Brooke and Manfred Stede

Detecting the Noteworthiness of Utterances in Human Meetings
Satanjeev Banerjee and Alexander Rudnicky

2:30–4:00 Poster Session 1:

A: An Experimental Investigation into... B: ...Split Utterances
Christine Howes, Patrick Healey and Gregory Mills

Interactive Gesture in Dialogue: a PTT Model
Hannes Rieser and Massimo Poesio

Tense, Temporal Expressions and Demonstrative Licensing in Natural Discourse.
Iker Zulaica-Hernández and Javier Gutiérrez-Rexach

Prosodic Turn-Yielding Cues With and Without Optical Feedback
Caroline Clemens and Christoph Diekhaus

Exploring Miscommunication and Collaborative Behaviour in Human-Robot Interaction
Theodora Koulouri and Stanislao Lauria

A Two-Tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies
Srinivasan Janarthanam and Oliver Lemon

Analysis of Listening-Oriented Dialogue for Building Listening Agents
Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, Yasuhiro Minami and Hideki Isozaki

On NoMatchs, NoInputs and BargeIns: Do Non-Acoustic Features Support Anger Detection?
Alexander Schmitt, Tobias Heinroth and Jackson Liscombe

Estimating Probability of Correctness for ASR N-Best Lists
Jason Williams and Suhrid Balakrishnan

Friday, 11th September, 2009 (continued)

Not a Simple Yes or No: Uncertainty in Indirect Answers
Marie-Catherine de Marneffe, Scott Grimm and Christopher Potts

Concept Form Adaptation in Human-Computer Dialog
Svetlana Stoyanchev and Amanda Stent

Demos:

Automatic Generation of Information State Update Dialogue Systems that Dynamically Create Voice XML, as Demonstrated on the iPhone
Helen Hastie, Xingkun Liu and Oliver Lemon

Dialog System for Mixed Initiative One-Turn Address Entry and Error Recovery
Rajesh Balchandran, Leonid Rachevsky, Larry Sansone and Roberto Sicconi

Leveraging POMDPs Trained with User Simulations and Rule-based Dialogue Management in a Spoken Dialogue System
Sebastian Varges, Silvia Quarteroni, Giuseppe Riccardi, Alexei Ivanov and Pierluigi Roberti

4:00–4:15 Coffee

4:15–5:15 Papers 10-12:

Speeding Up the Design of Dialogue Applications by Using Database Contents and Structure Information
Luis Fernando D'Haro, Ricardo de Cordoba, Juan Manuel Lucas, Roberto Barra-Chicote and Ruben San-Segundo

Modeling User Satisfaction with Hidden Markov Models
Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar and Sebastian Möller

Discourse Structure and Performance Analysis: Beyond the Correlation
Mihai Rotaru and Diane Litman

7:00–11:00 Conference Dinner

Saturday, 12th September, 2009

9:00–10:00 Papers 13-15:

The Role of Interactivity in Human-Machine Conversation for Automatic Word Acquisition
Shaolin Qu and Joyce Chai

Clarification Potential of Instructions
Luciana Benotti

What do We Know about Conversation Participants: Experiments on Conversation Entailment
Chen Zhang and Joyce Chai

10:00–11:00 Invited Talk 2:

Artificial Companions as Dialogue Agents
Yorick Wilks

11:00–11:15 Coffee

11:15–12:15 Papers 16-18:

Effects of Conversational Agents on Human Communication in Thought-Evoking Multi-Party Dialogues
Kohji Dohsaka, Ryota Asai, Ryuichiro Higashinaka, Yasuhiro Minami and Eisaku Maeda

Models for Multiparty Engagement in Open-World Dialog
Dan Bohus and Eric Horvitz

Extracting Decisions from Multi-Party Dialogue Using Directed Graphical Models and Semantic Similarity
Trung Bui, Matthew Frampton, John Dowding and Stanley Peters

12:15–1:30 Lunch & SIGDIAL Business Meeting

Saturday, 12th September, 2009 (continued)

1:30–2:30 Papers 19-21:

Learning to Predict Engagement with a Spoken Dialog System in Open-World Settings
Dan Bohus and Eric Horvitz

Turn-Yielding Cues in Task-Oriented Dialogue
Agustín Gravano and Julia Hirschberg

Split Utterances in Dialogue: a Corpus Study
Matthew Purver, Christine Howes, Eleni Gregoromichelaki and Patrick Healey

2:30–3:45 Poster Session 2:

k-Nearest Neighbor Monte-Carlo Control Algorithm for POMDP-Based Dialogue Systems
Fabrice Lefèvre, Milica Gašić, Filip Jurčíček, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu and Steve Young

Comparison of Classification and Ranking Approaches to Pronominal Anaphora Resolution in Czech
Giang Linh Nguy, Václav Novák and Zdeněk Žabokrtský

Spoken Tutorial Dialogue and the Feeling of Another's Knowing
Diane Litman and Kate Forbes-Riley

Evaluating Automatic Extraction of Rules for Sentence Plan Construction
Amanda Stent and Martin Molina

Eliciting Interactional Phenomena in Human-Human Dialogues
Joakim Gustafson and Miray Merkes

TELIDA: A Package for Manipulation and Visualization of Timed Linguistic Data
Titus von der Malsburg, Timo Baumann and David Schlangen

Cascaded Lexicalised Classifiers for Second-Person Reference Resolution
Matthew Purver, Raquel Fernández, Matthew Frampton and Stanley Peters

Attention and Interaction Control in a Human-Human-Computer Dialogue Setting
Gabriel Skantze and Joakim Gustafson

Ranking Help Message Candidates Based on Robust Grammar Verification Results and Utterance History in Spoken Dialogue Systems
Kazunori Komatani, Satoshi Ikeda, Yuichiro Fukubayashi, Tetsuya Ogata and Hiroshi Okuno

Dialogue Behaviour under High Cognitive Load
Jessica Villing

Saturday, 12th September, 2009 (continued)

A Comparison between Dialog Corpora Acquired with Real and Simulated Users
David Griol, Zoraida Callejas and Ramón López-Cózar

Simultaneous Dialogue Act Segmentation and Labelling using Lexical and Syntactic Features
Ramon Granell, Stephen Pulman and Carlos-D. Martínez-Hinarejos

The Spoken Dialogue Challenge
Alan Black and Maxine Eskenazi

3:45–4:15 Coffee & Grand Challenge Discussion

4:15–5:15 Papers 22-24:

Unsupervised Classification of Dialogue Acts using a Dirichlet Process Mixture Model
Nigel Crook, Ramon Granell and Stephen Pulman

A Handsome Set of Metrics to Measure Utterance Classification Performance in Spoken Dialog Systems
David Suendermann, Jackson Liscombe, Krishna Dayanidhi and Roberto Pieraccini

Contrasting the Interaction Structure of an Email and a Telephone Corpus: A Machine Learning Approach to Annotation of Dialogue Function Units
Jun Hu, Rebecca Passonneau and Owen Rambow

5:15–5:35 Close

Evaluating the Effectiveness of Information Presentation in a Full End-To-End Dialogue System

Taghi Paksima
Enterprise Search Group
Microsoft
D-81669 Munich, Germany
taghi.paksima@microsoft.com

Kallirroi Georgila
Institute for Creative Technologies
University of Southern California
Marina del Rey, CA 90292, USA
kgeorgila@ict.usc.edu

Johanna D. Moore
School of Informatics
University of Edinburgh
Edinburgh, EH8 9AB, UK
j.moore@ed.ac.uk

Abstract

Recent work on information presentation in dialogue systems combines user modelling (UM) and stepwise refinement through clustering and summarisation (SR) in the UMSR approach. An evaluation in which participants rated dialogue transcripts showed that UMSR presents complex trade-offs understandably, provides users with a good overview of their options, and increases users' confidence that all relevant options have been presented (Demberg and Moore, 2006). In this paper, we evaluate the effectiveness of the UMSR approach in a more realistic setting, by incorporating this information presentation technique into a full end-to-end dialogue system in the city information domain, and comparing it with the traditional approach of presenting information sequentially. Our results suggest that despite complications associated with a real dialogue system setting, the UMSR model retains its advantages.

1 Introduction

Spoken dialogue systems (SDS) that help users find a desired option (e.g., flight, restaurant, movie) from the set of options satisfying their constraints typically present options sequentially, ordered along a default dimension (e.g., by price or departure time). An example is shown in Fig. 1.

The user can then navigate through the options and refine them by offering new constraints until a suitable option has been found. However, when the number of available options is large, this process can be painstaking, leading to long dialogues and reduced user satisfaction. Thus a major challenge in the development of SDS is to improve information presentation algorithms. This is important for several reasons: (1) to avoid overburdening the user's memory by presenting too many options; (2) to ensure that the user is given an overview of the available option space so that the optimal option can be found; and (3) to minimise the number of dialogue turns (hence dialogue duration) required for the user to find an acceptable option. As Walker et al. (2001) showed, failing to meet this third goal may reduce overall user satisfaction.

> There are six restaurant options matching your query.
>
> Number 1: Voujon offers a bright, airy and contemporary dining area, with simple floral displays and leather seating. It serves Indian cuisine. It is located in the city centre. The average price is £24 per person.
>
> Number 2: Saffrani's decor is modern, the dining room wee, though the menu is enormous, and the atmosphere charming. It offers new Indian dishes never before seen in Edinburgh. It serves Indian, seafood cuisine. It is located in the city centre. The average price is £28 per person.
>
> Number 3: Britannia Spice ...

Figure 1: Example of sequential information presentation in the city information domain (modified version of the TownInfo system (Lemon et al., 2006)).

Recently several approaches have been proposed to overcome the shortcomings of the sequential enumeration strategy (Polifroni et al., 2003; Chung, 2004; Demberg and Moore, 2006; Polifroni and Walker, 2008). Because of the complexity of building a complete end-to-end SDS, these approaches have been evaluated using an "overhearer" methodology in which dialogues are either hand-crafted or simulated and then presented to subjects, either as textual transcripts

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 1–10,
Queen Mary University of London, September 2009.

(Demberg and Moore, 2006; Polifroni and Walker, 2008) or audio recordings (Walker et al., 2004), for rating. The general consensus from these studies is that users significantly prefer approaches that take their preferences into account. However, because users were not interacting with these SDS, the evaluation criteria were limited to users' perceptions (e.g., informativeness, good overview of options, confidence in choice, etc.), and metrics such as effectiveness (i.e., actual or perceived task completion) and efficiency (i.e., length of dialogue) could not be assessed. To address this issue, Winterboer and Moore (2007) carried out a Wizard-of-Oz (WOz) study in which users participated in dialogues controlled by two different information presentation algorithms. They found that not only did users prefer presentations based on a user model, dialogues employing the "user-model based summarise and refine" (UMSR) approach led to greater task success and dialogue efficiency.

In this paper, we take this one step further, and evaluate the effectiveness of the UMSR approach in a more realistic setting, incorporating this content selection and presentation strategy into a full end-to-end dialogue system, and comparing it to the traditional sequential enumeration approach. Our results suggest that despite complications associated with a real dialogue system setting, the UMSR model retains its advantages. Our results also verify the hypothesis that the UMSR model presents complex trade-offs in a concise, yet understandable way. Furthermore, as in the WOz study, the UMSR approach leads to a significant reduction in the number of dialogue turns.

The structure of the paper is as follows: In Sec. 2, we discuss related work. In Sec. 3 we present the full end-to-end SDS used for comparison between the standard sequential enumeration approach and the UMSR approach. In Sec. 4 we describe how we implemented the UMSR approach. Then in Sec. 5 we provide an example. In Sec. 6 we describe our experimental design and in Sec. 7 our results. Finally in Sec. 8, we present our conclusions.

2 Previous Approaches

As noted above, a number of approaches to information presentation in SDS have recently been proposed. The user-model based (UM) approach employs a model of the users preferences and decision theory techniques to identify and present a small number of options that best match the user's preferences (Carenini and Moore, 2001; Walker et al., 2004; Moore et al., 2004). Fig. 2 shows a sample presentation generated using the UM approach for a student user who cares most about price and flying direct.

> There's a direct flight on BMI with a good price. It arrives at four ten p.m. and costs a hundred and twelve pounds. The cheapest flight is on Ryanair. It arrives at twelve forty-five p.m. and costs just fifty pounds, but it requires a connection in Dublin.

Figure 2: Example of information presentation based on the UM approach, from (Moore et al., 2004).

Although the UM approach enables a concise presentation of a small number of options, highlighting the ways in which these options satisfy user preferences, it does not scale up to presenting a large number of options. When there are hundreds of potentially relevant options to consider (e.g., all Italian restaurants in New York City), there may be a large number of options that fit the user model, making it difficult to narrow down the number of options. In addition, users may not be able to provide constraints until they are presented with more details concerning the option space. Moreover, the system does not provide an overview of the available options, which may lead to the user's actual or perceived missing out on possibly better alternatives.

An alternative approach, which we call the summarise and refine approach (SR) (Polifroni et al., 2003; Chung, 2004) structures large datasets for summarisation and successive refinement. It supports the user in narrowing in on a suitable option by grouping the options in the database into clusters of options with similar features. The system can then summarise the common attributes of the options and thus suggest further refinement constraints to the user. In this content structuring approach, the prompts presented to the user and the order in which they appear, are determined at runtime based on an algorithm that computes the most useful set of attributes, as dictated by the set of options that satisfy the user query. For large datasets, attributes that partition the data into the minimal number of clusters are chosen, so that a concise summary can be presented to the user to refine (Fig. 3).

In addition to its many advantages, the SR approach also has several limitations. First, depending on the size and complexity of the domain, many turns may be required during the refinement

M: What restaurants do you have?
S: I have found 983 restaurants. Most of them are located in Boston and Cambridge. There are 32 choices for cuisine. I also have information about price range.
M: Okay tell me about the ones in Boston.
S: I have found 401 restaurants in Boston. There are 29 choices for cuisine.
M: Do you have any that serve seafood?
S: I have found 19 seafood restaurants. They are predominantly in Back Bay, the North End, South Boston and the South End.

Figure 3: Sample dialogue between simulator (M) and SR system (S), from (Polifroni et al., 2003).

process. Because the user's preferences are not taken into account, the clusters may contain many irrelevant entities, which must be filtered out successively with each refinement step. Second, if there is no optimal solution, exploration of trade-offs between options can be difficult. Finally, the chosen clusters may be based on attributes that are irrelevant for the specific user.

In an attempt to combine the benefits of the UM and SR approaches, Demberg & Moore (2006) devised the user-model based summarise and refine (UMSR) approach to information presentation. This approach first clusters the values of each attribute in order to group them so that the options can be summarised more easily later, and labels like "cheap", "moderate", "expensive" can be assigned to values of continuous categories such as "price". The system then structures options into an *option tree* based on the ranking of attributes in the user model, the options returned from the database, and the attribute-value clustering. The resulting option tree determines how different options relate to one another, and which ones are most attractive for the user. After the tree structure is built, it is pruned to decide which options are compelling to the user according to the user model. This allows the system to save time by omitting options that are not of any potential interest to the user. Once pruning is complete, each branch of the tree describes a possible refinement path, and thus can be used to direct dialogue flow. Trade-offs between alternative options are presented explicitly in order to provide the user with a better overview of the option space. In addition, to give users confidence that they are being presented with all of the relevant options, a brief account of all the remaining (irrelevant) options is also provided. For a more detailed discussion of the UMSR approach, see (Demberg and Moore, 2006). In Sec. 4 we describe how we employed the UMSR approach in our system.

3 The TownInfo System

The TownInfo SDS was developed as part of the EC project TALK (Lemon et al., 2006). Users can search for hotels, bars and restaurants in an artificial town. The system supports two dialogue strategies, one hand-crafted and another learnt using Reinforcement Learning (Henderson et al., 2008). For the current experiment we used the hand-crafted strategy. Natural language understanding is performed using a keyword-based parser and natural language generation is based on templates. The information presentation is sequential. An example is given in Fig. 1, taken from the modified version of TownInfo for the current experiment. Although the original TownInfo system supported speech input and speech output, here we use text input/output to make sure that our results are not influenced by poor recognition accuracy or intelligibility due to poor speech synthesis. Of course, as we mention in Sec. 8, the next step would be to perform an experiment with speech input/output.

For our current experiment we focussed on restaurant recommendations and the TownInfo database had to be extended to include a much wider range of options to provide more realistic information presentation scenarios. The database used in our experiments contains a total of 80 restaurants in Edinburgh, UK.

4 The UMSR Algorithm

This section briefly describes our implementation of the UMSR algorithm; for more details see (Demberg and Moore, 2006). Sec. 5 provides an example for clarity.

4.1 The User Model

The user model contains the user's ranking and preferred values for the relevant attributes in the restaurant domain: `price`, `distance`, `star rating`, `service rating`, and `cuisine type`. Table 1 shows a sample user model. The *Rank* field indicates the relative importance of the attributes for the user, with 1 being most important. The *Value* field indicates the user's preferred value for each attribute.[1]

[1] If two attributes in a user model have identical ranks, the order of the preferences is used to decide which has a higher priority.

UserID	Attribute	Value	Rank
1	Price	Cheap	1.00
1	Distance	Near	2.00
1	Star	High	3.00
1	Cuisine	Indian	4.00
1	Service	Don't Care	5.00

Table 1: Sample user model for a student.

According to Elzer et al. (1994), some preferences are enough to reject options outright (and therefore are more like goals) whereas others are more purely like preferences (to be weighed and ranked). Here we do not make such a distinction.

4.2 Adapting to Changes to the User Model

In the original design, the user model was created at the outset and not modified during the dialogue. However, during initial piloting of the system, we found that this design did not support "situational preferences". For example, consider the user model for the student in Table 1. This user normally prefers to have Indian food if she has the option to (a "dispositional preference"). If, however, in the current situation she is entertaining a friend from out of town who wishes to try Scottish food, the user may decide to explore options for Scottish cuisine (a "situational preference"). Here, the user changes her original query for the situation, thus redefining her preferences. When this occurs, we must perform a new database query and rebuild the option tree. To take these dynamic changes into account during the course of the dialogue, at each dialogue turn the user query is compared against the user model, and if any difference is noted, the user model is updated to reflect the current preferences, the tree is rebuilt using the new user model, and the dialogue continues with a summary of the available options based on this new tree.

Note that for individual models, i.e. user models that are designed for individual people and not for classes of users (student or business person), some queries could justify situational changes and some could indicate permanent (or at least less temporary) changes to the user model (e.g., "Are there any nicer restaurants? I got a new job"). In our experiment we use only class models and we do not allow permanent changes to the user model.

4.3 The Clustering Algorithm

Following (Polifroni et al., 2003) and (Demberg and Moore, 2006), we used agglomerative group-average clustering to automatically group values for each attribute. The algorithm begins by assigning each unique attribute value to its own bin, and successively merging bins whose means are most similar until a stopping criterion (a target of no more than three clusters, in our implementation) is met. The bins are then assigned predefined labels, e.g., "cheap", "moderately priced" and "expensive" for `price`. Clustering attribute values with this algorithm allows for database-dependent labelling. Therefore, a restaurant with a price of £35 might be considered as expensive for Edinburgh, but inexpensive for London.

4.4 Building the Option Tree

The tree building algorithm is recursive. It begins at the root node, which contains all entities in the retrieved dataset, and builds up the tree level by level based on the ranking of attributes in the user model. At each node of the tree, it retrieves the next attribute preference from the user model and then invokes the clustering algorithm for this attribute's values. Once the current dataset has been clustered, the algorithm then adds the resultant clusters as the children of the current node. After each cluster is added, the algorithm is invoked recursively on the newly created children of the current node.

As the tree is being constructed, the algorithm arranges the nodes in the tree such that the children of each node are ordered from left to right in decreasing order of desirability. For example, if the particular user prefers restaurants that are far from the city centre, the clusters based on `distance` would be ordered such that "far" is the leftmost child and "near" is the rightmost child. Fig. 5 depicts an option tree structure for the user model of Table 1, in the context of the example of Sec. 5. The numbers in the nodes indicate how many options are represented by the node.

Given an option tree ordered in this way, to find the best available options, the system traverses the tree in a depth-first fashion starting from the root and selecting the leftmost branch at each node.

4.5 Pruning the Option Tree

The goal of the UMSR algorithm is to present an overview of the available options, that are most relevant to the user's preferences, concisely and understandably. To determine the relevance of options, we use the notion of "dominance" defined in Demberg & Moore (2006). *Dominant* options are those for which there is no other option in the dataset that is better on all attributes. A *domi-*

nated option is in all respects equal to or worse than some other option in the relevant subset of the database; it should not be of interest for any rational user.

The pruning algorithm follows Demberg & Moore (2006), and thus we summarise it only briefly here. The algorithm operates directly on the ordered option tree, using the tree structure so that it can efficiently determine dominance relations without having to compare each pair of options. The algorithm traverses the tree in depth-first order, generating constraints during this process. These constraints encode the properties that other options would need to satisfy in order not to be dominated by the options which have already been deemed to be dominant. A node must fulfil the constraints that apply to it, otherwise it is pruned from the tree. If an option (or a cluster of options) satisfies a constraint, the property that satisfied the constraint is marked as the options' *justification*. If some, but not all, of the constraints can be satisfied by an option, the constraints are propagated to the other nodes (see Fig. 5).

4.6 Natural Language Generation

Once a pruned option tree has been constructed, the system can generate a presentation to the user. The natural language generation (NLG) algorithm includes three steps described below.

4.6.1 Identifying Trade-offs

To identify the trade-offs, the algorithm traverses the tree looking for constraints that were generated during the pruning process. For each node that generated a constraint, the algorithm finds the best sibling, which satisfies the constraint. It does this by first checking the siblings of the current node, and if none satisfy the constraint, it moves up the tree and recursively traverses siblings of the ancestor node. Once a trade-off node is found, it is recorded in the option tree at that point, and the algorithm then searches upward in the tree to find the lowest common parent of the trade-off nodes. This is the "Differentiating Parent" for the trade-off; the attribute that determines the children of this node indicates the trade-off the user has to make. In the example shown in Fig. 5, the attribute that the user must trade-off is `distance`. The system then generates a natural language description of the trade-off using templates, e.g., "For a restaurant that serves Indian cuisine you would have to go farther from the city centre". If a constraint cannot be satisfied, the system generates a sentence describing this situation, e.g., "However, there are no inexpensive restaurants that match your preferences".

4.6.2 Summarisation

The summarisation process aims to present the user with an understandable and concise overview of the available options, so that she can make informed decisions on how to proceed with her query. In a SDS, it is important not to present too much information in a single turn in order to keep the memory load on the user manageable. Thus, in our implementation, the system provides a summary of the first three attributes (corresponding to the first three levels of the tree), e.g., "I found 12 restaurants which are inexpensive. Only 7 of them are near the city centre, out of which only 4 have a high star rating". If there are any trade-offs over the first three attributes, they are also mentioned to the user, using the algorithm described above.

4.6.3 Turn Generation

The turn generation algorithm first determines whether the current user's query is consistent with the preferences defined in the user model. Since the option tree, and consequently the generation so far, have primarily been based on the user model, if the algorithm notices an inconsistency it will give priority to the most recent user query as a situational adjustment (see Sec. 4.2), and then update the user model to reflect the adjustment. If the user model is modified, the algorithm rebuilds the option tree, and generates the next turn using the new user model and the most recent user query.

To ensure the user that the system has understood her correctly, and to increase her confidence, the algorithm includes a summary of all attribute values that are common to the cluster in the option tree that is being used to generate the summary. For example, for the second turn in the dialogue of Fig. 4, although the user's most recent query was about `cuisine`, the system includes a summary of all the attributes already mentioned to the user: "All of them are cheap, are near the city centre, and have a high star rating".

5 Example

To clarify things further, here we present an example. A dialogue between the UMSR system and a student (user model of Table 1) is depicted in Fig. 4.

Fig. 5 depicts the tree structure for the student user of our example. The user begins by

U: Tell me about the restaurants you have.
S: I found 12 restaurants which are inexpensive. Only 7 of them are near the city centre, out of which only 4 have a high star rating. I also have information about cuisine and service quality.
U: Tell me about the cuisine.
S: There is only 1 restaurant that serves Scottish food, and 3 restaurants that serve French food. They all provide good service quality. All of them are cheap, are near the city centre, and have a high star rating. For a restaurant that serves Indian cuisine you would have to go farther from the city centre.
U: Okay, tell me about the Scottish one.
S: The Jam House's menu includes Scottish dishes sparked with international flavour, with a decent choice of vegetarian dishes. It is located in the centre. The average price is £22 per person.
U: What about Indian restaurants?
S: I found 2 restaurants that serve Indian food. Both of them are cheap, within average distance from the city centre, and have a high star rating. Only one of them provides good service quality.
U: Okay, tell me about this one.
S: The Raj on the Shore's theme extends from the soldier figurines behind the exotic blue bar to the predominantly Bangladeshi, Indian menu. It serves Indian and Bangladeshi cuisine. It is located in the north. The average price is £20 per person.

Figure 4: Example dialogue with our UMSR approach (student user of Table 1).

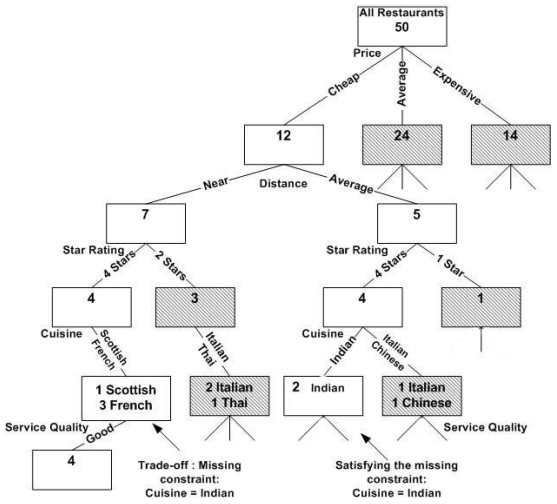


Figure 5: A sample option tree structure for the student user of Table 1. Pruned nodes are shown as shaded.

asking generally for a restaurant recommendation, and the system provides a summary based on the three top-ranking attributes for this user: `price`, `distance` and `star rating`, which correspond to the first three levels of the option tree. Next the user asks about `cuisine type`, which generates a trade-off since there are no Indian restaurants (user's preference) that are cheap, near the city centre, and of high star rating. The user then asks about the Scottish option, before switching back to her preferred cuisine type (Indian). Because Indian cuisine was in the user's initial preference model, a constraint of `cuisine=Indian` was generated when traversing the leftmost branch of the tree, and this justified not pruning the unshaded nodes in the right subtree of Fig. 5, in order to generate the trade-off. However, if the user had asked about expensive restaurants, then a new database query would have been made and a new option tree would have been built. A more complex example is given in the Appendix.

6 Experimental Design

In total 18 subjects interacted with our two systems. Each participant interacted three times with the modified TownInfo system, and another three times with the system that supported our implementation of the UMSR model (108 dialogues in total). The order of the dialogues was randomised among the subjects. Each experiment took between 40 and 50 minutes on average.

For each task, subjects were provided with the user profile and the actual scenario for the specific task in hand. The tasks were carefully constructed so that half of them could be solved without making any trade-offs and the other half required a trade-off to be made. At the end of each task the subjects had to fill out a questionnaire with 10 questions on a 7-point Likert scale. They were also asked if they had been able to accomplish the given task (perceived task completion), i.e., to find a suitable restaurant for the scenario and user profile in hand. Finally, after each task they had to provide the name(s) of the restaurants they chose for the task. The name(s) stated for this task were then used to compare perceived task completion with actual task completion. At the end of each task with the UMSR system, the profiles were reset to the default attribute values and ranks.

Both systems had identical software configurations, i.e., they only differed in the information presentation component. Yet another important feature was that the UMSR based model did not accept multiple attributes in a single query. So for instance the user could not ask "I am looking for a moderately priced restaurant near the city centre that serves Italian food". This seemed to be a major shortcoming of the UMSR based system compared to the TownInfo system with sequential information presentation. However, as we will see in the following, even with this shortcom-

System	U	CC	CF	A	E
UMSR-all	5.04	4.65	3.22	3.66	4.69
TownInfo-all	4.87	4.04	2.93	3.20	3.59
UMSR-with TO	4.74	4.59	2.67	3.26	4.15
TownInfo-with TO	4.59	3.41	2.74	2.33	2.70
UMSR-no TO	5.33	4.70	3.78	4.08	5.22
TownInfo-no TO	5.15	4.67	3.11	4.07	4.48

Table 2: Average scores of the questionnaires for all dialogues, dialogues with trade-offs (with TO) and dialogues without trade-offs (no TO) (U=understandability, CC=conciseness, CF=confidence, A=accessibility, E=efficiency).

ing the UMSR approach retained its advantages and proved more successful than the traditional sequential enumeration approach.

7 Results

The perceived task completion (PTC) for the UMSR system and the TownInfo system was 90.74% and 85.19% respectively, and the actual task completion (ATC) 74.07% and 62.96%. Thus the UMSR approach led to a relatively better user confidence in having achieved the task.

The average number of turns was 9.24 for UMSR compared to 17.78 for TownInfo, which denotes a significant reduction in the number of dialogue turns required to accomplish a given task. This reduction becomes even more prominent when there is a trade-off involved. With such dialogues, the average number of turns for UMSR remained almost constant at 9.41, whereas TownInfo showed an increase reaching up to 24.19. This huge difference is obviously a significant improvement in system efficiency and user satisfaction. It also supports our hypothesis that the UMSR approach can present trade-offs understandably. For dialogues without a trade-off the number of turns was 9.07 for UMSR and 11.37 for TownInfo.

Dialogue duration also showed a great improvement in UMSR over TownInfo (4:49 (m:s) vs. 6:11). The duration however was almost the same for the two systems when a trade-off existed (4:40 vs. 4:49). This could mean that although the number of turns in this case is smaller for UMSR, the length of the generated output is longer, and requires more attention to understand. Yet again in dialogues without a trade-off, UMSR had a considerably shorter duration than TownInfo (4:57 vs. 7:34).

Average scores of the questionnaires are given in Table 2.

In response to the question "I thought the way the system provided information to me was easy to understand" the average score over all 108 dialogues was 5.04 for UMSR and 4.87 for TownInfo. The preference for UMSR exists for dialogues both with and without a trade-off. However, for all three cases the differences were not significant ($p > 0.05$).

Conciseness is the quality of providing a concise overview of all the available options to the user. The UMSR system was preferred at 4.65 over 4.04 for TownInfo ($p = 0.034$). The difference between the two systems is very significant for dialogues with a trade-off ($p < 0.003$). However, for dialogues without a trade-off $p = 0.92$. This was predictable as the main innovation in UMSR is the ability to present trade-offs in a concise and understandable way, hence the significant difference for the dialogues with trade-offs.

To evaluate their confidence in having heard all the relevant options, the subjects were asked to rate the statement "I thought there were better options for my request than what the system gave me". Because of the negative nature of the question, the Likert scale was inverted before analysis. The average score was 3.22 and 2.93 for UMSR and TownInfo respectively. This indicates that the users have slightly more confidence in having heard all the relevant options with the UMSR system, although this difference is not significant ($p > 0.05$). For dialogues with a trade-off, the average confidence score was slightly better for TownInfo (2.74 vs. 2.67), but not significant ($p = 0.8$). However, there is a significant difference for dialogues without a trade-off ($p < 0.03$). Another notable issue is the overall low scores for the cases with a trade-off. This signifies that perhaps more information needs to be given to the user for dialogue turns describing a trade-off. A careful balance needs to be drawn between conciseness and comprehensiveness in these cases. This however, will obviously increase dialogue duration, and might affect understandability.

By accessibility, we mean ease of use and communication with the system. The scores for UMSR and TownInfo were 3.66 and 3.20 respectively ($p = 0.18$). A more significant difference in accessibility was noted for dialogues with a trade-off ($p = 0.008$). Again it seemed that users preferred UMSR when it came down to dealing with trade-offs. However, the accessibility scores for dialogues without a trade-off were almost the same ($p = 0.92$).

Efficiency is the quality of enabling users to find the optimal option quickly. The statement "In this task, the system allowed me to find the optimal restaurant quickly", resulted in an average score of 4.69 for UMSR vs. 3.59 for TownInfo ($p = 0.002$). Once again, a significant difference was noted for dialogues with a trade-off, with 4.15 and 2.70 for UMSR and TownInfo respectively ($p = 0.004$). However, the difference for dialogues without a trade-off was not significant ($p = 0.12$).

8 Conclusions and Future Work

In this paper, we evaluated the effectiveness of the UMSR approach in information presentation in a full end-to-end dialogue system. The UMSR approach was compared with the traditional sequential enumeration of options. Our results verified our hypothesis that the UMSR approach presents a better overview of the trade-offs within the option space, and improves user experience and confidence in the system. Furthermore, with the UMSR approach there is a significant reduction in the number of dialogue turns required to complete the task. The results also showed that UMSR specifically outperforms TownInfo when there is a trade-off involved. The UMSR results presented statistically significant improvement for conciseness, accessibility, and efficiency. Overall, subjects were more satisfied with the UMSR system. When they were asked if they would use the system again as a deployed product the score was 4.74 for UMSR and 3.70 for TownInfo ($p = 0.002$), further verifying that the users preferred the UMSR approach over the sequential enumeration of TownInfo.

In future work we intend to make a number of improvements. For example in the turn generation algorithm, we will optimise the generated output in an effort to strike a balance between understandability and complexity. Another important issue is to modify the UMSR algorithm so that it can accept multiple attributes in a single query. Moreover, we will perform experiments with both speech input and output. Finally, we will compare the UMSR approach with the UM and SR approaches in the same setting, i.e., a full end-to-end SDS.

Acknowledgements

This paper is based on a research experiment conducted at the University of Edinburgh. Paksima was funded by the European Commission Erasmus Mundus scholarship program. Georgila was partially funded by the Wellcome Trust VIP Award and is currently funded 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. The authors thank the three anonymous reviewers.

References

G. Carenini and J.D. Moore. 2001. An empirical study of the influence of user tailoring on evaluative argument effectiveness. In *Proc. of IJCAI*.

G. Chung. 2004. Developing a flexible spoken dialog system using simulation. In *Proc. of ACL*.

V. Demberg and J.D. Moore. 2006. Information presentation in spoken dialogue systems. In *Proc. of EACL*.

S. Elzer, J. Chu-Carroll, and S. Carberry. 1994. Recognizing and utilizing user preferences in collaborative consultation dialogues. In *Proc. of UM '94*.

J. Henderson, O. Lemon, and K. Georgila. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed datasets. *Computational Linguistics*, 34(4):487–511.

O. Lemon, K. Georgila, J. Henderson, and M. Stuttle. 2006. An isu dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the talk in-car system. In *Proc. of EACL*.

J.D. Moore, M.E. Foster, O. Lemon, and M. White. 2004. Generating tailored, comparative descriptions in spoken dialogue. In *Proc. of FLAIRS*.

J. Polifroni and M. Walker. 2008. Intensional summaries as cooperative responses in dialogue: Automation and evaluation. In *Proc. of ACL–HLT*.

J. Polifroni, G. Chung, and M. Walker. 2003. Towards automatic generation of mixed-initiative dialogue systems from web content. In *Proc. of Eurospeech*.

M.A. Walker, R.J. Passonneau, and J.E. Boland. 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Proc. of ACL*.

M. Walker, S. Whittaker, A. Stent, P. Maloor, J.D. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in dialogue. *Cognitive Science*, 28:811–840.

A. Winterboer and J.D. Moore. 2007. Evaluating information presentation strategies for spoken recommendations. In *RecSys '07: Proc. of the 2007 ACM conference on Recommender systems*.

Appendix

Option Tree A for User Model A is built.
U: I'd like to get some food tonight.
S: I found 17 restaurants with a high star rating. Only 8 of them provide good service quality, out of which only 3 serve Scottish food. I also have information about distance and price.
U: Are they close to the city centre?
S: I found 3 restaurants that are located far from the city centre. 1 of them is cheap and 2 are expensive. They all have a high star rating, provide good service quality, and serve Scottish food. For a restaurant near the city centre you would have to go for average service quality.
U: How about French cuisine?
The System cannot answer the query because the "French"cuisine branches of Option Tree A have been pruned and their nodes are shown as shaded.
User Model A changes to User Model B (situational query).
Option Tree B for User Model B is built.
S: I found 10 restaurants that serve French food. Only 5 of them have a high star rating, out of which only 3 provide good service quality. I also have information about distance and price.
U: Tell me about the distance from the city centre.
S: Only 1 of them is located near the city centre and is inexpensive. It serves French food, has a high star rating, and provides good service quality.
U: Tell me about it.
S: La Bagatelle's dishes include a starter of roast pigeon with canelloni of shiitake mushroom, and a main course of panfried veal with prunes.
U: That's great. I'll take it.
...

Figure 6: Example dialogue with our UMSR approach for a business person, User Model A (Table 3) and User Model B (Table 4), Option Tree A (Fig. 7) and Option Tree B (Fig. 8).

UserID	Attribute	Value	Rank
2	Star	High	1.00
2	Service	Good	2.00
2	Cuisine	Scottish	3.00
2	Distance	Near	4.00
2	Price	Don't Care	5.00

Table 3: Sample user model for a business person (User Model A).

UserID	Attribute	Value	Rank
2	Cuisine	French	0.95
2	Star	High	1.00
2	Service	Good	2.00
2	Distance	Near	4.00
2	Price	Don't Care	5.00

Table 4: Modified user model for a business person after the situational query "I'd like a French restaurant" (User Model B).

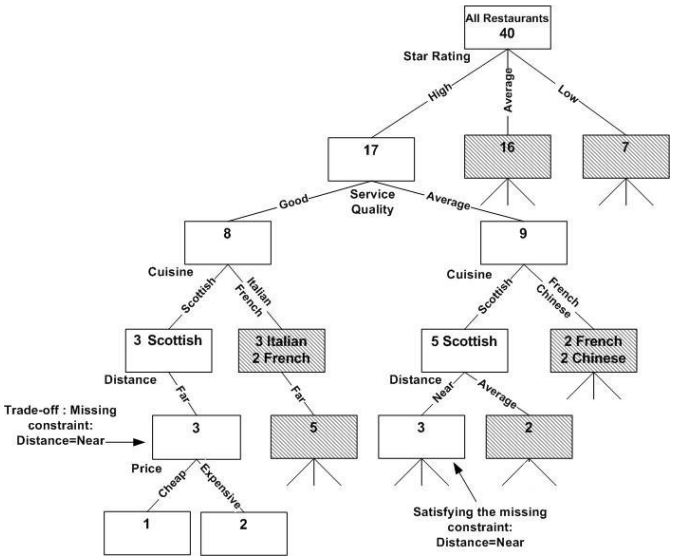


Figure 7: Option tree structure (Option Tree A) corresponding to the User Model A of Table 3. Pruned nodes are shown as shaded.

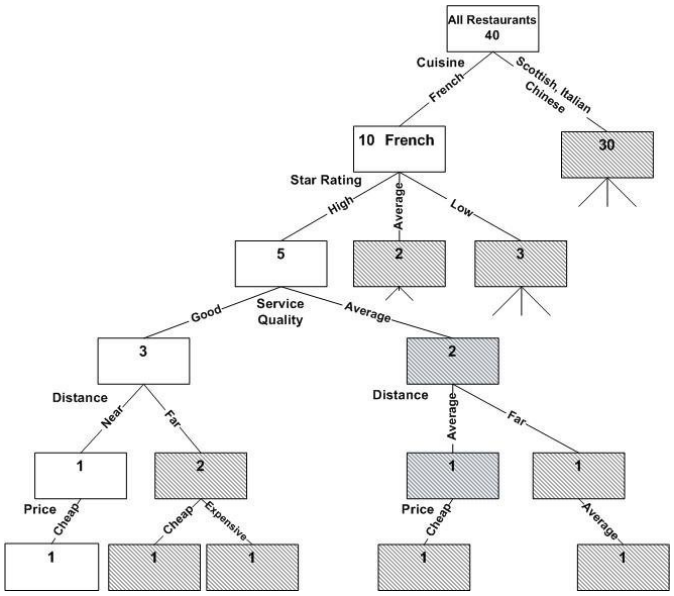


Figure 8: Option tree structure (Option Tree B) corresponding to the User Model B of Table 4. Pruned nodes are shown as shaded.

Can I finish? Learning when to respond to incremental interpretation results in interactive dialogue

David DeVault and **Kenji Sagae** and **David Traum**
USC Institute for Creative Technologies
13274 Fiji Way
Marina del Rey, CA 90292
{devault,sagae,traum}@ict.usc.edu

Abstract

We investigate novel approaches to responsive overlap behaviors in dialogue systems, opening possibilities for systems to interrupt, acknowledge or complete a user's utterance while it is still in progress. Our specific contributions are a method for determining when a system has reached a point of maximal understanding of an ongoing user utterance, and a prototype implementation that shows how systems can use this ability to strategically initiate system completions of user utterances. More broadly, this framework facilitates the implementation of a range of overlap behaviors that are common in human dialogue, but have been largely absent in dialogue systems.

1 Introduction

Human spoken dialogue is highly interactive, including feedback on the speech of others while the speech is progressing (so-called "backchannels" (Yngve, 1970)), monitoring of addressees and other listener feedback (Nakano et al., 2003), fluent turn-taking with little or no delays (Sacks et al., 1974), and overlaps of various sorts, including collaborative completions, repetitions and other grounding moves, and interruptions. Interruptions can be either to advance the new speaker's goals (which may not be related to interpreting the other's speech) or in order to prevent the speaker from finishing, which again can be for various reasons. Few of these behaviors can be replicated by current spoken dialogue systems. Most of these behaviors require first an ability to perform incremental interpretation, and second, an ability to predict the final meaning of the utterance.

Incremental interpretation enables more rapid response, since most of the utterance can be interpreted before utterance completion (Skantze and Schlangen, 2009). It also enables giving early feedback (e.g., head nods and shakes, facial expressions, gaze shifts, and verbal backchannels) to signal how well things are being perceived, understood, and evaluated (Allwood et al., 1992).

For some responsive behaviors, one must go beyond incremental interpretation and predict some aspects of the full utterance before it has been completed. For behaviors such as complying with the evocative function (Allwood, 1995) or intended perlocutionary effect (Sadek, 1991), grounding by demonstrating (Clark and Schaefer, 1987), or interrupting to avoid having the utterance be completed, one must predict the semantic content of the full utterance from a partial prefix fragment. For other behaviors, such as timing a reply to have little or no gap, grounding by saying the same thing at the same time (called "chanting" by Hansen et al. (1996)), performing collaborative completions (Clark and Wilkes-Gibbs, 1986), or some corrections, it is important not only to predict the meaning, but also the form of the remaining part of the utterance.

We have begun to explore these issues in the context of the dialogue behavior of virtual human (Rickel and Johnson, 1999) or embodied conversational agent (Cassell et al., 2000) characters for multiparty negotiation role-playing (Traum et al., 2008b). In these kinds of systems, human-like behavior is a goal, since the purpose is to allow a user to practice this kind of dialogue with the virtual humans in training for real negotiation dialogues. The more realistic the characters' dialogue behavior is, the more kinds of negotiation situations can be adequately trained for. We discuss these sys-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 11–20,
Queen Mary University of London, September 2009.

tems further in Section 2.

In Sagae et al. (2009), we presented our first results at prediction of semantic content from partial speech recognition hypotheses, looking at length of the speech hypothesis as a general indicator of semantic accuracy in understanding. We summarize this previous work in Section 3.

In the current paper, we incorporate additional features of real-time incremental interpretation to develop a more nuanced prediction model that can accurately identify moments of maximal understanding within individual spoken utterances (Section 4). We demonstrate the value of this new ability using a prototype implementation that collaboratively completes user utterances when the system becomes confident about how the utterance will end (Section 5). We believe such predictive models will be more broadly useful in implementing responsive overlap behaviors such as rapid grounding using completions, confirmation requests, or paraphrasing, as well as other kinds of interruptions and multi-modal displays. We conclude and discuss future work in Section 6.

Figure 1: SASO-EN negotiation in the cafe: Dr. Perez (left) looking at Elder al-Hassan.

$$\begin{bmatrix} mood : declarative \\ sem : \begin{bmatrix} type : event \\ agent : captain - kirk \\ event : deliver \\ theme : power - generator \\ modal : [\, possibility : can \,] \\ speech - act : [\, type : offer \,] \end{bmatrix} \end{bmatrix}$$

Figure 2: AVM utterance representation.

2 Domain setting

The case study we present in this paper is taken from the SASO-EN scenario (Hartholt et al., 2008; Traum et al., 2008b). This scenario is designed to allow a trainee to practice multi-party negotiation skills by engaging in face to face negotiation with virtual humans. The scenario involves a negotiation about the possible re-location of a medical clinic in an Iraqi village. A human trainee plays the role of a US Army captain, and there are two virtual humans that he negotiates with: Doctor Perez, the head of the NGO clinic, and a local village elder, al-Hassan. The doctor's main objective is to treat patients. The elder's main objective is to support his village. The captain's main objective is to move the clinic out of the marketplace, ideally to the US base. Figure 1 shows the doctor and elder in the midst of a negotiation, from the perspective of the trainee. Figure A-1 in the appendix shows a sample dialogue from this domain.

The system has a fairly typical set of processing components for virtual humans or dialogue systems, including ASR (mapping speech to words), NLU (mapping from words to semantic frames), dialogue interpretation and management (handling context, dialogue acts, reference and deciding what content to express), NLG (mapping frames to words), non-verbal generation, and synthesis and realization. The doctor and elder use the same ASR and NLU components, but have different modules for the other processing, including different models of context and goals, and different output generators. In this paper, we will often refer to the characters with various terms, including "virtual humans", "agents", or "the system".

In this paper, we are focusing on the NLU component, looking at incremental interpretation based on partial speech recognition results, and the potential for using this information to change the dialogue strategy where warranted, and provide responses before waiting for the final speech result. The NLU output representation is an attribute-value matrix (AVM), where the attributes and values represent semantic information that is linked to a domain-specific ontology and task model (Hartholt et al., 2008). Figure 2 shows an example representation, for an utterance such as "we can provide you with power generators". The AVMs are linearized, using a path-value notation, as shown in Figure 3.

To develop and test the new incremental/prediction models, we are using a corpus of

```
<s>.mood declarative
<s>.sem.type event
<s>.sem.agent captain-kirk
<s>.sem.event deliver
<s>.sem.theme power-generator
<s>.sem.modal.possibility can
<s>.sem.speechact.type offer
```

Figure 3: Example NLU frame.

utterances collected from people playing the role of captain and negotiating with the virtual doctor and elder. In contrast with Figure A-1, which is a dialogue with one of the system designers who knows the domain well, dialogues with naive users are generally longer, and often have a fairly high word error rate (average 0.54), with many out of domain utterances. The system is robust to these kinds of problems, both in terms of the NLU approach (Leuski and Traum, 2008; Sagae et al., 2009) as well as the dialogue strategies (Traum et al., 2008a). This is accomplished in part by approximating the meaning of utterances. For example, the frame in Figure 3 is also returned for an utterance of *we are prepared to give you guys generators for electricity downtown* as well as the ASR output for this utterance, *we up apparently give you guys generators for a letter city don town*.

3 Predicting interpretations from partial recognition hypotheses

Our NLU module, mxNLU (Sagae et al., 2009), is based on maximum entropy classification (Berger et al., 1996), where we treat entire individual frames as classes, and extract input features from ASR. The training data for mxNLU is a corpus of approximately 3,500 utterances, each annotated with the appropriate frame. These utterances were collected from user sessions with the system, and the corresponding frames were assigned manually. Out-of-domain utterances (about 15% of all utterances in our corpus) could not be mapped to concepts in our ontology and task model, and were assigned a "garbage" frame. For each utterance in our corpus, we have both a manual transcription and the output of ASR, although only ASR is used by mxNLU (both at training and at runtime). Each training instance for mxNLU consists of a frame, paired with a set of features that represent the ASR output for user utterances. The specific features used by the classifier are: each word in the input string (bag-of-words representation of the input), each bigram (pairs of consecutive words), each pair of any two words in the input, and the number of words in the input string.

In the 3,500-utterance training set, there are 136 unique frames (135 that correspond to the semantics of different utterances in the domain, plus one frame for out-of-domain utterances).[1] The NLU task is then framed as a multiclass classification approach with 136 classes, and about 3,500 training examples.

Although mxNLU produces entire frames as output, we evaluate NLU performance by looking at precision and recall of the attribute-value pairs (or *frame elements*) that compose frames. Precision represents the portion of frame elements produced by mxNLU that were correct, and recall represents the portion of frame elements in the gold-standard annotations that were proposed by mxNLU. By using precision and recall of frame elements, we take into account that certain frames are more similar than others and also allow more meaningful comparative evaluation with NLU modules that construct a frame from subelements or for cases when the actual frame is not in the training set. The precision and recall of frame elements produced by mxNLU using complete ASR output are 0.78 and 0.74, respectively, for an F-score (harmonic mean of precision and recall) of 0.76.

3.1 NLU with partial ASR results

The simplest way to perform NLU of partial ASR results is simply to process the partial utterances using the NLU module trained on complete ASR output. However, better results may be obtained by training separate NLU models for analysis of partial utterances of different lengths. To train these separate NLU models, we first ran the audio of the utterances in the training data through our ASR module, recording all partial results for each utterance. Then, to train a model to analyze partial utterances containing N words, we used only partial utterances in the training set containing N words (unless the entire utterance contained less than N words, in which case we simply used the complete utterance). In some cases, multiple partial ASR results for a single utterance

[1] In a separate development set of 350 utterances, annotated in the same way as the training set, we found no frames that had not appeared in the training set.

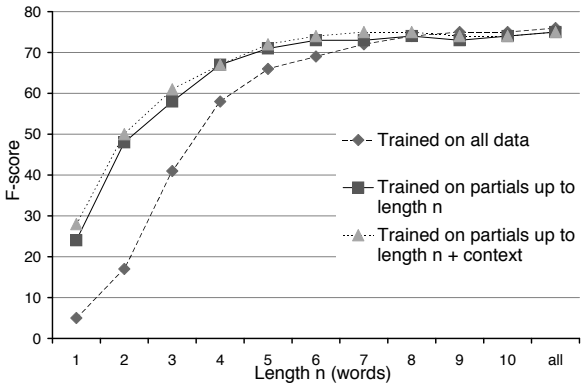


Figure 4: F-score for three NLU models on partial ASR results up to N words.

contained the same number of words, and we used the last partial result with the appropriate number of words.[2] We trained ten separate partial NLU models for N varying from one to ten.

Figure 4 shows the F-score for frames obtained by processing partial ASR results up to length N using three variants of mxNLU. The dashed line is our baseline NLU model, trained on complete utterances only, and the solid line shows the results obtained with length-specific NLU models. The dotted line shows results for length-specific models that also use features that capture aspects of dialogue context. In these experiments, we used unigram and bigram word features extracted from the most recent system utterance to represent context, but found that these context features did not improve NLU performance. Our final NLU approach for partial ASR hypotheses is then to train separate models for specific lengths, using hypotheses of that length during training (solid line in figure 4).

4 How well is the system understanding?

In this section, we present a strategy that uses machine learning to more closely characterize the performance of a maximum entropy based incremental NLU module, such as the mxNLU module described in Section 3. Our aim is to identify strategic points in time, as a specific utterance is occurring, when the system might react with confidence that the interpretation will not signif-

[2] At run-time, this can be closely approximated by taking the partial utterance immediately preceding the first partial utterance of length $N + 1$.

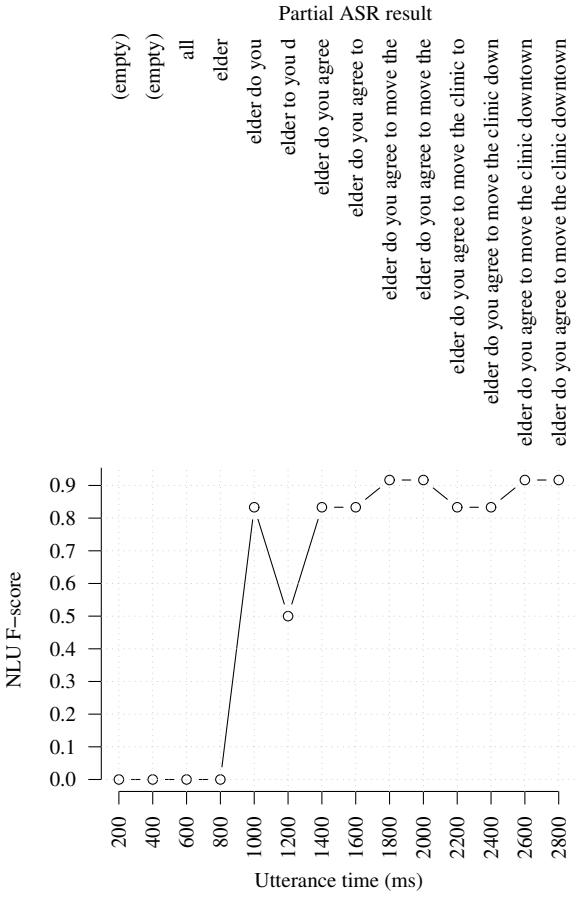


Figure 5: Incremental interpretation of a user utterance.

icantly improve during the rest of the utterance. This reaction could take several forms, including providing feedback, or, as described in Section 5 an agent might use this information to opportunistically choose to initiate a completion of a user's utterance.

4.1 Motivating example

Figure 5 illustrates the incremental output of mxNLU as a user asks, *elder do you agree to move the clinic downtown?* Our ASR processes captured audio in 200ms chunks. The figure shows the partial ASR results after the ASR has processed each 200ms of audio, along with the F-

score achieved by mxNLU on each of these partials. Note that the NLU F-score fluctuates somewhat as the ASR revises its incremental hypotheses about the user utterance, but generally increases over time.

For the purpose of initiating an overlapping response to a user utterance such as this one, the agent needs to be able (in the right circumstances) to make an assessment that it has already understood the utterance "well enough", based on the partial ASR results that are currently available. We have implemented a specific approach to this assessment which views an utterance as understood "well enough" if the agent would not understand the utterance any better than it currently does even if it were to wait for the user to finish their utterance (and for the ASR to finish interpreting the complete utterance).

Concretely, Figure 5 shows that after the entire 2800ms utterance has been processed by the ASR, mxNLU achieves an F-score of 0.91. However, in fact, mxNLU already achieves this maximal F-score at the moment it interprets the partial ASR result *elder do you agree to move the* at 1800ms. The agent therefore could, in principle, initiate an overlapping response at 1800ms without sacrificing any accuracy in its understanding of the user's utterance.

Of course the agent does not automatically realize that it has achieved a maximal F-score at 1800ms. To enable the agent to make this assessment, we have trained a classifier, which we call MAXF, that can be invoked for any specific partial ASR result, and which uses various features of the ASR result and the current mxNLU output to estimate whether the NLU F-score for the current partial ASR result is at least as high as the mxNLU F-score would be if the agent were to wait for the entire utterance.

4.2 Machine learning setup

To facilitate the construction of our MAXF classifier, we identified a range of potentially useful features that the agent could use at run-time to assess its confidence in mxNLU's output for a given partial ASR result. These features are exemplified in the appendix in Figure A-2, and include: K, the number of partial results that have been received from the ASR; N, the length (in words) of the current partial ASR result; Entropy, the entropy in the probability distribution mxNLU assigns to alternative output frames (lower entropy corresponds to a more focused distribution); $P_{\max}$, the probability mxNLU assigns to the most probable output frame; NLU, the most probable output frame (represented for convenience as fI, where I is an integer index corresponding to a specific complete frame). We also define MAXF (GOLD), a boolean value giving the ground truth about whether mxNLU's F-score for this partial is at least as high as mxNLU's F-score for the final partial for the same utterance. In the example, note that MAXF (GOLD) is true for each partial where mxNLU's F-score ($F(K)$) is $\geq$ 0.91, the value achieved for the final partial (*elder do you agree to move the clinic downtown*). Of course, the actual F-score $F(K)$ is not available at run-time, and so cannot serve as an input feature for the classifier.

Our general aim, then, is to train a classifier, MAXF, whose output predicts the value of MAXF (GOLD) as a function of the input features. To create a data set for training and evaluating this classifier, we observed and recorded the values of these features for the 6068 partial ASR results in a corpus of ASR output for 449 actual user utterances.[3]

We chose to train a decision tree using Weka's J48 training algorithm (Witten and Frank, 2005).[4] To assess the trained model's performance, we carried out a 10-fold cross-validation on our data set.[5] We present our results in the next section.

4.3 Results

We will present results for a trained decision tree model that reflects a specific precision/recall tradeoff. In particular, given our aim to enable an agent to sometimes initiate overlapping speech, while minimizing the chance of making a wrong assumption about the user's meaning, we selected a model with high precision at the expense of lower recall. Various precision/recall tradeoffs are possible in this framework; the choice of a specific tradeoff is likely to be system and domain-dependent and motivated by specific design goals.

We evaluate our model using several features which are exemplified in the appendix in Figure A-3. These include MAXF (PREDICTED), the trained MAXF classifier's output (TRUE or

[3] This corpus was not part of the training data for mxNLU.

[4] Of course, other classification models could be used.

[5] All the partial ASR results for a given utterance were constrained to lie within the same fold, to avoid training and testing on the same utterance.

FALSE) for each partial; K_{MAXF}, the first partial number for which MAXF (PREDICTED) is TRUE; $\Delta F(K) = F(K) - F(K_{\mathrm{final}})$, the "loss" in F-score associated with interpreting partial K rather than the final partial K_{final} for the utterance; $T(K)$, the remaining length (in seconds) in the user utterance at each partial.

We begin with a high level summary of the trained MAXF model's performance, before discussing more specific impacts of interest in the dialogue system. We found that our trained model predicts that MAXF = TRUE for at least one partial in 79.2% of the utterances in our corpus. For the remaining utterances, the trained model predicts MAXF = FALSE for all partials. The precision/recall/F-score of the trained MAXF model are 0.88/0.52/0.65 respectively. The high precision means that 88% of the time that the model predicts that F-score is maximized at a specific partial, it really is. On the other hand, the lower recall means that only 52% of the time that F-score is in fact maximized at a given partial does the model predict that it is.

For the 79.2% of utterances for which the trained model predicts MAXF = TRUE at some point, Figure 6 shows the amount of time in seconds, $T(K_{\mathrm{MAXF}})$, that remains in the user utterance at the time partial K_{MAXF} becomes available from the ASR. The mean value is 1.6 seconds; as the figure shows, the time remaining varies from 0 to nearly 8 seconds per utterance. This represents a substantial amount of time that an agent could use strategically, for example by immediately initiating overlapping speech (perhaps in an attempt to improve communication efficiency), or by exploiting this time to plan an optimal response to the user's utterance.

However, it is also important to understand the cost associated with interpreting partial K_{MAXF} rather than waiting to interpret the final ASR result K_{final} for the utterance. We therefore analyzed the distribution in $\Delta F(K_{\mathrm{MAXF}}) = F(K_{\mathrm{MAXF}}) - F(K_{\mathrm{final}})$. This value is at least 0.0 if mxNLU's output for partial K_{MAXF} is no worse than its output for K_{final} (as intended). The distribution is given in Figure 7. As the figure shows, 62.35% of the time (the median case), there is no difference in F-score associated with interpreting K_{MAXF} rather than K_{final}. 10.67% of the time, there is a loss of -1, which corresponds to a completely incorrect frame at K_{MAXF} but a completely correct frame at K_{final}. The converse also happens 2.52% of the time: mxNLU's output frame is completely correct at the early partial but completely incorrect at the final partial. The remaining cases are mixed. While the median is no change in F-score, the mean case is a loss in F-score of -0.1484. This is the mean penalty in NLU performance that could be paid in exchange for the potential gain in communication efficiency suggested by Figure 6.

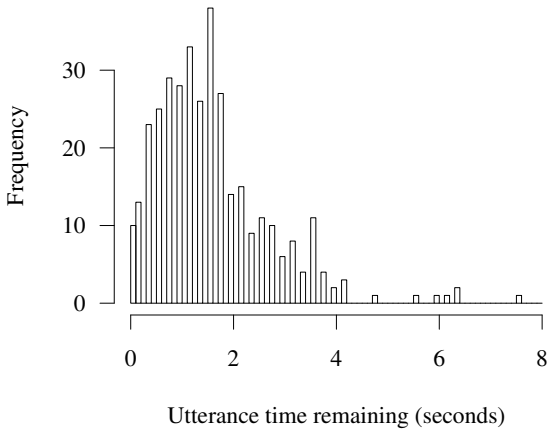


Figure 6: Distribution of $T(K_{\mathrm{MAXF}})$.

$\Delta F(K_{\mathrm{MAXF}})$ range	Percent of utterances
-1	10.67%
$(-1, 0)$	17.13%
0	62.35%
$(0, 1)$	7.30%
1	2.52%
mean($\Delta F(K_{\mathrm{MAXF}})$)	-0.1484
median($\Delta F(K_{\mathrm{MAXF}})$)	0.0000

Figure 7: The distribution in $\Delta F(K_{\mathrm{MAXF}})$, the "loss" associated with interpreting partial K_{MAXF} rather than K_{final}.

5 Prototype implementation

To illustrate one use of the techniques described in the previous sections, we have implemented a prototype module that performs *user utterance completion*. This allows an agent to jump in during a user's utterance, and say a completion of the utterance before it is finished, at a point when the agent

thinks it understands what the user means. This type of completion is often encountered in human-human dialogue, and may be used, for example, for grounding or for bringing the other party's turn to a conclusion.

We have equipped one of our virtual humans, Doctor Perez, with an ability to perform completions as follows. The first step is for the agent to recognize when it understands what the user wants to say. As discussed in Sections 3 and 4, this often happens before the user has completed the utterance. NLU is performed on partial ASR hypotheses as they become available, and MAXF decides whether the agent's understanding of the current partial hypothesis is likely to improve given more time. Once MAXF indicates that the agent's understanding is likely to be already maximized for the utterance, we take the current partial ASR hypothesis and attempt to generate text to complete it in a way that is fluent and agrees with the meaning of the utterance the user has in mind.

The generation of the surface text for completions takes advantage of the manual transcriptions in the corpus of utterances used to train the NLU module. For each frame that the agent understands, our training set contains several user utterances that correspond to the meaning in that frame. At the point where the agent is ready to formulate a completion, mxNLU has already predicted a frame for the user's utterance (even though it is still incomplete). We then consider only the set of known utterances that correspond to that frame as possible sources of completions. As a simple distance metric, we compute the word error rate (WER) between the current partial hypothesis for the user's utterance and a prefix of each of these known utterances. In our prototype, these prefixes have the same length as the current partial ASR hypothesis. We then select the utterance whose prefix has the lowest WER against the current partial ASR hypothesis. As a final step, we look in the prefix of our selected utterance for the last occurrence of the last word in the partial ASR, and if such a word is found, we take the remainder of the utterance as the agent's completion. Considering only the set of utterances that correspond to the frame predicted by mxNLU makes it likely that the completion will have the appropriate meaning. Since the completion is a suffix of a transcript of a previous user utterance, and this suffix follows the last word uttered by the user, it is likely to form a fluent completion of the user's partial utterance.

For 356 utterances in our development set, the procedure above generated completions containing at least one word for 177 utterances. We list below four representative examples of user utterance completions generated by Doctor Perez:

1. Partial utterance: *we need to*
   Predicted completion: *move your clinic*
   Actual user completion: *move the clinic*

2. Partial utterance: *I have orders*
   Predicted completion: *to move you and this clinic*
   Actual user completion: *to help you in moving the clinic to a new location*

3. Partial utterance: *the market*
   Predicted completion: *is not safe*
   Actual user completion: *is not a safer location*

4. Partial utterance: *we can also*
   Predicted completion: *give you medical supplies*
   Actual user completion: *build you a well*

Although agent completions for user utterances are often acceptably accurate (examples 1-3), occasionally they are incorrect (example 4). The frequency of appropriate completions depends on the accuracy of mxNLU and the chosen precision/recall tradeoff in MAXF. Finally, although the agent has the ability to generate these completions, clearly it should not complete the user's utterance at every opportunity. Determining a policy that results in natural behavior with respect to the frequency of completions for different types of agents is a topic under current investigation.

6 Summary and future work

We have presented a framework for interpretation of partial ASR hypotheses of user utterances, and high-precision identification of points within user utterances where the system already understands the intended meaning. Our initial implementation of an utterance completion ability for a virtual human serves to illustrate the capabilities of this framework, but only scratches the surface of the new range of dialogue behaviors and strategies it allows.

Immediate future work includes the design of policies for completions and interruptions that re-

sult in natural conversational behavior. Other applications of this work include the generation of paraphrases that can be used for grounding, in addition to extra-linguistic behavior during user utterances, such as head nods and head shakes.

Acknowledgments

The project or effort described here 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 also like to thank Anton Leuski for facilitating the use of incremental speech results, and David Schlangen and the ICT dialogue group, for helpful discussions.

References

Jens Allwood, Joakim Nivre, and Elisabeth Ahlsen. 1992. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9.

Jens Allwood. 1995. An activity based approach to pragmatics. Technical Report (GPTL) 75, Gothenburg Papers in Theoretical Linguistics, University of Göteborg.

Adam L. Berger, Stephen D. Della Pietra, and Vincent J. D. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors. 2000. *Embodied Conversational Agents*. MIT Press, Cambridge, MA.

Herbert H. Clark and Edward F. Schaefer. 1987. Collaborating on contributions to conversation. *Language and Cognitive Processes*, 2:1–23.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39. Also appears as Chapter 4 in (Clark, 1992).

Herbert H. Clark. 1992. *Arenas of Language Use*. University of Chicago Press.

B. Hansen, D. Novick, and S. Sutton. 1996. Prevention and repair of breakdowns in a simple task domain. In *Proceedings of the* AAAI-96 Workshop on Detecting, Repairing, and Preventing Human-Machine Miscommunication, pages 5–12.

A. Hartholt, T. Russ, D. Traum, E. Hovy, and S. Robinson. 2008. A common ground for virtual humans: Using an ontology in a natural language oriented virtual human architecture. In *Language Resources and Evaluation Conference (LREC)*, May.

A. Leuski and D. Traum. 2008. A statistical approach for text processing in virtual humans. In *26th Army Science Conference*.

Yukiko I. Nakano, Gabe Reinstein, Tom Stocky, and Justine Cassell. 2003. Towards a model of face-to-face grounding. In *ACL*, pages 553–561.

Jeff Rickel and W. Lewis Johnson. 1999. Virtual humans for team training in virtual reality. In *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, pages 578–585. IOS Press.

H. Sacks, E. A. Schegloff, and G. Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50:696–735.

M. D. Sadek. 1991. Dialogue acts are rational plans. In *Proceedings of the ESCA/ETR workshop on multi-modal dialogue*.

K. Sagae, G. Christian, D. DeVault, and D. R. Traum. 2009. Towards natural language understanding of partial speech recognition results in dialogue systems. In Short Paper Proceedings of NAACL HLT.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of EACL 2009*, pages 745–753.

D. Traum, W. Swartout, J. Gratch, and S. Marsella. 2008a. A virtual human dialogue model for non-team interaction. In L. Dybkjaer and W. Minker, editors, *Recent Trends in Discourse and Dialogue*. Springer.

D. R. Traum, S. Marsella, J. Gratch, J. Lee, and A. Hartholt. 2008b. Multi-party, multi-issue, multi-strategy negotiation for multi-modal virtual agents. In Helmut Prendinger, James C. Lester, and Mitsuru Ishizuka, editors, *IVA*, volume 5208 of *Lecture Notes in Computer Science*, pages 117–130. Springer.

I. H. Witten and E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Victor H. Yngve. 1970. On getting a word in edgewise. In *Papers from the Sixth Regional Meeting*, pages 567–78. Chicago Linguistic Society.

A Appendix

1	C	Hello Doctor Perez.
2	D	Hello captain.
3	E	Hello captain.
4	C	Thank you for meeting me.
5	E	How may I help you?
6	C	I have orders to move this clinic to a camp near the US base.
7	E	We have many matters to attend to.
8	C	I understand, but it is imperative that we move the clinic out of this area.
9	E	This town needs a clinic.
10	D	We can't take sides.
11	C	Would you be willing to move downtown?
12	E	We would need to improve water access in the downtown area, captain.
13	C	We can dig a well for you.
14	D	Captain, we need medical supplies in order to run the clinic downtown.
15	C	We can deliver medical supplies downtown, Doctor.
16	E	We need to address the lack of power downtown.
17	C	We can provide you with power generators.
18	E	Very well captain, I agree to have the clinic downtown.
19	E	Doctor, I think you should run the clinic downtown.
20	D	Elder, the clinic downtown should be in an acceptable condition before we move.
21	E	I can renovate the downtown clinic, Doctor.
22	D	OK, I agree to run the clinic downtown, captain.
23	C	Excellent.
24	D	I must go now.
25	E	I must attend to other matters.
26	C	Goodbye.
26	D	Goodbye.
26	E	Farewell, sir.

Figure A-1: Successful negotiation dialogue between C, a captain (human trainee), D, a doctor (virtual human), and E, a village elder (virtual human).

		MAXF model training features					
Partial ASR result	$F(K)$	K	N	Entropy	$P_{\max}$	NLU	MAXF (GOLD)
(empty)	0.00	1	0	2.96	0.48	f82	FALSE
(empty)	0.00	2	0	2.96	0.48	f82	FALSE
all	0.00	3	1	0.82	0.76	f72	FALSE
elder	0.00	4	1	0.08	0.98	f39	FALSE
elder do you	0.83	5	3	1.50	0.40	f68	FALSE
elder to you d	0.50	6	3	1.31	0.75	f69	FALSE
elder do you agree	0.83	7	4	1.84	0.35	f68	FALSE
elder do you agree to	0.83	8	5	1.40	0.61	f68	FALSE
elder do you agree to move the	0.91	9	7	0.94	0.49	f10	TRUE
elder do you agree to move the	0.91	10	7	0.94	0.49	f10	TRUE
elder do you agree to move the clinic to	0.83	11	9	1.10	0.58	f68	FALSE
elder do you agree to move the clinic down	0.83	12	9	1.14	0.66	f68	FALSE
elder do you agree to move the clinic downtown	0.91	13	9	0.50	0.89	f10	TRUE
elder do you agree to move the clinic downtown	0.91	14	9	0.50	0.89	f10	TRUE

Figure A-2: Features used to train the MAXF model.

		MAXF model evaluation features		
K	$F(K)$	$\Delta F(K)$	$T(K)$	MAXF (PREDICTED)
1	0.00	-0.91	2.6	FALSE
2	0.00	-0.91	2.4	FALSE
3	0.00	-0.91	2.2	FALSE
4	0.00	-0.91	2.0	FALSE
5	0.83	-0.08	1.8	FALSE
6	0.50	-0.41	1.6	FALSE
7	0.83	-0.08	1.4	FALSE
8	0.83	-0.08	1.2	FALSE
9 ($= K_{\mathrm{MAXF}}$)	0.91	0.00 (=$\Delta F(K_{\mathrm{MAXF}})$)	1.0	TRUE
10	0.91	0.00	0.8	TRUE
11	0.83	-0.08	0.6	FALSE
12	0.83	-0.08	0.4	FALSE
13	0.91	0.00	0.2	TRUE
14	0.91	0.00	0.0	TRUE

Figure A-3: Features used to evaluate the MAXF model.

Are You Being Addressed? - real-time addressee detection to support remote participants in hybrid meetings

Harm op den Akker
Roessingh Research and Development
Enschede
the Netherlands
h.opdenakker@rrd.nl

Rieks op den Akker
Human Media Interaction Twente
Enschede
the Netherlands
infrieks@cs.utwente.nl

Abstract

In this paper, we describe the development of a meeting assistant agent that helps remote meeting participants by notifying them when they are being addressed. We present experiments that have been conducted to develop machine classifiers to decide whether "you are being addressed" where "you" refers to a fixed (remote) participant in a meeting. The experimental results back up the choices made regarding the selection of data, features, and classification methods. We discuss variations of the addressee classification problem that have been considered in the literature and how suitable they are for addressee detection in a system that plays a role in a live meeting.

1 Introduction

In order to understand what is going on in a meeting, it is important to know who is talking, what is being said, and who is being addressed (talked to). Here, we focus on the question of whom the speech is addressed to. We present results obtained in developing a classifier for real-time addressee prediction to be used in an assistant for a remote participant in a *hybrid* meeting, a meeting where a number of participants share a common meeting room and one or more others take part via teleconferencing software.

It is obvious that in order to effectively participate in a meeting, participants need to know who is being addressed at all times. For remote participants in hybrid meetings, understanding the course of the conversation can be difficult due to the fact that it is hard to figure out who is being addressed. But it is not only meeting participants who are interested in addressees. The question who is being addressed has long been of interest for science: group therapists (Bales, 1950), small group research, or *outside observers* who analyse recorded meetings.

How speakers address listeners, what kind of procedures speakers use to designate their audience and to make clear whom they address has been the focus of conversational analysis, sociolinguistics and ethnomethodology for quite some time. An analysis of addressee selection is presented in (Lerner, 1996). Addressing as a special type of multi-modal interactional referring expression generation behavior is considered in (op den Akker and Theune, 2008).

The problem of *automatic addressee detection* is one of the problems that come up when technology makes the move from *two-party* man-machine natural dialogue systems to systems for *multi-party* conversations. In this context the addressing problem was raised by Traum (2004).

Since Jovanović (2004), presented her research on addressee prediction in meetings at SigDial, quite a few publications on the topic appeared. Jovanović used a number of multi-modal meeting corpora developed in the European projects M4 and AMI. In (Jovanović et al., 2006b) the first multi-modal multi-party corpus containing hand labeled addressee annotations was presented. The public release of the multi-modal AMI meeting corpus (Carletta, 2007; McCowan et al., 2005), a hour annotated corpus of small group meetings has already shown to be an important achievement for research; not only for conversational speech recognition and tracking of visual elements

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 21–28,
Queen Mary University of London, September 2009.

but also for automatic multi-modal conversational scene analysis. The M4 and AMI corpora are the only multi-modal meeting corpora (partly) annotated with addressee labels. Addressee detection in robot-human interaction is studied in (Katzenmaier et al., 2004) and in multi-party dialogue systems in (Knott and Vlugter, 2008; van Turnhout et al., 2005; Bakx et al., 2003; Rickel et al., 2002). Addressing in face-to-face conversations is achieved by multi-modal behavior and addressee detection is thus a multi-modal recognition task. This task requires not only speech recognition but also gaze and gesture recognition, the recognition of deictic references, and, ideally, the understanding of the "what's going on" in the meeting. It requires the detection of who is involved in current (parallel) activities. Speakers show explicit addressing behavior when they are not confident that the participants they want to address are paying attention to their words. Analysis of the remote meetings recorded in the EC project AMIDA reinforces our experiences that this happens more in remote meetings than in small group face-to-face meetings.

In AMIDA, the European follow-up project of AMI, the two new research goals are: (1) real-time processing (real-time speech recognition (Hain et al., 2008), focus of attention recognition (Ba and Odobez, 2009), real-time dialogue act labeling (Germesin et al., 2008) and addressee detection); and (2) technology for (remote) meeting support. Technology based on the analysis of how people behave and converse in meetings is now going to re-shape the meetings, and hopefully make them more effective and more engaging. Social interaction graphs that show who is talking to whom and how frequently in a meeting may help the group by mirroring its interpersonal relations, dominance, and group dynamics, and understand social mechanisms as possible causes of ineffectiveness. Although, feedback about the social interactions may also be useful *during* meetings, it doesn't require the prediction of the speaker's addressees in real-time. A participant in a meeting, however, needs to know who is being addressed by the speaker *at "the time of speaking"*. This holds for humans as well as for an artificial partner, a robot or a virtual Embodied Conversational Agent in a multi-party conversation.

The problem of addressee prediction comes in different flavors, depending on the relations that the subject who is in need of an answer, has with the event itself. *Time* is one of the aspects that play a role here: whether the subject needs to know the addressee of an utterance in real-time or offline. But it is not only time that plays a role. The addressing problem is an *interactional problem*, meaning that it is determined by the role that the subject has in the interaction itself; if and how the speaker and others communicate with each other and with the subject. Is he himself a possible addressee of the speaker or is he an outside observer? What type of communication channels are available to the subject and which channels of communication are available to the conversational partners in the meeting? It is often harder to follow a face-to-face discussion on the radio than to follow a radio broadcasted multi-party discussion that was held via a point-to-point telephone connection.

What speakers do to make clear whom they are addressing depends on the status and capacities of the communication lines with their interlocutors. Discussion leaders in TV shows are aware of their TV audience. Every now and then, they explicitly address their *virtual* audience at home. They also design their questions so as to make clear to the TV viewer whom their questions are addressed to. Outside observers in the form of a video camera will, however, not affect the way speakers make clear whom they address as long as the camera is not considered as a participant interested in the speaker's intention. Because remote participants are often out of sight, speakers in the meeting room do not take them into account when they converse to others in the meeting room. Remote participants become a kind of outside observers and share the same problems that annotators have when they watch video recordings of meetings to see what is happening in the meeting and who is being addressed by the speaker.

In section 2 we will specify the particular type of addressing problem that we are trying to tackle here. We make clear how our problem and approach differ from those of other researchers and what this means for the applicability of previous results and available data. In section 3 we present the data we used for testing and training. We set a baseline for the performance of our classifiers as

well as a hypothesized maximum value, or ceiling, based on the complexity of the task at hand. In section 4 we discuss the experiments, for selecting the optimal features, classifiers, and parameters. In section 5 we present the experimental results. In section 6 we discuss how the currently implemented addressing module works in the meeting assistant and what is required to use all the features of the addressee predictor in a hybrid meeting.

2 The Addressing Problem Considered Here

Jovanović et al. (2004) and Jovanović et al. (2006a) describe the classifiers that have been trained and tested on the M4 and AMI corpora. The classification problem is to assign an addressee label to a dialogue act, a hand-labeled and hand-segmented sequence of words, which is obtained by manual transcription of a speaker's utterance. The output of the classifier is one of a set of possible addressee labels: Group, or P0,P1,P2,P3, which are the four fixed positions around the table of the four participants in the meeting. Since the AMI data contains several meetings of different groups of four people, the class value cannot be the name of a participant, as that is not an invariant of the meeting setting. Positions at the rectangular table are invariant. This implies that the classifiers can only be used for meetings with this setting and four participants. A comparison of the statistical classifier of Jovanović with a rule-based method using the same part of the AMI corpus is presented in (op den Akker and Traum, 2009). The same data is also used by Gupta et al. (2007) in their study of a related problem: finding the person the speaker refers to when he uses a second person pronoun (e.g. 'you' or 'your') as a deictic referring expression. Their class values are not positions at the table but "virtual positions" in the speaking order (e.g. next speaker, previous speaker), a solution that generalises to a broader class of conversations than four participants in a face-to-face meeting. In a more recent study, Frampton et al. (2009) use positions at the table relative to the position of the speaker as class values: L1, L2, L3. The reason for this is to alleviate the problem of class imbalance in the corpus.

We will also use the AMI corpus but we will look at a different variant of the addressing problem. This is motivated by our application: to support a remote participant in a hybrid meeting. The question that we will try to answer is "are you being addressed?", where "you" refers to an individual participant in a conversation. The possible answers we consider are "yes" or "no"[1]. The addressing classifier that solves this problem is thus dedicated to a personal buddy. Note that this makes the method useable for any type of conversational setting. Note also that the addressing prediction problem "are you being addressed?" for a meeting assistant who is not himself participating in the meeting is different from the problem "am I being addressed?" that a participant himself may have to solve. The meeting assistant does not have direct "internal" knowledge about the processes or attentiveness of his buddy participant; he has to rely on outside observations. Our view on the problem implies that we have to take another look at the AMI data and that we will analyse and use it in a different way for training, testing and performance measuring. It also implies that we cannot rely for our binary classification problem on the results of Jovanović (2007) with (dynamic) Bayesian networks.

3 The Data and How Complex Our Task Is

We use a subset of the AMI corpus, containing those fourteen meetings that have not only been annotated with dialogue acts, but where dialogue acts are also attributed an addressee label, telling if the speaker addresses the Group, or the person sitting at position P0,P1,P2 or P3[2]. They have also been annotated with visual focus of attention: at any time it is known for each partner where he is looking and during what time frame. Annotated gaze targets are persons in the meeting, whiteboard, laptop, table or some other object.

Another level of annotations that we use concerns the topic being discussed during a topic segment of the meeting. Participants in the AMI corpus play a role following a scenario, the group has to design a remote TV control and team members each have one of four roles in the design project: PM - project manager; UI - user interface designer; ID - industrial designer; or ME - marketing expert. For details on the meeting scenario see

[1] A 'yes' means that the dialogue act is addressed to 'you' only. Group-addressed dialogue acts are considered to be 'no' (not addressed to you only).

[2] Annotators could also use label *Unknown* in case they could not decide the addressee of the speaker, this is treated as Group-addressed or 'no'.

(Post et al., 2004). In training and testing the classifiers we alternately take up the position in the meeting of one of the participants, who is treated as the target for addressee prediction.

3.1 Base-line and Ceiling-value

Because most of the dialogue acts are not specifically addressed to one and the same meeting participant, the baseline for the binary classification task is already quite high: , being the percentage of all dialogue acts annotated with addressing information "*not* addressed to You", which is out of a total of dialogue acts.

The performance of a supervised machine learning method depends on (1) the selection of features (2) the type of classifier including the settings of the hyper-parameters of the classifiers (Daelemans et al., 2003), and (3) the quality and the amount of training data (Reidsma, 2008; Reidsma and Carletta, 2008). Since we measure the classifier's performance with a part of the annotated data it is interesting to see how human annotators (or, 'human classifiers') perform on this task.

One of the AMI meetings[3] has been annotated with addressing information by four different annotators. We will use this to measure how ambiguous the task of addressee labeling is. Table 1 shows the confusion matrix for two annotators: *s95* and *vka*. This shows the (dis-)agreements for labelling the dialogue acts as addressed to A, B, C, D or to the Group. [4] However, because we use our data differently, we will look at the confusion matrices in a different way. We split it up into 4 matrices, each from the view of one of the four meeting participants. Table 2 is an example of this, taking the view of participant A (i.e. for the binary decision task "is **Participant A** being addressed?", and having annotator s95 as gold standard.

Table 2 shows that when taking annotator *s95* as gold standard, and considering annotator *vka* as the classifier, he achieves an accuracy of (380 out of 412 instances classified correctly).

[3] IS1003d

[4] Note that the annotators first independently segmented the speaker's turns into dialogue act segments; then labeled them with a dialogue act type label and then labeled the dialogue acts with an addressee label. The dialogues acts are those segments that both annotators identified as a dialogue act segment.

	A	B	C	D	Group	Total
A	**29**				10	39
B		**14**			8	22
C			**32**		7	39
D	1		1	**49**	18	69
Group	21	10	19	22	**171**	243
Total	51	24	52	71	214	412

Table 1: Confusion matrix for one pair of annotators ( ).

	A		Total
A	**29**	10	39
	22	**351**	373
Total	51	361	412

Table 2: Confusion matrix for one pair of annotators, considering addressed to A or not (derived from the matrix in Table 1).

We can argue that we can use these human annotators/classifiers scores as a measure of "maximum performance", because it indicates a level of task ambiguity. Classifiers can achieve higher scores, because they can learn through noise in the data. Thus, the inter-annotator confusion value is not an absolute limit of actual performance, but cases in which the classifier is "right" and the test-set "wrong" would not be reflected in the results. Since the inter-annotator confusion does also say something about the inherent task ambiguity, it can be used as a measure to compare a classifier score with. Table 3 contains the overall scores (taken over all 4 individual participants) for the 6 annotator pairs. The average values for Recall, Precision, F-Measure and Accuracy in Table 3 are considered as *ceiling* values for the performance measures for this binary classification task[5]. The Hypothesized Maximum Score (HMS) is the average accuracy value: .

Pair	Rec	Prec	F	Acc
s-v	73.37	62.63	67.58	92.78
m-s	59.75	70.59	64.72	91.87
m-v	69.92	74.78	72.27	93.11
m-d	37.77	81.61	51.64	91.79
v-d	42.04	80.49	55.23	92.22
s-d	43.68	77.55	55.88	93.02
Average:	**54.42**	**74.61**	**61.22**	**92.47**

Table 3: Recall, Precision, F-measure and Accuracy values for the 6 pairs of annotators.

[5] Inter-changing the roles of the two annotators, i.e. consider vka as "gold standard" in Table 2, means inter-changing the Recall and Precision values. The F-value remains the same, though.

The baseline (for all dialogue acts annotated with addressing) and the HMS ( ) accuracy values will be used for comparison with the performance of our classifiers.

4 The Methods and Their Features

In the experiments, four different classifiers were created:

1. Lexical and Context Classifier

2. Visual Focus of Attention Classifier

3. Combined Classifier

4. Topic and Role Extended Classifier

For each of these classifiers a large number of experiments were performed with a varying number of 15 to 30 different machine learning methods -using Weka (Witten and Frank, 1999)- to select optimal feature sets. In this section we summarize the most important findings. For a more detailed analysis refer to (op den Akker, 2009). Because of the large number of features and classifiers used, the various classifier hyper parameters have largely been kept to their default values. Where it was deemed critical (Neural Network training epochs and number of trees in RandomForest classifier) these parameters were varied afterwards to make sure that the performance did not deviate too much from using the default values. It didn't.

4.1 Lexical and Context Classifier

The lexical and context based classifier uses features that can be derived from words and dialogue acts only. A total of 14 features were defined, 7 of which say something about the dialogue act (type, number of words, contains 1st person singular personal pronoun, and so on) and 7 of which say something about the context of the dialogue act (how often was I addressed in the previous 6 dialogue acts, how often did I speak in the previous 5 dialogue acts, and so on). Of these 14 features, the optimal feature subset was selected by trying out all the subsets. This was repeated using 15 different classifiers from the WEKA toolkit. The best result was achieved with a subset of 10 features, by the MultiLayerPerceptron classifier. In this way an accuracy of 90.93 was reached. Given the baseline of the used train and test set of 89.20 and the HMS of 92.47, this can be seen as 53% of what 'can' be achieved.

4.2 Visual Focus of Attention Classifier

The VFOA classifier uses features derived from a meeting participant's visual focus of attention. A total of 8 features were defined, such as: the total time that the speaker looks at me, the total time everyone is looking at me, and so on. The optimal time interval in which to measure who is looking at you was extensively researched by trying out different intervals around the start of a dialogue act, and training and testing a classifier on the feature. These optimal interval values differ for every feature, but is usually somewhere between a few seconds before the start of the dialogue act, to 1 second into the dialogue act. The difference in performance for using the optimal interval compared to using the start- and end times of the dialogue act is sometimes as much as 0.93 accuracy (which is a lot given a base score of 89.20 and HMS of 92.47). This shows, that when looking at VFOA information, one should take into account the participant's gaze before the dialogue act, instead of looking at the utterance duration as in (Jovanović, 2007; Frampton et al., 2009)[6]. The representation of feature values was also varied by either normalizing to the duration of the window or using the raw values. Again the optimal feature subset was calculated using brute-force. Because of the reduced time complexity for possible feature subsets, 30 different classifiers from the WEKA toolkit were trained and tested. One of the best results was achieved with a feature set of 4 features again with the MultiLayerPerceptron: 90.80 accuracy. The train and test sets used for this classifier are slightly smaller than those used for the LexCont classifier because not all dialogue acts are annotated with VFOA. The base score for the data here is 89.24, and given the HMS of 92.47, this result can be seen as 48% of what can be achieved.

4.3 Combined Classifier

The third classifier is a combination of the first two. We tried three different methods of combining the results of the LexCont and VFOA classifiers. First we tried to train a classifier using all the features (14 lexical, 8 vfoa) which exploded the feature subset search space to over 4 million possibilities. A second approach was to combine the output of the LexCont and VFOA classifiers using a simple rule-based approach. The OR-rule

[6]Note that a dialogue act segment can be preceded by an other utterance unit of the same speaker.

(if either of the two classifiers thinks the DA is addressed to you, the outcome is 'yes') performed the best (91.19% accuracy). But the best results were achieved by training a rule based (Ridor) classifier on the output of the first two. For these experiments the test-set of the previous two classifiers was split again into a new train (3080 instances) and test set (1540 instances). The features are the outputs of the VFOA and LexCont classifiers (both class and class-probabilities). For this task, 35 classifiers have been trained with the best results coming from the Ridor classifier: 92.53 accuracy. The results of all the different techniques for combining the classifiers can be seen in Table 4. The baseline score for this smaller test set is 89.87, so given the HMS of 92.47, this result can be seen as 102% of what can be achieved. Note that this is not 'impossible', because the Hypothesized Maximum Score is merely an indication of how humans perform on the task, not an absolute ceiling.

4.4 Topic and Role Extended Classifier

As a final attempt to improve the results we used topic and role information as features to our combined classifier. In the AMI corpus, every meeting participant has a certain role (project manager, interface designer, etc...) and the meetings were segmented into broad topic (opening, discussion, industrial designer presentation). Now the idea is that participants with certain roles are more likely to be addressed during certain topics. As an illustration of how much these a-priori chances of being addressed can change, take the example of an industrial designer during an 'industrial designer presentation'. The a-priori probability of you being addressed as industrial designer in the entire corpus is 13%. This probability, given also the fact that the current topic is 'industrial designer presentation' becomes 46%. This is a huge difference, and this information can be exploited. For all combinations of topic and role, the a-priori probability of you being addressed as having that role and during that topic, have been calculated. These values have been added as features to the features used in the Combined Classifier, and the experiments have been repeated. This time, the best performing classifier is Logistic Model Trees with an accuracy of 92.99%. Given the baseline of 89.87 and HMS of 92.47, this can be seen as 120% of what 'can' be achieved, which is better by a fairly large margin than the results of the inter-annotator agreement values.

5 Summary of Results

Table 4 summarizes the results for the various classifiers. The LexCont and VFOA classifiers individually achieve only about 50% of what can be achieved, but if combined in a clever way, their performance seems to reach the limit of what is possible based on the comparison with inter-annotator agreement. The fact that the topic-role extended classifier achieves so much more than 100% can be ascribed to the fact that it is cheating. It uses pre-calculated a-priori chances of 'you' being addressed given the circumstances. This knowledge could be calculated by the machine learner by feeding it the topic and role features, and letting it learn these a-priori probabilities for itself. But the classifier that uses these types of features can not easily be deployed in any different setting, where participants have different roles and where different topics are being discussed.

Method	Acc	Rec	Prec	F	PoM
HMS	92.47	54.42	74.61	61.22	-
LexCont	90.93	33.10	66.02	44.09	53
VFoA	90.80	27.77	67.65	39.38	48
CombinedFeat	91.56	36.62	70.82	48.28	72
ClassOfResults	43.68	77.55	55.88	93.02	102
LogComb(AND)	90.24	9.86	94.23	17.85	31
LogComb(OR)	91.19	47.08	61.90	53.48	60
TopicRoleExt	92.99	41.03	80.00	54.24	120

Table 4: Performance values of the Methods discussed in this paper: Accuracy, Recall, Precision, F-measure and Percentage of Hypothesized Maximum Score (PoM).

6 How Does The Assistant Work?

At the time of writing, the assistant that has been implemented is based on the simple visual focus of attention classifier. The focus of attention is inferred from the head pose and head movements of a participant in the meeting room who is being observed by a close-up camera. The real-time focus of attention module sends the coordinates of the head pose to a central database 15 times per second (Ba and Odobez, 2009). The coordinates are translated into targets: objects and persons in the meeting room. For the addressing module most important are the persons and in particular the screen in the meeting room where the remote

participant is visible. The addressing module is notified of updates of who is speaking and decides whether the remote participant is being looked at by the speaker.

If the remote participant (RP) is not attentive (which can be detected automatically based on his recent activity) he is called when he is addressed or when the real-time keyword spotter has detected a word or phrase that occurs on the list of topics of interest to the RP. For a detailed description of the remote meeting assistant demonstrator developed in the AMIDA project refer to (op den Akker et al., 2009).

The meeting assistant allows the RP to distribute his attention over various tasks. The system can give a transcript of the fragment of the meeting that is of interest to the RP, so he can catch up with the meeting if he was not following. The simple focus of attention based addressing module works fine. The question is now if an addressing module that uses the output of the real-time dialogue act recognizer, which in turn uses the output of the real-time speech recognizer will outperform the visual focus of attention based addressee detector. Experiments make us rather pessimistic about this: the performance drop of state of the art real-time dialogue segmentation and labeling technology based on real-time ASR output is too large in comparison with those based on hand-annotated transcripts (Jovanović, 2007). For real-time automatic addressee detection more superficial features need to be used, such as: speech/non-speech, who is speaking, some prosodic information and visual focus of attention, by means of head orientation.

The most explicit way of addressing is by using a vocative, the proper name of the addressed person. In small group face-to-face meetings, where people constantly pay attention and keep track of others' attentiveness to what is being said and done, this method of addressing hardly ever occurs. In remote meetings where it is often not clear to the speaker if others are paying attention, people call other's names when they are addressing them. Other properties of the participant relevant for addressee detection include his role and his topics of interest. These can either be obtained directly from the participant when he subscribes for the meeting, or they can be recognized during an introduction round that most business meetings start with. For automatic topic detection further analysis of the meeting will be needed (Purver et al., 2007). Probability tables for the conditional probabilities of the chance that someone with a given role is being addressed when the talk is about a given topic, can be obtained from previous data, and could be updated on the fly during the meeting. Only when that has been achieved will it be possible for our extended topic/role addressee classifier to be fully exploited by a live meeting assistant.

Acknowledgements

The research of the first author was performed when he was a Master's student at the Human Media Interaction group of the University of Twente. This work is supported by the European IST Programme Project FP6-0033812 (AMIDA). We are gratefull to the reviewers of SigDial 2009 for their encouraging comments, and to Lynn Packwood for correcting our English.

References

Sileye Ba and Jean-Marc Odobez. 2009. Recognizing human visual focus of attention from head pose in meetings. In *IEEE Transaction on Systems, Man, and Cybernetics, Part B (Trans. SMC-B)*, volume 39, pages 16–33.

I. Bakx, K. van Turnhout, and J. Terken. 2003. Facial orientation during multi-party interaction with information kiosks. In *Proceedings of 9th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT)*, Zurich, Switzerland.

Robert Freed Bales. 1950. *Interaction Process Analysis; A Method for the Study of Small Groups*. Addison Wesley, Reading, Mass.

Jean C. Carletta. 2007. Unleashing the killer corpus: experiences in creating the multi-everything AMI meeting corpus. *Language Resources and Evaluation*, 41(2):181–190, May.

Walter Daelemans, Véronique Hoste, Fien De Meulder, and Bart Naudts. 2003. Combined optimization of feature selection and algorithm parameter interaction in machine learning of language. In *Proceedings of the 14th European Conference on Machine Learning (ECML-2003)*, Lecture Notes in Computer Science 2837, pages 84–95, Cavtat-Dubrovnik, Croatia. Springer-Verlag.

Matthew Frampton, Raquel Fernndez, 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 *Proceedings of the 12th Conference of the EACL*.

Sebastian Germesin, Tilman Becker, and Peter Poller. 2008. Determining latency for on-line dialog act classification. In *Poster Session for the 5th International Workshop on Machine Learning for Multimodal Interaction*, volume 5237.

Surabhi Gupta, John Niekrasz, Matthew Purver, and Daniel Jurafsky. 2007. Resolving "you" in multiparty dialog. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, September.

Thomas Hain, Asmaa El Hannani, Stuart N. Wrigley, and Vincent Wan. 2008. Automatic speech recognition for scientific purposes - webasr. In *Proceedings of the international conference on spoken language processing (Interspeech 2008)*.

Natasa Jovanović and Rieks op den Akker. 2004. Towards automatic addressee identification in multiparty dialogues. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 89–92, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Natasa Jovanović, Rieks op den Akker, and Anton Nijholt. 2006a. Addressee identification in face-to-face meetings. In *Proceedings of 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Trento, Italy.

Natasa Jovanović, Rieks op den Akker, and Anton Nijholt. 2006b. A corpus for studying addressing behaviour in multi-party dialogues. *Language Resources and Evaluation Journal*, 40(1):5–23.

Natasa Jovanović. 2007. *To whom it may concern: adressee identification in face-to-face meetings*. Ph.D. thesis, University of Twente.

M. Katzenmaier, R. Stiefelhagen, and T. Schultz. 2004. Identifying the addressee in human-human-robot interactions based on head pose and speech. In *Proceedings of International Conference on Multimodal Interfaces (ICMI)*, pages 144–151, State College, PA.

A. Knott and P. Vlugter. 2008. Multi-agent human-machine dialogue: issues in dialogue management and referring expression semantics. *Artificial Intelligence*, 172:69–102.

Gene H. Lerner. 1996. On the place of linguistic resources in the organization of talk-in interaction: "Second person" reference in multi-party conversation. *Pragmatics*, 6(3):281–294.

I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. The AMI meeting corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*.

Rieks op den Akker and Mariet Theune. 2008. How do I address you? - modelling addressing behavior based on an analysis of a multi-modal corpus of conversational discourse. In *Proceedings of the AISB 2008 Symposium on Multimodal Output Generation (MOG 2008), Aberdeen, UK*, pages 10–17.

Rieks op den Akker and David Traum. 2009. A comparison of addressee detection methods for multiparty conversations. In *Proceedings of DiaHolmia, 13th Workshop on the Semantics and Pragmatics of Dialogue*.

Rieks op den Akker, Dennis Hofs, Hendri Hondorp, Harm op den Akker, Job Zwiers, and Anton Nijholt. 2009. Engagement and floor control in hybrid meetings. In *Proceedings COST Action Prague 2008 (to appear)*, LNCS. Springer Verlag.

Harm op den Akker. 2009. On addressee detection for remote hybrid meeting settings. Master's thesis, University of Twente.

W.M. Post, A.H. Cremers, and O.B. Henkemans. 2004. A research environment for meeting behavior. In A. Nijholt, T. Nishida, R. Fruchter, and D. Rosenberg, editors, *Social Intelligence Design*, Enschede, The Netherlands.

Matthew Purver, John Dowding, John Niekrasz, Patrick Ehlen, Sharareh Noorbaloochi, and Stanley Peters. 2007. Detecting and summarizing action items in multi-party dialogue. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, September.

Dennis Reidsma and Jean C. Carletta. 2008. Reliability measurement without limits. *Computational Linguistics*, 34(3):319–326, September.

Dennis Reidsma. 2008. *Annotations and Subjective Machines*. Ph.D. thesis, University of Twente.

J. Rickel, S. Marsella, J. Gratch, R. Hill, D. Traum, and W. Swartout. 2002. Towards a new generation of virtual humans for interactive experiences. *Intelligent Systems*, 17:32–36.

David Traum. 2004. Issues in multiparty dialogues. In *Advances in Agent Communication*, pages 201–211.

K. van Turnhout, J. Terken, I. Bakx, and B. Eggen. 2005. Identifying the intended addressee in mixed human-human and human-computer interaction from non-verbal features. In *Proceedings of 7th International Conference on Multimodal Interfaces (ICMI'05)*, Trento, Italy.

Ian H. Witten and Eibe Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1st edition, October.

Invited Talk

What's Unique About Dialogue? Hand gestures, figurative language, facial displays, and direct quotation

Janet Beavin Bavelas
Department of Psychology, University of Victoria, Victoria, B.C., Canada
web.uvic.ca/psyc/bavelas
bavelas@uvic.ca

Face-to-face dialogue is the basic site of language use. Our group's program of research focuses on unique features of face-to-face dialogue, especially the ways in which participants collaborate moment-by-moment (e.g. Bavelas et al., 1995; Bavelas and Chovil, 1997; Bavelas et al., 2000, 2002). Current experiments are showing that the availability of collaborative processes in dialogue significantly affects whether speakers use the modality that Peirce called iconic and Clark and Gerrig (1990) called demonstration. Demonstrations resemble their referents, creating an image for the addressee; for example, hand gestures, facial displays, direct quotation, and figurative language are all demonstrations. We have shown the effect of dialogue on these four kinds of demonstration by using an experimental design with three conditions: a face-to-face dialogue; a dialogue on the telephone; and a monologue to a tape recorder. The first experiment on gesture (Bavelas et al., 2008) showed an independent effect of dialogue, over and above the effect of visibility. The rate of hand gestures was higher in dialogue than in monologue, that is, both the face-to-face and the telephone dialogues had significantly higher rates of gesturing than for the same task in a monologue. Figurative language also showed a dialogue effect; for example, the rate of figurative language was significantly higher in a telephone dialogue than a monologue to a tape recorder. We have subsequently replicated these two effects in a different data set. The second experiment with the same design examined the effects of dialogue on conversational facial displays and direct quotations. Again, the dialogues produced significantly higher rates of these forms of demonstration, while the monologues consisted almost entirely of conventional verbal description. We propose that monologue suppresses both verbal and nonverbal forms of demonstration because demonstrations require an addressee. Current research is investigating which particular feature of speaker-addressee interaction is essential to the use of demonstrations.

References

J. B. Bavelas and N. Chovil. 1997. Faces in dialogue. In J. A. Russell and J. M. Fernandez-Dols, editors, *The psychology of facial expression*, pages 334–346. Cambridge University Press.

J. B. Bavelas, N. Chovil, L. Coates, and L. Roe. 1995. Gestures specialized for dialogue. *Personality and Social Psychology Bulletin*, 21:394–405.

J. B. Bavelas, L. Coates, and T. Johnson. 2000. Listeners as co-narrators. *Journal of Personality and Social Psychology*, 79:941–952.

J. B. Bavelas, L. Coates, and T. Johnson. 2002. Listener responses as a collaborative process: The role of gaze. *Journal of Communication*, 52:566–580.

J. B. Bavelas, J. Gerwing, C. Sutton, and D. Prevost. 2008. Gesturing on the telephone: Independent effects of dialogue and visibility. *Journal of Memory and Language*, 58:495–520.

H. H. Clark and R. J. Gerrig. 1990. Quotations as demonstrations. *Language*, 66:764–805.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, page 29,
Queen Mary University of London, September 2009.

Incremental Reference Resolution: The Task, Metrics for Evaluation, and a Bayesian Filtering Model that is Sensitive to Disfluencies

David Schlangen, Timo Baumann, Michaela Atterer
Department of Linguistics
University of Potsdam, Germany
{das|timo|atterer}@ling.uni-potsdam.de

Abstract

In this paper we do two things: a) we discuss in general terms the task of incremental reference resolution (IRR), in particular resolution of exophoric reference, and specify metrics for measuring the performance of dialogue system components tackling this task, and b) we present a simple Bayesian filtering model of IRR that performs reasonably well just using words directly (no structure information and no hand-coded semantics): it picks the right referent out of 12 for around 50 % of real-world dialogue utterances in our test corpus. It is also able to learn to interpret not only words but also hesitations, just as humans have shown to do in similar situations, namely as markers of references to hard-to-describe entities.

1 Introduction

Like other tasks involved in language comprehension, reference resolution—that is, the linking of natural language expressions to contextually given entities—is performed incrementally by human listeners. This was shown for example by Tanenhaus et al. (1995) in a famous experiment where addressees of utterances containing referring expressions made eye movements towards target objects very shortly after the end of the first word that unambiguously specified the referent, even if that wasn't the final word of the phrase. In fact, as has been shown in later experiments (Brennan and Schober, 2001; Bailey and Ferreira, 2007; Arnold et al., 2007), such disambiguating material doesn't even have to be lexical: under certain circumstances, a speaker's hesitating already seems to be understood as increasing the likelihood of subsequent reference to hard-to-describe entities.

Recently, efforts have begun to build dialogue systems that make use of incremental processing as well (Aist et al., 2006; Skantze and Schlangen, 2009). These efforts have so far focused on aspects other than resolution of references ((Stoness et al., 2004) deals with the interaction of reference and parsing). In this paper, we discuss in general terms the task of incremental reference resolution (IRR) and specify metrics for evaluating incremental components for this task. To make the discussion more concrete, we also describe a simple Bayesian filtering model of IRR in a domain with a small number of possible referents, and show that it performs better wrt. our metrics if given information about hesitations—thus providing computational support for the rationality of including observables other than words into models of dialogue meaning.

The remainder of the paper is structured as follows: We discuss the IRR task in Section 2, and suitable evaluation metrics in Section 3. In Section 4 we describe and analyse the data for which we present results with our Bayesian model for IRR in Section 5.

2 Incremental Reference Resolution

To a first approximation, IRR can be modeled as the 'inverse' as it were of the task of generating referring expressions (GRE; which is well-studied in computational linguistics, see e. g. (Dale and Reiter, 1995)). Where in GRE words are *added* that express features which reduce the size of the set of possible distractors (with which the object that the expression is intended to pick out can be confused), in IRR words are *encountered* that express features that reduce the size of the set of possible

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 30–37,
Queen Mary University of London, September 2009.

referents. To give a concrete example, for the expression in (1-a), we could imagine that the logical representation in (1-b) is built on a word-by-word basis, and at each step the expression is checked against the world model to see whether the reference has become unique.

(1) a. the red cross
b. $\iota x(red(x) \wedge cross(x))$

To give an example, in a situation where there are available for reference only one red cross, one green circle, and two blue squares, we can say that after "the red" the referent should have been found; in a world with two red crosses, we would need to wait for further restricting information (e. g. ". . . on the left").

This is one way to describe the task, then: a component for incremental reference resolution takes expressions as input in a word-by-word fashion and delivers for each new input a set (possibly a singleton set) as output which collects those discourse entities that are compatible with the expression up to that point. (This description is meant to be neutral as to whether reference is exophoric, i. e. directly to entities in the world, or anaphoric, via previous mentions; we will mainly discuss the former case, though.)

As we will see below, this does however not translate directly into a usable metric for evaluation. While it is easy to identify the contributions of individual words in simple, constructed expressions like (1-a), reference in real conversations is often much more complex, and is a collaborative process that isn't confined to single expressions (Clark and Schaefer, 1987): referring is a pragmatic action that is not reducible to denotation. In our corpus (see below), we often find descriptions as in (2), where the speaker continuously adds (rather vague) material, typically until the addressee signals that she identified the item, or proposes a different way to describe it.

(2) Also das S Teil sieht so aus dass es ein einzelnes . Teilchen hat . dann . vier am Stück im rechten Winkel .. dazu nee . nee warte .. dann noch ein einzelnes das guckt auf der anderen Seite raus.
well, the S piece looks so that it has a single . piece . and then . four together in a 90 degree angle .. and also . no .. wait .. and then a single piece that sticks out on the other side.

While it's difficult to say in the individual case what the appropriate moment is to settle on a hypothesis about the intended referent, and what the "correct" time-course of the development of hypotheses is, it's easy to say what we want to be true in general: we want a referent to be found as early as possible, with as little change of opinion as possible during the utterance.[1] Hence a model that finds the correct referent earlier and makes fewer wrong decisions than a competing one will be considered better. The metrics we develop in the next section spell out this idea.

3 Evaluation Metrics for IRR

In previous work, we have discussed metrics for evaluating the performance of incremental speech recognition (Baumann et al., 2009). There, our metrics could rely on time-aligned gold-standard information against which the incremental results could be measured. For the reasons discussed in the previous section, we do not assume that we have such temporally-aligned information for evaluating IRR. Our measures described here simply assume that there is one intention behind the referring utterances (namely to identify a certain entity), and that this intention is there from the beginning of the utterance and stays constant.[2] This is not to be understood as the claim that it is reasonable to expect an IRR component to pick out a referent even if the only part of the utterance that has already been processed for example is "now take the"—it just facilitates the "earlier is better" ranking discussed above.

We use two kinds of metrics for IRR: *positional metrics*, which measure when (which percentage into the utterance) a certain event happens, and *edit metrics* which capture the "jumpiness" of the decision process (how often the component changes its mind during an utterance).

Figure 1 shows a constructed example that il-

[1] We leave open here what "as early as possible" means—a well-trained model might be able to resolve a reference before the speaker even deems that possible, and hence appear to do unnatural (or supernatural?) 'mind reading'. Conversely, frequent changes of opinion might be something that human listeners would exhibit as well (e. g. in their gaze direction). We abstract away from these finer details in our heuristic.

[2] Note that our metrics would also work for corpora where the correct point-of-identification is annotated; this would simply move the reference point from the beginning of the utterance to that point. Gallo et al. (2007) describe an annotation effort in a simpler domain where entities can easily be described which would make such information available.

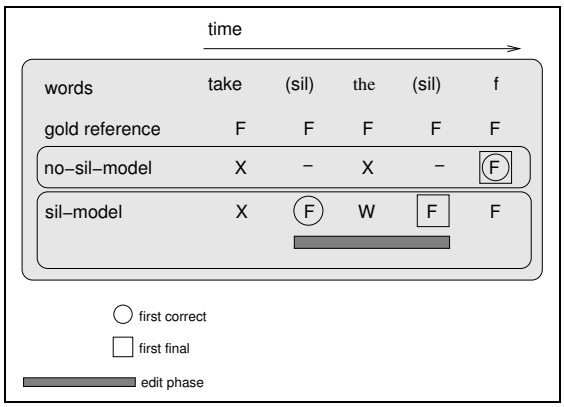


Figure 1: Simple constructed example that illustrates the evaluation measures

lustrates these ideas. We assume that reference is to an object that is internally represented by the letter F. The example shows two models, no-sil and sil (what exactly they are doesn't matter for now). The former model guesses that reference is to object X already after the first word, and stays with this opinion until it encounters the final word, when it chooses F as most likely referent. (Why the decision for the items *sil* is "-" will be explained below; here this can be read as "repetition of previous decision".) The other model changes its mind more often, but also is correct for the first time earlier and stays correct earlier. Our metrics make this observation more precise:

• **average fc** (first correct): how deep into the utterance do we make the first correct guess? (If the decision component delivers n-best lists instead of single guesses, "correct" means here and below "is member of n-best list".)

E. g., if the referent is recognised only after the final word of the expression, the score for this metric would be 1. In our example it is 2/5 for the sil-model and 1 for the non-sil model.

• **fc applicable**: since the previous measure can only be specified for cases where the correct referent has been found, we also specify for how many utterances this is the case.

• **average ff** (first final): how deep into the utterance do we make the correct guess *and* don't subsequently change our mind? This would be 4/5 for the sil-model in our example and 1 for the no-sil-model.

• **ff applicable**: again, the previous measure can only be given where the final guess of the component is correct, so we also need to specify how often this is the case. Note that whenever ff is applicable, fc is applicable as well, so **ff applicable**$\leq$**fc applicable**.

• **ed-utt** (mean edits per utterance): an IRR module may still change its mind even after it has already made a correct guess. This metric measures how often the module changes its mind before it comes back to the right guess (if at all). Since such decision-revisions (edits) may be costly for later modules, which possibly need to retract their own hypotheses that they've built based on the output of this module, ideally this number should be low.

In our example the number of edits between fc and ff is 2 for the sil-model and 0 for the non-sil model (because here fc and ff are at the same position).

• **eo** (edit overhead): ratio unnecessary edits / necessary edits. (In the ideal case, there is exactly one edit, from "no decision" to the correct guess.)

• **correctness**: how often the model guesses correctly. This is 3/5 for the sil-model in the example and 1/5 for the non-sil-model.

• **sil-correctness**: how often the model guesses correctly *during hesitations*. The correctness measure applied only to certain data-points; we use this to investigate whether informing the model about hesitations is helpful.

• **adjusted error**: some of our IRR models can return "undecided" as reply. The correctness measures defined above would punish this in the same way as a wrong guess. The **adjusted error** measure implements the idea that undecidedness is better than a wrong guess, at least early in the utterance. More precisely, it's defined to be 0 if the guess is correct, $pos \,/\, pos_{max}$ if the reply is "undecided" (with pos denoting the position in the utterance), and 1 if the guess is incorrect. That way uncertainty is not punished in the beginning of the utterance and counted like an error towards its end.

Note that these metrics characterise different aspects of the performance of a model. In practical cases, they may not be independent from each other, and a system designer will have to decide which one to optimize. If it is helpful to be informed about a likely referent early, for example to prepare a reaction, and is not terribly costly to later have to revise hypotheses, then a low **first correct** may be the target. If hypothesis revisions are costly, then a low **edit overhead** may be preferred over a low **first correct**. (**first final** and **ff applicable**, however, are parameters that are useful for global optimisation.)

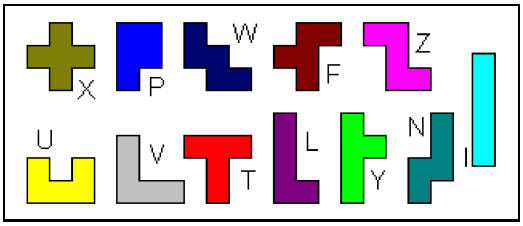


Figure 2: The Twelve Pentomino Pieces with their canonical names (which were not known to the dialogue participants). The pieces used in the dialogues all had the same colour.

In the remaining sections, we describe a probabilistic model of IRR that we have implemented, and evaluate it in terms of these metrics. We begin with describing the data from which we learnt our model.

4 Data

4.1 Our Corpora

As the basis for training and testing of our model we used data from three corpora of task-oriented dialogue that differ in some details of the set-up, but use the same task: an Instruction Giver (IG) instructs an Instruction Follower (IF) on which puzzle pieces (from the "Pentomino" game, see Figure 2) to pick up. In detail, the corpora were:

• The Pento Naming corpus described in (Siebert and Schlangen, 2008). In this variant of the task, IG *records* instructions for an absent IF; so these aren't fully interactive dialogues. The corpus contained 270 utterances out of which we selected those 143 that contained descriptions of puzzle pieces (and not of their position on the gameboard).

• Selections from the FTT/PTT corpus described in (Fernández et al., 2007), where IF and IG are connected through an audio-only connection, and in some dialogues a simplex / push-to-talk one. We selected all utterances from IG that contained references to puzzle pieces (286 altogether).

• The third part of our corpus was constructed specifically for the experiments described here. We set-up a Wizard of Oz experiment where users were given the task to describe puzzle pieces for the "dialogue system" to pick up. The system (i. e. the wizard) had available a limited number of utterances and hence could conduct only a limited form of dialogue. We collected 255 utterances containing descriptions of puzzle pieces in this way.

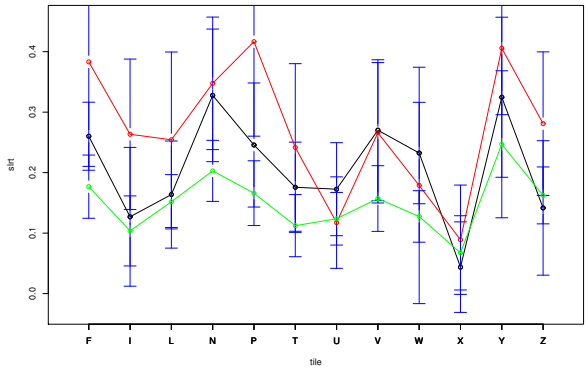


Figure 3: Silence rate per referent and corpus (WOz:black, PentoNaming:red, FTT:green)

All utterances were hand-transcribed and the transcriptions were automatically aligned with the speech data using the MAUS system (Schiel, 2004); this way, we could automatically identify pauses during utterances and measure their length. For some experiments (see below), pauses were "re-ified" through the addition of silence pseudo-words (one for each 333 ms of silence).

The resulting corpus is not fully balanced in terms of available material for the various pieces or contributions by sub-corpora.

4.2 Descriptive Statistics

We were interested to see whether intra-utterance silences (hesitations) could potentially be used as an information source in our (more or less) real-world data in the same way as was shown in the much more controlled situations described in the psycholinguistics literature mentioned above in the introduction (Arnold et al., 2007). Figure 3 shows the mean ratio of within-utterance silences per word for the different corpora and different referents. We can see that there are clear differences between the pieces. For example, references to the piece whose canonical name is X contain very few or short hesitations, whereas references to Y tend to contain many. We can also see that the tendencies seem to be remarkably similar between corpora, but with relatively stable offsets between them, PentoDescr having the longest, PTT/FTT the shortest silences. We speculate that this is the result of the differing degrees of interactivity (none in PentoDescr, restricted in WOz, less restricted in PTT, free in FTT) which puts different pressures on speakers to avoid silences. To balance our data with respect to this difference, we performed some experiments with adjusted data

where silence lengths in PentoDescr were adjusted by 0.7 and in PTT/FTT by 1.3. This brings the silence rates in the corpora, if plotted in the style of Figure 3, almost in congruence.

To test whether the differences in silence rate between utterances referring to different pieces are significant, we performed an ANOVA and found a main effect of silence rate, $F(11, 672) = 6.2102, p < 8.714^{-10}$. A post-hoc t-test reveals that there are roughly two groups whose members are not significantly different within-group, but are across groups: I, L, U, W and X form one group with relatively low silence rate, F, N, P, T, V, Y, and Z another with relatively high silence rate. We will see in the next section whether our model picked up on these differences.

5 A Bayesian Filtering Model of IRR

To explore incremental reference resolution, and as part of a larger incremental dialogue system we are building, we implemented a probabilistic reference resolver that works in the pentomino domain. At its base, the resolver has a Bayesian Filtering model (see e. g. (Thrun et al., 2005)) that with each new observation (word) computes a belief distribution over the available objects (the twelve puzzle pieces); in a second step, a decision for a piece (or a collection of pieces in the n-best case) is derived from this distribution. This model is incremental in a very natural and direct way: new input increments are simply treated as new observations that update the current belief state. Note that this model does not start with any assumptions about semantic word classes: whether an observed word carries information about what is being referred to will be learnt from data.

5.1 The Belief-Update Model

We use a Bayesian model which treats the intended referent as a latent variable generating a sequence of observations ($w_{1:n}$ is the sequence of words $w_1, w_2, \ldots, w_n$):
$P(r|w_{1:n}) = \alpha * P(w_n|r, w_{1:n-1}) * P(r|w_{1:n-1})$
where
• $P(w_n|r, w_{1:n-1})$ is the likelihood of the new observation (see below for how we approximate that); and
• the prior $P(r|w_{1:n-1})$ at step n is the posterior of the previous step. Before the first observation is made (i. e., the first word is seen), the prior is simply a distribution over the possible referents, $P(r)$.

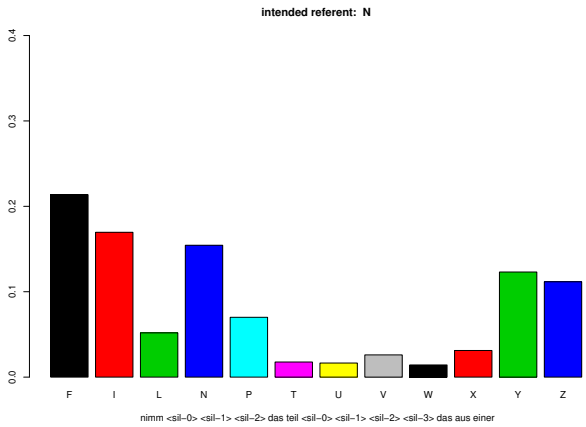


Figure 4: Example of Belief Distribution after Observation

In our experiment, we set this to a uniform distribution, but if there is prior information from other sources (e. g., because the dialogue state makes certain pieces more salient), this can be reflected.
• α is a normalising constant, ensuring that the result is indeed a probability distribution.

The output of the model is a distribution of belief over the 12 available entities, as shown in Figure 4. Figure 5 shows in a 3D plot the development of the belief state (pieces from front to back, strength of belief as height of the peaks) over the course of a whole utterance (with observations from left to right).

5.2 The Decision Step

We implemented several ways to derive a decision for a referent from such a distribution:

i) In the *arg max* approach, at each state the referent with the highest posterior probability is chosen. For Figure 4, that would be F (and hence, a wrong decision). As Figure 5 shows (and the example is quite representative for the model behaviour), there often are various local maxima over the course of an utterance, and hence a model that takes as its decision always the maximum can be expected to perform many edits.

ii) In the *adaptive threshold* approach, we start with a default decision for a special 13th class, "undecided", and a new decision is only made if the maximal value at the current step is above a certain threshold, where this threshold is reset every time this condition is met. In other words, this draws a plane into the belief space and only makes a new decision when a peak rises above this plane and hence above the previous peak. In effect, this approach favours strong convictions and reduces

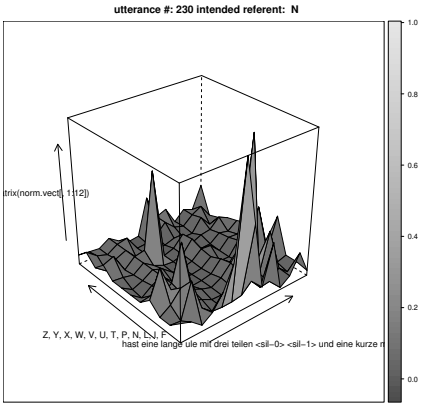


Figure 5: Belief Update over Course of Utterance

the "jitter" in the decision making.

In our example from Figure 4, this would mean that the maximum, F, would only be the decision if its value was higher than the threshold and there was no previous guess that was even higher.

iii) The final model implements a *threshold n-best* approach, where not just a single piece is selected but all pieces that are above a certain threshold. Assuming that the threshold is 0.1 for example this would select F, I, N, Y, and Z—and hence would include the correct reference in Figure 4.

5.3 Implementation

To learn and query the observation likelihoods $P(w_n|r, w_{1:n-1})$, we used referent-specific language models. More precisely, we computed the likelihood as $P(r, w_{1:n})/P(r, w_{1:n-1})$ (definition conditional probability), and approximated the joint probabilities of referent and word sequence via n-grams with specialised words. E. g., an utterance like "take the long, narrow piece" referring to piece I (or tested for reference to this piece) would be rewritten as "take_I the_I long_I narrow_I piece_I" and presented to the n-gram learner / inference component. (Both taken from the SRI LM package, (Stolcke, 2002).)

During evaluation of the models, the test utterances are fed word-by-word to the model and the decision is evaluated against the known intended referent. Since we were interested in testing whether disfluencies contained information that would be learned, for one variant of the system we also fed pseudo-words for silences and hesitation markers like *uhm*, numbered by their position (i. e., "take the .." becomes "take the sil-1 sil-2"), to both learning and inference for the *silence-sensitive* variant; the *silence-ignorant* variant simply repeats the previous decision at such points and does not update its belief state; this way, it is guaranteed that both variants generate the same number of decisions and can be compared directly. (Cf. the dashes in the "no-sil-model" in Figure 1 above: those are points where no real computation is made in the no-sil case.)

5.4 Experiments

All experiments were performed with 10-fold cross-validation. We always ran both versions, the one that showed silences to the model and the one that didn't. We tested various combinations of language model parameters and deciders, of which the best-performing ones are discussed in the next section.

5.5 Results

Table 1 shows the results for the different decision methods and for models where silences are included as observations and where they aren't, and, as a baseline, the result for a resolver that makes a random decision after each observation.

As we can see, the different decision methods have different characteristics wrt. individual measures. The *threshold n-best* approach performs best across the board—but of course has a slightly easier job since it does not need to make unambiguous decisions. We will look into the development of the n-best lists in a second, but for now we note that this model is for almost all utterances correct at least once (97 % **fc applicable**) and if so, typically very early (after 30 % of the utterance). In over half of the cases (54.68 %), the final decision is correct (i. e. is an n-best list that contains the correct referent), and similarly for a good third of all silence observations. Interestingly, silence-correctness is decidedly higher for the silence model (which does actually make new decisions during silences and hence based on the information that the speaker is hesitating) than for the non-sil model (which at these places only repeats the previously made decision). The model performs significantly bettern than a baseline that randomly selects n-best lists of the same size (see *rnd-nb* in Table 1).

As can be expected, the *adaptive threshold* approach is more stable with its decisions, as witnessed by the low **edit overhead**. The fact that it changes its decision not as often has an impact on the other measures, though: in more cases, the model is correct not even once (**fc applicable** is

	n-best		rnd-nb	adapt		max		random
Measure / Model	w/ h	w/o h	w/ h	w/ h	w/o h	w/ h	w/o h	w/ h
fc applicable	97.22 %	95.03 %	85.38 %	63.15 %	66.67 %	86.55 %	82.89 %	59.94 %
average fc	30.43 %	33.73 %	29.61 %	53.87 %	55.25 %	46.55 %	49.31 %	42.60 %
ff applicable	54.68 %	54.24 %	17.54 %	48.68 %	53.07 %	39.77 %	40.64 %	9.65 %
average ff	87.74 %	85.01 %	97.08 %	71.24 %	70.89 %	96.08 %	94.28 %	98.44 %
edit overhead	93.49 %	90.65 %	96.65 %	69.61 %	67.66 %	92.57 %	89.44 %	93.16 %
correctness	37.81 %	36.81 %	23.37 %	23.01 %	26.61 %	17.83 %	20.23 %	7.83 %
sil-correctness	36.60 %	31.09 %	26.39 %	18.71 %	22.58 %	13.67 %	19.34 %	8.63 %
adjusted error	60.07 %	56.96 %	76.63 %	76.29 %	70.90 %	82.17 %	79.42 %	92.16 %

Table 1: Results for different decision methods (*n-best*, *adapt*ive, *max* arg and *random*) and for models with and without silence-observations (*w/ h* and *w/o h*, respectively)

lower than for the other two models). But it is still correct with almost half of its final decisions, and these come even earlier than for the n-best model. Silence information does not seem to help this model; this suggests that the information provided by knowledge about the fact that the speaker hesitates is too subtle to push through the threshold in order to change decisions.

The *arg max* approach fares worst. Since neither the relative strength of the strongest belief (as compared to that in the competing pieces) nor the global strength (have I been more convinced before?) is taken into account, the model changes its mind too often, as evidenced by the edit overhead, and does not settle on the correct referent often (and if, then late). Again, silence information does not seem to be helpful for this model.

As a more detailed look at what happens during silence sequences, Figure 6 plots the average change in probability from onset of silence to a point at 1333 ms of silence. (Recall that the underlying Bayesian model is the same for all models evaluated above, they differ only in how they derive a decision.) We can see that the gains and losses are roughly as expected from the analysis of the corpora: pieces like L and P become more expected after a silence of that length, pieces like X less. So the model does indeed seem to learn that hesitations systematically occur together with certain pieces. (The reader can convince herself with the help of Figure 2 that these shapes are indeed comparatively hard-to-describe; but the interesting point here is that this categorisation does not have to be brought to the model but rather is discovered by it.)

Finally, a look at the distribution and the sizes of the n-best groupings: the most frequent decision is

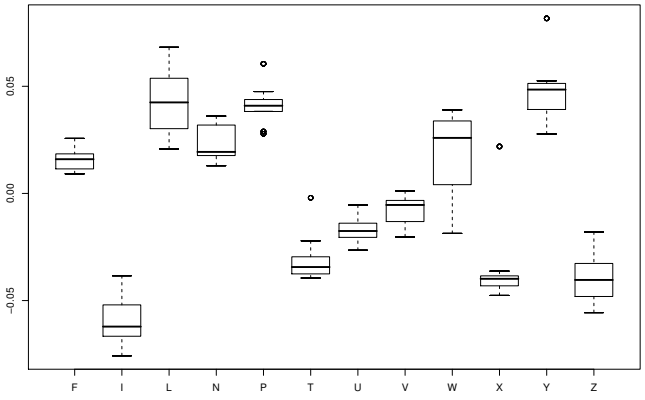


Figure 6: Average change in probability from onset of silence to 1333 ms into silence

"undecided" (474 times), followed by the groupings F_N, N_Y, and N_Y_P (343, 342 and 196, respectively). Here again we find groupings that reflect the differences w.r.t. hesitation rate. The average size of the n-best lists is 2.58 (sd = 1.4).

6 Conclusions and Further Work

We discussed the task of incremental reference resolution (IRR), in particular with respect to exophoric reference. From a theoretical perspective, it might seem easy to specify what the ideal behaviour of an IRR component should be, namely to always produce the set of entities (the extension) that is compatible with the part of the expression seen so far. In practice, however, this is difficult to annotate, for both practical reasons as well as theoretical (referring is a pragmatic activity that is not reducible to denotation). The metrics we defined for evaluation of IRR components account for this in that they do not require a gold

standard annotation that fixes the dynamics of the resolution process; they simply make it possible to quantify the assumption that "early and with strong convictions" is best.

We then presented our probabilistic model of IRR that works directly on word observations without any further processing (POS tagging, parsing). It achieves a reasonable success (as measured with our metrics); for example, in over half of the cases, the final guess of the model is correct, and comes before the utterance is over. As an additional interesting feature, the model is able to interpret hesitations (silences lifted to pseudo-word status) in a way shown before only in controlled psycholinguistic experiments, namely as making reference to hard-to-describe pieces more likely.[3]

In future work, we want to explore the model's performance on ASR output. It is not clear a priori that this would degrade performance much, as it can be expected that the learning components are quite robust against noise. Connected to this, we want to explore more complex statistical models, e. g. a hierarchical model where one level generates parts of the utterance (e. g. non-referential parts and referential parts) and the second the actual words. We also want to test how this approach scales up to worlds with a larger number of possible referents, where consequently approximation methods like particle filtering have to be used. Finally, we will test how the module contributes to a working dialogue system, where further decisions (e. g. for clarification requests) can be built on its output.

Acknowledgments This work was funded by a grant from DFG in the Emmy Noether Programme. We would like to thank the anonymous reviewers for their detailed comments.

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.

Jennifer E. Arnold, Carla L. Hudson Kam, and Michael K. Tanenhaus. 2007. If you say *thee uh* you are describing something hard: The on-line attribution of disfluency during reference comprehension. *Journal of Experimental Psychology*.

Karl Bailey and F. Ferreira. 2007. The processing of filled pause disfluencies in the visual world. In R. P. G. von Gompel, M H. Fischer, W. S. Murray, and R. L. Hill, editors, *Eye Movements: A Window on Mind and Brain*, chapter 22. Elsevier.

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.

Susan E. Brennan and Michael F. Schober. 2001. How listeners compensate for disfluencies in spontaneous speech. *Journal of Memory and Language*, 44:274–296.

Herbert H. Clark and Edward F. Schaefer. 1987. Collaborating on contributions to conversations. *Language and Cognitive Processes*, 2(1):19–41.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive Science*, 19:233–263.

Raquel Fernández, David Schlangen, and Tatjana Lucht. 2007. Push-to-talk ain't always bad! comparing different interactivity settings in task-oriented dialogue. In *Proceeding of DECALOG (SemDial'07)*, Trento, Italy, June.

Carlos Gómez Gallo, Gregory Aist, James Allen, William de Beaumont, Sergio Coria, Whitney Gegg-Harrison, Joana Paulo Pardal, and Mary Swift. 2007. Annotating continuous understanding in a multimodal dialogue corpus. In *Proceeding of DECALOG (SemDial07)*, Trento, Italy, June.

Florian Schiel. 2004. Maus goes iterative. In *Proc. of the IV. International Conference on Language Resources and Evaluation*, Lisbon, Portugal.

Alexander Siebert and David Schlangen. 2008. A simple method for resolution of definite reference in a shared visual context. In *Procs of SIGdial*, Columbus, Ohio.

Gabriel Skantze and David Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of EACL 2009*, Athens, Greece, April.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings Intl. Conf. Spoken Language Processing (ICSLP'02)*, Denver, Colorado, USA, September.

Scott C. Stoness, Joel Tetreault, and James Allen. 2004. Incremental parsing with reference interaction. In *Proceedings of the Workshop on Incremental Parsing at the ACL 2004*, pages 18–25, Barcelona, Spain, July.

Michael K. Tanenhaus, Michael J. Spivey-Knowlton, Kathllen M. Eberhard, and Julie C. Sedivy. 1995. Intergration of visual and linguistic information in spoken language comprehension. *Science*, 268.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. 2005. *Probabilistic Robotics*. MIT Press, Cambridge, Massachusetts, USA.

[3] It is interesting to speculate whether this could have implications for generation of referring expressions as well. It might be a good strategy to make your planning problems observable or even to fake planning problems that are understandable to humans.

Dealing with Interpretation Errors in Tutorial Dialogue

Myroslava O. Dzikovska, Charles B. Callaway, Elaine Farrow, Johanna D. Moore
School of Informatics
University of Edinburgh, Edinburgh, United Kingdom
mdzikovs,ccallawa,efarrow,jmoore@inf.ed.ac.uk

Natalie Steinhauser, Gwendolyn Campbell
Naval Air Warfare Training Systems Division
Orlando, Florida, USA

Abstract

We describe an approach to dealing with interpretation errors in a tutorial dialogue system. Allowing students to provide explanations and generate contentful talk can be helpful for learning, but the language that can be understood by a computer system is limited by the current technology. Techniques for dealing with understanding problems have been developed primarily for spoken dialogue systems in information-seeking domains, and are not always appropriate for tutorial dialogue. We present a classification of interpretation errors and our approach for dealing with them within an implemented tutorial dialogue system.

1 Introduction

Error detection and recovery is a known problem in the spoken dialogue community, with much research devoted to determining the best strategies, and learning how to choose an appropriate strategy from data. Most existing research is focused on dealing with problems in an interaction resulting from speech recognition errors. This focus is justified, since the majority of understanding problems observed in current spoken dialogue systems (SDS) are indeed due to speech recognition errors.

Recovery strategies, therefore, are sometimes devised specifically to target speech recognition problems - for example, asking the user to repeat the utterance, or to speak more softly, which only makes sense if speech recognition is the source of trouble. However, errors can occur at all levels of processing, including parsing, semantic interpretation, intention recognition, etc. As speech recognition improves and more sophisticated systems are developed, strategies for dealing with errors coming from higher (and potentially more complex) levels of processing will have to be developed.

This paper presents a classification of *non-understandings*, defined as the errors where the system fails to arrive at an interpretation of the user's utterance (Bohus and Rudnicky, 2005), and a set of strategies for dealing with them in an implemented tutorial dialogue system. Our system differs from many existing systems in two ways. First, all dialogue is typed. This was done in part to avoid speech recognition issues and allow for more complex language input than would otherwise be possible. But it is also a valid modality for tutoring - there are now many GUI-based tutoring systems in existence, and as distance and online learning have become more popular, students are increasingly familiar with typed dialogue in chat rooms and discussion boards. Second, different genres impose different constraints on the set of applicable recovery strategies - as we discuss in Section 2, certain help strategies developed for task-oriented dialogue systems are not suitable for tutorial dialogue, because tutoring systems should not give away the answer.

We propose a targeted help approach for dealing with interpretation problems in tutorial dialogue by providing help messages that target errors at different points in the pipeline. In our system they are combined with hints as a way to lead the student to an answer that can be understood. While some

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 38–45,
Queen Mary University of London, September 2009.

parts of the system response are specific to tutorial dialogue, the targeted help messages themselves can serve as a starting point for developing appropriate recovery strategies in other systems where errors at higher levels of interpretation are a problem.

The rest of this paper is organized as follows. In Section 2, we motivate the need for error handling strategies in tutorial dialogue. In Section 3 we describe the design of our system. Section 4 discusses a classification of interpretation problems and our targeted help strategy. Section 5 provides a preliminary evaluation based on a set of system tests conducted to date. Finally, we discuss how the approach taken by our system compares to other systems.

2 Background and Motivation

Tutorial dialogue systems aim to improve learning by engaging students in contentful dialogue. There is a mounting body of evidence that dialogue which encourages students to explain their actions (Aleven and Koedinger, 2000), or to generate contentful talk (Purandare and Litman, 2008), results in improved learning. However, the systems' ability to understand student language, and therefore to encourage contentful talk, is limited by the state of current language technology. Moreover, student language may be particularly difficult to interpret since students are often unaware of proper terminology, and may phrase their answers in unexpected ways. For example, a recent error analysis for a domain-independent diagnoser trained on a large corpus showed that a high proportion of errors were due to unexpected paraphrases (Nielsen et al., 2008).

In small domains, domain-specific grammars and lexicons can cover most common phrasings used by students to ensure robust interpretation (Aleven, 2003; Glass, 2000). However, as the size of the domain and the range of possible questions and answers grows, achieving complete coverage becomes more difficult. For essays in large domains, statistical methods can be used to identify problems with the answer (Jordan et al., 2006; Graesser et al., 1999), but these approaches do not perform well on relatively short single-sentence explanations, and such systems often revert to short-answer questions during remediation to ensure robustness.

To the best of our knowledge, none of these tutorial systems use sophisticated error handling techniques. They rely on the small size of the domain or simplicity of expected answers to limit the range of student input. They reject utterances they cannot interpret, asking the user to repeat or rephrase, or tolerate the possibility that interpretation problems will lead to repetitive or confusing feedback.

We are developing a tutorial dialogue system that behaves more like human tutors by supporting open-ended questions, as well as remediations that allow for open-ended answers, and gives students detailed feedback on their answers, similar to what we observed with human tutors. This paper takes the first step towards addressing the problem of handling errors in tutorial dialogue by developing a set of non-understanding recovery strategies - i.e. strategies used where the system cannot find an interpretation for an utterance.

In early pilot experiments we observed that if the system simply rejects a problematic student utterance, saying that it was not understood, then students are unable to determine the reason for this rejection. They either resubmit their answer making only minimal changes, or else they rephrase the sentence in a progressively more complicated fashion, causing even more interpretation errors. Even after interacting with the system for over an hour, our students did not have an accurate picture as to which phrasings are well understood by the system and which should be avoided. Previous research also shows that users are rarely able to perceive the true causes of ASR errors, and tend to form incorrect theories about the types of input a system is able to accept (Karsenty, 2001).

A common approach for dealing with these issues in spoken dialogue systems is to either change to system initiative with short-answer questions ("Is your destination London?"), or provide targeted help ("You can say plane, car or hotel"). Neither of these is suitable for our system. The expected utterances in our system are often more complex (e.g., "The bulb must be in a closed path with the battery"), and therefore suggesting an utterance may be equivalent to giving away the entire answer. Giving students short-answer questions such as "Are the terminals connected or not connected?" is a valid tutoring strategy sometimes used by the tutors. However, it changes the nature of the question from a recall

task to a recognition task, which may affect the student's ability to remember the correct solution independently. Therefore, we decided to implement strategies that give the student information about the nature of the mistake without directly giving information about the expected answer, and encourage them to rephrase their answers in ways that can be understood by the system.

We currently focus on strategies for dealing with non-understanding rather than misunderstanding strategies (i.e. cases where the system finds an interpretation, but an incorrect one). It is less clear in tutorial dialogue what it means for a misunderstanding to be corrected. In task-oriented dialogue, if the system gets a slot value different from what the user intended, it should make immediate corrections at the user's request. In tutoring, however, it is the system which knows the expected correct answer. So if the student gives an answer that does not match the expected answer, when they try to correct it later, it may not always be obvious whether the correction is due to a true misunderstanding, or due to the student arriving at a better understanding of the question. Obviously, true misunderstandings can and will still occur - for example, when the system resolves a pronoun incorrectly. Dealing with such situations is planned as part of future work.

3 System Architecture

Our target application is a system for tutoring basic electricity and electronics. The students read some introductory material, and interact with a simulator where they can build circuits using batteries, bulbs and switches, and measure voltage and current. They are then asked two types of questions: factual questions, like "If the switch is open, will bulb A be on or off?", and explanation questions. The explanation questions ask the student to explain what they observed in a circuit simulation, for example, "Explain why you got the voltage of 1.5 here", or define generic concepts, such as "What is voltage?". The expected answers are fairly short, one or two sentences, but they involve complex linguistic phenomena, including conjunction, negation, relative clauses, anaphora and ellipsis.

The system is connected to a knowledge base which serves as a model for the domain and a reasoning engine. It represents the objects and relationships the system can reason about, and is used to compute answers to factual questions.[1] The student answers are processed using a standard NLP pipeline. All utterances are parsed to obtain syntactic analyses.[2] The lexical-semantic interpreter takes analyses from the parser and maps them to semantic representations using concepts from the domain model. A reference resolution algorithm similar to (Byron, 2002) is used to find referents for named objects such as "bulb A" and for pronouns.

Once an interpretation of a student utterance has been obtained, it is checked in two ways. First, its internal consistency is verified. For example, if the student says "Bulb A will be on because it is in a closed path", we first must ensure that their answer is consistent with what is on the screen - that bulb A is indeed in a closed path. Otherwise the student probably has a problem either with understanding the diagrams or with understanding concepts such as "closed path". These problems indicate lack of basic background knowledge, and need to be remediated using a separate tutorial strategy.

Assuming that the utterance is consistent with the state of the world, the explanation is then checked for correctness. Even though the student utterance may be factually correct (Bulb A is indeed in a closed path), it may still be incomplete or irrelevant. In the example above, the full answer is "Bulb A is in a closed path with the battery", hence the student explanation is factually correct but incomplete, missing the mention of the battery.

In the current version of our system, we are particularly concerned about avoiding misunderstandings, since they can result in misleading tutorial feedback. Consider an example of what can happen if there is a misunderstanding due to a lexical coverage gap. The student sentence "the path is broken" should be interpreted as "the path is no longer closed", corresponding to the `is-open` relation. However, the

[1] Answers to explanation questions are hand-coded by tutors because they are not always required to be logically complete (Dzikovska et al., 2008). However, they are checked for consistency as described later, so they have to be expressed in terms that the knowledge base can reason about.

[2] We are using a deep parser that produces semantic analyses of student's input (Allen et al., 2007). However, these have to undergo further lexical interpretation, so we are treating them as syntactic analyses for purposes of this paper.

most frequent sense of "broken" is `is-damaged`, as in "the bulb is broken". Ideally, the system lexicon would define "broken" as ambiguous between those two senses. If only the "damaged" sense is defined, the system will arrive at an incorrect interpretation (misunderstanding), which is false by definition, as the `is-damaged` relation applies only to bulbs in our domain. Thus the system will say "you said that the path is damaged, but that's not true". Since the students who used this phrasing were unaware of the proper terminology in the first instance, they dismissed such feedback as a system error. A more helpful feedback message is to say that the system does not know about damaged paths, and the sentence needs to be rephrased.[3]

Obviously, frequent non-understanding messages can also lead to communication breakdowns and impair tutoring. Thus we aim to balance the need to avoid misunderstandings with the need to avoid student frustration due to a large number of sentences which are not understood. We approach this by using robust parsing and interpretation tools, but balancing them with a set of checks that indicate potential problems. These include checking that the student answer fits with the sortal constraints encoded in the domain model, that it can be interpreted unambiguously, and that pronouns can be resolved.

4 Error Handling Policies

All interpretation problems in our system are handled with a unified tutorial policy. Each message to the user consists of three parts: a social response, the explanation of the problem, and the tutorial response. The social response is currently a simple apology, as in "I'm sorry, I'm having trouble understanding." Research on spoken dialogue shows that users are less frustrated if systems apologize for errors (Bulyko et al., 2005).

The explanation of the problem depends on the problem itself, and is discussed in more detail below.

The tutorial response depends on the general tutorial situation. If this is the first misunderstanding, the student will be asked to rephrase/try again. If they continue to phrase things in a way that is misunderstood, they will be given up to two different hints (a less specific hint followed by a more specific hint); and finally the system will bottom out with a correct answer. Correct answers produced by the generator are guaranteed to be parsed and understood by the interpretation module, so they can serve as templates for future student answers.

The tutorial policy is also adjusted depending on the interaction history. For example, if a non-understanding comes after a few incorrect answers, the system may decide to bottom out immediately in order to avoid student frustration due to multiple errors. At present we are using a heuristic policy based on the total number of incorrect or uninterpretable answers. In the future, such policy could be learned from data, using, for example, reinforcement learning (Williams and Young, 2007).

In the rest of this section we discuss the explanations used for different problems. For brevity, we omit the tutorial response from our examples.

4.1 Parse Failures

An utterance that cannot be parsed represents the worst possible outcome for the system, since detecting the reason for a syntactic parse failure isn't possible for complex parsers and grammars. Thus, in this instance the system does not give any description of the problem at all, saying simply "I'm sorry, I didn't understand."

Since we are unable to explain the source of the problem, we try hard to avoid such failures. We use a spelling corrector and a robust parser that outputs a set of fragments covering the student's input when a full parse cannot be found. The downstream components are designed to merge interpretations of the fragments into a single representation that is sent to the reasoning components.

Our policy is to allow the system to use such fragmentary parses when handling explanation questions, where students tend to use complex language. However, we require full parses for factual questions, such as "Which bulbs will be off?" We found that for those simpler questions students are able to easily phrase an acceptable answer, and the lack of a full parse signals some unusually complex language that downstream components are likely to have problems with as well.

[3] This was a real coverage problem we encountered early on. While we extended the coverage of the lexical interpreter based on corpus data, other gaps in coverage may remain. We discuss the issues related to the treatment of vague or incorrect terminology in Section 4.

One risk associated with using fragmentary parses is that relationships between objects from different fragments would be missed by the parser. Our current policy is to confirm the correct part of the student's answer, and prompt for the missing parts, e.g., " Right. The battery is contained in a closed path. And then?" We can do this because we use a diagnoser that explicitly identifies the correct objects and relationships in the answer (Dzikovska et al., 2008), and we are using a deep generation system that can take those relationships and automatically generate a rephrasing of the correct portion of the content.

4.2 Lexical Interpretation Errors

Errors in lexical interpretation typically come from three main sources: unknown words which the lexical interpreter cannot map into domain concepts, unexpected word combinations, and incorrect uses of terminology that violate the sortal constraints encoded in the domain model.

Unknown words are the simplest to deal with in the context of our lexical interpretation policy. We do not require that every single word of an utterance should be interpreted, because we want the system to be able to skip over irrelevant asides. However, we require that if a predicate is interpreted, all its arguments should be interpreted as well. To illustrate, in our system the interpretation of "the bulb is still lit" is `(LightBulb Bulb-1-1) (is-lit Bulb-1-1 true)`. The adverbial "still" is not interpreted because the system is unable to reason about time.[4] But since all arguments of the `is-lit` predicate are defined, we consider the interpretation complete.

In contrast, in the sentence "voltage is the measurement of the power available in a battery", "measurement" is known to the system. Thus, its argument "power" should also be interpreted. However, the reading material in the lessons never talks about power (the expected answer is "Voltage is a measurement of the difference in electrical states between two terminals"). Therefore the unknown word detector marks "power" as an unknown word, and tells the student "I'm sorry, I'm having a problem understanding. I don't know the word *power*."

[4]The lexical interpretation algorithm makes sure that frequency and negation adverbs are accounted for.

The system can still have trouble interpreting sentences with words which are known to the lexical interpreter, but which appear in unexpected combinations. This involves two possible scenarios. First, unambiguous words could be used in a way that contradicts the system's domain model. For example, the students often mention "closed circuit" instead of the correct term "closed path". The former is valid in colloquial usage, but is not well defined for parallel circuits which can contain many different paths, and therefore cannot be represented in a consistent knowledge base. Thus, the system consults its knowledge base to tell the student about the appropriate arguments for a relation with which the failure occurred. In this instance, the feedback will be "I'm sorry, I'm having a problem understanding. I don't understand it when you say that circuits can be closed. Only paths and switches can be closed."[5]

The second case arises when a highly ambiguous word is used in an unexpected combination. The knowledge base uses a number of fine-grained relations, and therefore some words can map to a large number of relations. For example, the word "has" means `circuit-component` in "The circuit has 2 bulbs", `terminals-of` in "The bulb has terminals" and `voltage-property` in "The battery has voltage". The last relation only applies to batteries, but not to other components. These distinctions are common for knowledge representation and reasoning systems, since they improve reasoning efficiency, but this adds to the difficulty of lexical interpretation. If a student says "Bulb A has a voltage of 0.5", we cannot determine the concept to which the word "has" corresponds. It could be either `terminals-of` or `voltage-property`, since each of those relations uses one possible argument from the student's utterance. Thus, we cannot suggest appropriate argument types and instead we indicate the problematic word combination, for example, "I'm sorry, I'm having trouble understanding. I didn't understand *bulb has voltage*."

Finally, certain syntactic constructions involving comparatives or ellipsis are known to be difficult

[5]Note that these error messages are based strictly on the fact that sortal constraints from the knowledge base for the relation that the student used were violated. In the future, we may also want to adjust the recovery strategy depending on whether the problematic relation is relevant to the expected answer.

open problems for interpretation. While we are working on interpretation algorithms to be included in future system versions, the system currently detects these special relations, and produces a message telling the student to rephrase without the problematic construction, e.g., "I'm sorry. I'm having a problem understanding. I do not understand *same as*. Please try rephrasing without the word *as*."

4.3 Reference Errors

Reference errors arise when a student uses an ambiguous pronoun, and the system cannot find a suitable object in the knowledge base to match, or on certain occasions when an attachment error in a parse causes an incorrect interpretation. We use a generic message that indicates the type of the object the system perceived, and the actual word used, for example, "I'm sorry. I don't know which switch you're referring to with *it*."

To some extent, reference errors are instances of misunderstandings rather than non-understandings. There are actually 2 underlying cases for reference failure: either the system cannot find any referent at all, or it is finding too many referents. In the future a better policy would be to ask the student which of the ambiguous referents was intended. We expect to pilot this policy in one of our future system tests.

5 Evaluation

So far, we have run 13 pilot sessions with our system. Each pilot consisted of a student going through 1 or 2 lessons with the system. Each lesson lasts about 2 hours and has 100-150 student utterances (additional time is taken with building circuits and reading material). Both the coverage of the interpretation component and the specificity of error messages were improved between each set of pilots, thus it does not make sense to aggregate the data from them. However, over time we observed the trend that students are more likely to change their behavior when the system issues more specific messages.

Examples of successful and unsuccessful interactions are shown in Figure 1. In (a), the student used incorrect terminology, and a reminder about how the word "complete" is interpreted was enough to get the conversation back on track.

The dialogue fragment in (b) shows how messages which are not specific enough can cause a breakdown in conversation. The system used an insufficiently specific message at the beginning (omitting the part that says that only switches and paths can be closed). This led the student away from an answer which was nearly correct with slightly imperfect terminology to an answer which was insufficient (it's not enough for the components to be connected, they have to be in a closed path), and then to rephrase it in a more complicated way that was impossible for the system to understand (consistent with findings of Bulyko et al. (2005)).

The next step would be to conduct a formal evaluation of our policy. We are planning to do this using recovery rate after different strategies as our evaluation measure (Bohus and Rudnicky, 2005), which is the percentage of times when the utterance immediately following a non-understanding was successfully interpreted by the system.

6 Discussion

Our error handling policy attempts to balance robustness in interpretation with the need for precise tutorial feedback. Human tutors accept a wide range of variation in terminology and phrasing. However, there is a known trade-off between non-understandings and misunderstandings. We decided that misunderstandings are likely to cause more serious problems with respect to tutoring. Therefore we set strict understanding requirements and focused on developing a policy that would encourage students to use constructions that the system can interpret reliably. This is also supported by our observations in the human-human data – tutors believe that teaching correct terminology is important, and we observed them saying things like "It's good to put it into your own words like that. Just for the sake of the lesson, could you also use the term *closed path*?".

Our approach is similar to the TargetedHelp approach of (Hockey et al., 2003), where in the case of an interpretation failure the task assistant produces a report of what the recognizer heard, a description of the problem, and a possible in-domain utterance as an alternative. In a user evaluation such targeted help messages resulted in significantly lower task completion times compared to a system that produced a generic "I didn't understand" message, and

(a)

S: the open switch makes the path incomplete
T: I'm sorry, I'm having a problem understanding. Paths can't be complete. Only circuits can be complete. Please try again.
S: the open switch makes the path open
T: Right. An open switch creates a gap.

(b)

T: What are the conditions that are required to make a bulb light up?
S: The lightbulb has to be in a closed circuit with a battery
T: I'm sorry, I'm having a problem understanding. I don't understand when you say that circuits can be closed. Please try again.
S: The lightbulb has to be connected to a battery
T: Yes. You correctly mentioned a battery and a bulb. Keep trying.
S: Both the positive and negative connectors have to be connected to
T: I'm sorry, I'm having a problem understanding. Try rephrasing your answer.
S: The lightbulb has to be connected to the positive and negative connectors of the battery
T: I'm sorry, I didn't understand. Here's the answer. The battery and the bulb must be contained in the same closed path.

Figure 1: Sample student-system interactions. **S:** is student, **T:** is tutor (system). (a) A successful interaction where the student changes their language; (b) a failed interaction where system feedback was unhelpful.

subjects gradually learned how to talk to the system, reducing the number of misunderstandings over time. This gives us reason to believe that our system can achieve similar effects in tutorial dialogue. While we don't suggest alternative domain utterances due to the tutoring reasons described earlier, the progressively more specific hints serve a similar function. To what extent this impacts learning and interaction with the system will have to be determined in future evaluations.

The error handling in our system is significantly different from systems that analyze user essays because it needs to focus on a single sentence at a time. In a system that does essay analysis, such as AUTOTUTOR (Graesser et al., 1999) or Why2-Atlas (Jordan et al., 2006) a single essay can have many flaws. So it doesn't matter if some sentences are not fully understood as long as the essay is understood well enough to identify at least one flaw. Then that particular flaw can be remediated, and the student can resubmit the essay. However, this can also cause student frustration and potentially affect learning if the student is asked to re-write an essay many times due to interpretation errors.

Previous systems in the circuit domain focused on troubleshooting rather than conceptual knowledge. The SHERLOCK tutor (Katz et al., 1998) used only menu-based input, limiting possible dialogue. Circuit Fix-It Shop (Smith and Gordon, 1997) was a task-oriented system which allowed for speech input, but with very limited vocabulary. Our system's larger vocabulary and complex input result in different types of non-understandings that cannot be resolved with simple confirmation messages.

A number of researchers have developed error taxonomies for spoken dialogue systems (Paek, 2003; Möller et al., 2007). Our classification does not have speech recognition errors (since we are using typed dialogue), and we have a more complex interpretation stack than the domain-specific parsing utilized by many SDSs. However, some types of errors are shared, in particular, our "no parse", "unknown word" and "unknown attachment" errors correspond to *command-level errors*, and our sortal constraint and reference errors correspond to *concept-level errors* in the taxonomy of Möller et al. (2007). This correspondence is not perfect because of the nature of the task - there are no commands in a tutoring system. However, the underlying causes are very similar, and so research on the best way

to communicate about system failures would benefit both tutoring and task-oriented dialogue systems. In the long run, we would like to reconcile these different taxonomies, leading to a unified classification of system errors and recovery strategies.

7 Conclusion

In this paper we described our approach to handling non-understanding errors in a tutorial dialogue system. Explaining the source of errors, without giving away the full answer, is crucial to establishing effective communication between the system and the student. We described a classification of common problems and our approach to dealing with different classes of errors. Our experience with pilot studies, as well as evidence from spoken dialogue systems, indicates that our approach can help improve dialogue efficiency. We will be evaluating its impact on both student learning and on dialogue efficiency in the future.

8 Acknowledgments

This work has been supported in part by Office of Naval Research grant N000140810043.

References

V. A. Aleven and K. R. Koedinger. 2000. The need for tutorial dialog to support self-explanation. In *Proc. of AAAI Fall Symposion on Building Dialogue Systems for Tutorial Applications*.

O. P. V. Aleven. 2003. A knowledge-based approach to understanding students' explanations. In *School of Information Technologies, University of Sydney*.

J. Allen, M. Dzikovska, M. Manshadi, and M. Swift. 2007. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the ACL-07 Workshop on Deep Linguistic Processing*.

D. Bohus and A. Rudnicky. 2005. Sorry, i didn't catch that! - an investigation of non-understanding errors and recovery strategies. In *Proceedings of SIGdial-2005*, Lisbon, Portugal.

I. Bulyko, K. Kirchhoff, M. Ostendorf, and J. Goldberg. 2005. Error-correction detection and response generation in a spoken dialogue system. *Speech Communication*, 45(3):271–288.

D. K. Byron. 2002. *Resolving Pronominal Reference to Abstract Entities*. Ph.D. thesis, University of Rochester.

M. O. Dzikovska, G. E. Campbell, C. B. Callaway, N. B. Steinhauser, E. Farrow, J. D. Moore, L. A. Butler, and C. Matheson. 2008. Diagnosing natural language answers to support adaptive tutoring. In *Proceedings 21st International FLAIRS Conference*.

M. Glass. 2000. Processing language input in the CIRCSIM-Tutor intelligent tutoring system. In *Proc. of the AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*.

A. C. Graesser, P. Wiemer-Hastings, P. Wiemer-Hastings, and R. Kreuz. 1999. Autotutor: A simulation of a human tutor. *Cognitive Systems Research*, 1:35–51.

B. A. Hockey, O. Lemon, E. Campana, L. Hiatt, G. Aist, J. Hieronymus, A. Gruenstein, and J. Dowding. 2003. Targeted help for spoken dialogue systems: intelligent feedback improves naive users' performance. In *Proceedings of EACL*.

P. Jordan, M. Makatchev, U. Pappuswamy, K. VanLehn, and P. Albacete. 2006. A natural language tutorial dialogue system for physics. In *Proceedings of FLAIRS'06*.

L. Karsenty. 2001. Adapting verbal protocol methods to investigate speech systems use. *Applied Ergonomics*, 32:15–22.

S. Katz, A. Lesgold, E. Hughes, D. Peters, G. Eggan, M. Gordin, and L. Greenberg. 1998. Sherlock 2: An intelligent tutoring system built on the lrdc framework. In C. Bloom and R. Loftin, editors, *Facilitating the development and use of interactive learning environments*. ERLBAUM.

S. Möller, K.-P. Engelbrecht, and A. Oulasvirta. 2007. Analysis of communication failures for spoken dialogue systems. In *Proceedings of Interspeech*.

R. D. Nielsen, W. Ward, and J. H. Martin. 2008. Classification errors in a domain-independent assessment system. In *Proc. of the Third Workshop on Innovative Use of NLP for Building Educational Applications*.

T. Paek. 2003. Toward a taxonomy of communication errors. In *Proceedings of ISCA Workshop on Error Handling in Spoken Dialogue Systems*.

A. Purandare and D. Litman. 2008. Content-learning correlations in spoken tutoring dialogs at word, turn and discourse levels. In *Proc.of FLAIRS*.

R. W. Smith and S. A. Gordon. 1997. Effects of variable initiative on linguistic behavior in human-computer spoken natural language dialogue. *Computational Linguistics*.

J. D. Williams and S. Young. 2007. Scaling POMDPs for spoken dialog management. *IEEE Trans. on Audio, Speech, and Language Processing*, 15(7):2116–2129.

Towards the Interpretation of Utterance Sequences in a Dialogue System

Ingrid Zukerman and Patrick Ye and Kapil Kumar Gupta and Enes Makalic
Faculty of Information Technology
Monash University
Clayton, VICTORIA 3800, Australia
ingrid@infotech.monash.edu.au, {ye.patrick,kapil.k.gupta,emakalic}@gmail.com

Abstract

This paper describes a probabilistic mechanism for the interpretation of sentence sequences developed for a spoken dialogue system mounted on a robotic agent. The mechanism receives as input a sequence of sentences, and produces an interpretation which integrates the interpretations of individual sentences. For our evaluation, we collected a corpus of hypothetical requests to a robot. Our mechanism exhibits good performance for sentence pairs, but requires further improvements for sentence sequences.

1 Introduction

DORIS (Dialogue Oriented Roaming Interactive System) is a spoken dialogue system under development, which will eventually be mounted on a household robot. The focus of our current work is on *DORIS*'s language interpretation module called *Scusi?*. In this paper, we consider the interpretation of a sequence of sentences.

People often utter several separate sentences to convey their wishes, rather than producing a single sentence that contains all the relevant information (Zweig et al., 2008). For instance, people are likely to say "Go to my office. Get my mug. It is on the table.", instead of "Get my mug on the table in my office". This observation, which was validated in our corpus study (Section 4), motivates the mechanism for the interpretation of a sequence of sentences presented in this paper. Our mechanism extends our probabilistic process for interpreting single spoken utterances (Zukerman et al., 2008) in that (1) it determines which sentences in a sequence are related, and if so, combines them into an integrated interpretation; and (2) it provides a formulation for estimating the probability of an interpretation of a sentence sequence, which supports the selection of the most probable interpretation. Our evaluation demonstrates that our mechanism performs well in understanding textual sentence pairs of different length and level of complexity, and highlights particular aspects of our algorithms that require further improvements (Section 4).

In the next section, we describe our mechanism for interpreting a sentence sequence. In Section 3, we present our formalism for assessing the probability of an interpretation. The performance of our system is evaluated in Section 4, followed by related research and concluding remarks.

2 Interpreting a Sequence of Utterances

Scusi? employs an anytime algorithm to interpret a sequence of sentences (**Algorithm 1**). The algorithm generates interpretations until time runs out (in our case, until a certain number of iterations has been executed). In Steps 1–5, Algorithm 1 processes each sentence separately according to the interpretation process for single sentences described in (Zukerman et al., 2008).[1] Charniak's probabilistic parser[2] is applied to generate parse trees for each sentence in the sequence. The parser produces up to N ($= 50$) parse trees for each sentence, associating each parse tree with a probability. The parse trees for each sentence are then iteratively considered in descending order of probability, and algorithmically mapped into *Uninstantiated Concept Graphs (UCGs)* — a representa-

[1] Although *DORIS* is a spoken dialogue system, our current results pertain to textual input only. Hence, we omit the aspects of our work pertaining to spoken input.

[2] ftp://ftp.cs.brown.edu/pub/nlparser/

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 46–53,
Queen Mary University of London, September 2009.

Algorithm 1 Interpret a sentence sequence

Require: Sentences $T_1, \ldots, T_n$

{ **Interpret Sentences** }

1: **for all** sentences T_i **do**
2: Generate parse trees $\{P_i\}$, and UCGs $\{U_i\}$
3: Generate candidate modes $\{M_i\}$
4: For each identifier j in T_i, generate candidate referents $\{R_{ij}\}$
5: **end for**

{ **Combine UCGs** }

6: **while** there is time **do**
7: Get $\{(U_1, M_1, R_1), \ldots, (U_n, M_n, R_n)\}$ — a sequence of tuples (one tuple per sentence)
8: Generate $\{U^D\}$, a sequence of declarative UCGs, by merging the declarative UCGs in $\{(U_i, M_i, R_i)\}$ as specified by their identifier-referent pairs and modes
9: Generate $\{U^I\}$, a sequence of imperative UCGs, by merging each imperative UCG in $\{(U_i, M_i, R_i)\}$ with declarative UCGs as specified by their identifier-referent pairs and modes
10: Generate candidate ICG sequences $\{I^I_j\}$ for the sequence $\{U^I\}$
11: Select the best sequence of ICGs $\{I^{I*}\}$
12: **end while**

tion based on Concept Graphs (Sowa, 1984) — one parse tree yielding one UCG (but several parse trees may produce the same UCG). UCGs represent syntactic information, where the concepts correspond to the words in the parent parse tree, and the relations are derived from syntactic information in the parse tree and prepositions (Figure 1(a) illustrates UCGs U^D and U^I generated from the sentences "The mug is on the table. Clean it.").

Our algorithm requires sentence mode (declarative, imperative or interrogative[3]), and resolved references to determine how to combine the sentences in a sequence. Sentence mode is obtained using a classifier trained on part of our corpus (Section 2.2). The probability distribution for the referents of each identifier is obtained from the corpus and from rules derived from (Lappin and Leass, 1994; Ng et al., 2005) (Section 2.3).

At this point, for each sentence T_i in a sequence, we have a list of UCGs, a list of modes, and lists

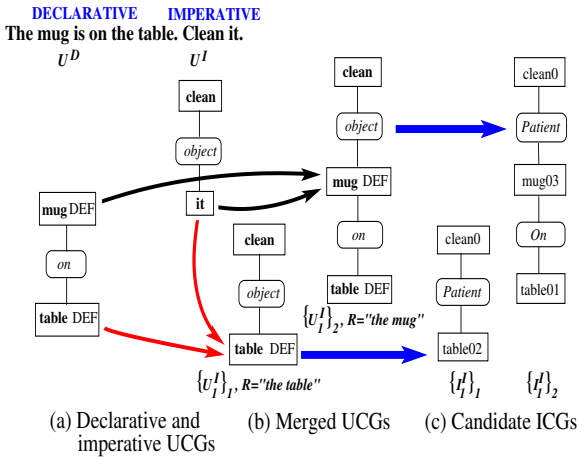


Figure 1: Combining two sentences

of referents (one list for each identifier in the sentence). In Step 7, Algorithm 1 generates a tuple (U_i, M_i, R_i) for each sentence T_i by selecting from these lists a UCG, a mode and a referent for each identifier (yielding a list of identifier-referent pairs). Each element in each (U, M, R) tuple is iteratively selected by traversing the appropriate list in descending order of probability. For instance, given sentences T_1, T_2, T_3, the top UCG for T_1 is picked first, together with the top mode and the top identifier-referent pairs for that sentence (likewise for T_2 and T_3); next the second-top UCG is chosen for T_1, but the other elements remain the same; and so on.

Once the (U, M, R) tuples have been determined, the UCGs for the declarative sentences are merged in the order they were given (Step 8). This is done by first merging a pair of declarative UCGs, then merging the resultant UCG with the next declarative UCG, and so on. The idea is that if the declarative sentences have co-referents, then the information about these co-referents can be combined into one representation. For example, consider the sequence "The mug is on the table. It is blue. Find it. The mug is near the phone. Bring it to me." Some of the UCG sequences obtained from the declarative sentences (first, second and fourth) are:

$\{U^D_1\}_1$={mug(*CLR* blue)-(*on*-table & *near*-phone)}

$\{U^D_1\}_2$={mug-(*on*-table(*CLR* blue) & *near*-phone)}

$\{U^D_1, U^D_2\}_3$={mug(*CLR* blue)-*on*-table, mug-*near*-phone}.[4]

[3] Interrogatives are treated as imperatives at present, so in the remainder of the paper we do not mention interrogatives.

[4] The different notations are because colour (and size) are properties of objects, while prepositions indicate relations.

The first two sequences contain one declarative merged UCG, and the third contains two UCGs.

In Step 9, Algorithm 1 considers a UCG for each imperative sentence in turn, and merges it with declarative UCGs (which may have resulted from a merger), as specified by the modes and identifier-referent pairs of the sentences in question. For example, consider the sentence sequence "Find my mug. It is in my office. Bring it." One of the (U, M, R)-tuple sequences for this instruction set is

{(find-*obj*-mug-*owner*-me, imperative, NIL),
(it1-*in*-office-*owner*-me, declarative, it1-mug),
(bring-*obj*-it2, imperative, it2-mug)}.

After merging the first two UCGs (imperative-declarative), and then the second and third UCGs (declarative-imperative), we obtain the imperative UCG sequence $\{U_1^I, U_2^I\}$:

U_1^I=find-*obj*-mug-(*owner*-me &
in-office-*owner*-me)

U_2^I=bring-*obj*-mug-(*in*-office-*owner*-me).

This process enables *Scusi?* to iteratively merge ever-expanding UCGs with subsequent UCGs, eventually yielding UCG sequences which contain detailed UCGs that specify an action or object. A limitation of this merging process is that the information about the objects specified in an imperative UCG is not aggregated with the information about these objects in other imperative UCGs, and this sometimes can cause the merged imperative UCGs to be under-specified. This limitation will be addressed in the immediate future.

After a sequence of imperative UCGs has been generated, candidate *Instantiated Concept Graphs (ICGs)* are proposed for each imperative UCG, and the most probable ICG sequence is selected (Steps 10–11 of Algorithm 1). We focus on imperative UCGs because they contain the actions that the robot is required to perform; these actions incorporate relevant information from declarative UCGs. ICGs are generated by nominating different instantiated concepts and relations from the system's knowledge base as potential realizations for each concept and relation in a UCG (Zukerman et al., 2008); each UCG can generate many ICGs. Since this paper focuses on the generation of UCG sequences, the generation of ICGs will not be discussed further.

2.1 Merging UCGs

Given tuples (U_i, M_i, R_i) and (U_j, M_j, R_j) where $j > i$, pronouns and one-anaphora in U_j are replaced with their referent in U_i on the basis of the set of identifier-referent pairs in R_j (if there is no referent in U_i for an identifier in U_j, the identifier is left untouched). U_i and U_j are then merged into a UCG U_m by first finding a node n that is common to U_i and U_j, and then copying the sub-tree of U_j whose root is n into a copy of U_i. If more than one node can be merged, the node (head noun) that is highest in the U_j structure is used. If one UCG is declarative and the other imperative, we swap them if necessary, so that U_i is imperative and U_j declarative.

For instance, given the sentences "The mug is on the table. Clean it." in Figure 1, Step 4 of Algorithm 1 produces the identifier-referent pairs {(it, mug), (it, table)}, yielding two intermediate UCGs for the imperative sentence: (1) clean-*object*-mug, and (2) clean-*object*-table. The first UCG is merged with a UCG for the declarative sentence using `mug` as root node, and the second UCG is merged using `table` as root node. This results in merged UCG sequences (of length 1) corresponding to "Clean the table" and "Clean the mug on the table" ($\{U_1^I\}_1$ and $\{U_1^I\}_2$ respectively in Figure 1(b), which in turn produce ICG sequences $\{I_1^I\}_1$ and $\{I_1^I\}_2$ in Figure 1(c), among others).

2.2 Determining modes

We use the MaxEnt classifier[5] to determine the mode of a sentence. The input features to the classifier (obtained from the highest probability parse tree for this sentence) are: (1) top parse-tree node; (2) position and type of the top level phrases under the top parse-tree node, e.g., (0, NP), (1, VP), (2, PP); (3) top phrases under the top parse-tree node reduced to a regular expression, e.g., VP-NP$^+$ to represent, say, VP NP NP; (4) top VP head – the head word of the first top level VP; (5) top NP head – the head word of the first top level NP; (6) first three tokens in the sentence; and (7) last token in the sentence. Using leave-one-out cross validation, this classifier has an accuracy of 97.8% on the test data — a 30% improvement over the majority class (imperative) baseline.

2.3 Resolving references

Scusi? handles pronouns, one-anaphora and NP identifiers (e.g., "the book"). At present, we consider only precise matches between NP identifiers

[5] `http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`

and referents, e.g., "the cup" does not match "the dish". In the future, we will incorporate similarity scores based on WordNet, e.g., Leacock and Chodorow's (1998) scores for approximate lexical matches; such matches occurred in 4% of our corpus (Section 4).

To reduce the complexity of reference resolution across a sequence of sentences, and the amount of data required to reliably estimate probabilities (Section 3), we separate our problem into two parts: (1) identifying the sentence being referred to, and (2) determining the referent within that sentence.

Identifying a sentence. Most referents in our corpus appear in the *current, previous* or *first* sentence in a sequence, with a few referents appearing in *other* sentences (Section 4). Hence, we have chosen the sentence classes {*current, previous, first, other*}. The probability of referring to a sentence of a particular class from a sentence in position i is estimated from our corpus, where $i = 1, \ldots, 5, > 5$ (there are only 13 sequences with more than 5 sentences). We estimate this distribution for each leave-one-out cross-validation fold in our evaluation (Section 4).

Determining a referent. We use heuristics based on those described in (Lappin and Leass, 1994) to classify pronouns (an example of a non-pronoun usage is "*It* is ModalAdjective that S"), and heuristics based on the results obtained in (Ng et al., 2005) to classify one-anaphora (an example of a high-performing feature pattern is "*one* as head-noun with NN or CD as Part-of-speech and no attached *of* PP"). If a term is classified as a pronoun or one-anaphor, then a list of potential referents is constructed using the head nouns in the target sentence. We use the values in (Lappin and Leass, 1994) to assign a score to each anaphor-referent pair according to the grammatical role of the referent in the target UCG (obtained from the highest probability parse tree that is a parent of this UCG). These scores are then converted to probabilities using a linear mapping function.

3 Estimating the Probability of a Merged Interpretation

We now present our formulation for estimating the probability of a sequence of UCGs, which supports the selection of the most probable sequence.

One sentence. The probability of a UCG generated from a sentence T is estimated as described in (Zukerman et al., 2008), resulting in

$$\Pr(U|T) \propto \textstyle\sum_P \Pr(P|T) \cdot \Pr(U|P) \qquad (1)$$

where T, P and U denote text, parse tree and UCG respectively. The summation is taken over all possible parse trees from the text to the UCG, because a UCG can have more than one ancestor. As mentioned above, the parser returns an estimate of $\Pr(P|T)$; and $\Pr(U|P) = 1$, since the process of generating a UCG from a parse tree is deterministic.

A sentence sequence. The probability of an interpretation of a sequence of sentences $T_1, \ldots, T_n$ is

$$\Pr(U_1, \ldots, U_m | T_1, \ldots, T_n) = \Pr(U_1, \ldots, U_n, M_1, \ldots, M_n, R_1, \ldots, R_n | T_1, \ldots, T_n)$$

where m is the number of UCGs in a merged sequence.

By making judicious conditional independence assumptions, and incorporating parse trees into the formulation, we obtain

$$\Pr(U_1, \ldots, U_m | T_1, \ldots, T_n) = \prod_{i=1}^{n} \Pr(U_i|T_i) \cdot \Pr(M_i|P_i, T_i) \cdot \Pr(R_i|P_1, \ldots, P_i)$$

This formulation is independent of the number of UCGs in a merged sequence generated by Algorithm 1, thereby supporting the comparison of UCG sequences of different lengths (produced when different numbers of mergers are performed).

$\Pr(U_i|T_i)$ is calculated using Equation 1, and $\Pr(M_i|P_i, T_i)$ is obtained as described in Section 2.2 (recall that the input features to the classifier depend on the parse tree and the sentence). In principle, $\Pr(M_i|P_i, T_i)$ and $\Pr(R_i|P_1, \ldots, P_i)$ could be obtained by summing over all parse trees, as done in Equation 1. However, at present we use the highest-probability parse tree to simplify our calculations.

To estimate $\Pr(R_i|P_1, \ldots, P_i)$ we assume conditional independence between the identifiers in a sentence, yielding

$$\Pr(R_i|P_1, \ldots, P_i) = \prod_{j=1}^{k_i} \Pr(R_{ij}|P_1, \ldots, P_i)$$

where k_i is the number of identifiers in sentence i, and R_{ij} is the referent for identifier j in sentence i. As mentioned in Section 2.3, this factor is

separated into determining a sentence, and determining a referent in that sentence. We also include in our formulation the Type of the identifier (pronoun, one-anaphor or NP) and sentence position i, yielding

$$\Pr(R_{ij}|P_1,\ldots,P_i) =$$
$$\Pr(R_{ij} \text{ ref } NP_a \text{ in sent } b, \text{Type}(R_{ij})|i, P_1,\ldots,P_i)$$

After additional conditionalization we obtain

$$\Pr(R_{ij}|P_1,\ldots,P_i) =$$
$$\Pr(R_{ij} \text{ ref } NP_a|R_{ij} \text{ ref sent } b, \text{Type}(R_{ij}), P_i, P_b) \times$$
$$\Pr(R_{ij} \text{ ref sent } b|\text{Type}(R_{ij}), i) \times \Pr(\text{Type}(R_{ij})|P_i)$$

As seen in Section 2.3, $\Pr(\text{Type}(R_{ij})|P_i)$ and $\Pr(R_{ij} \text{ ref } NP_a|R_{ij} \text{ ref sent } b, \text{Type}(R_{ij}), P_i, P_b)$ are estimated in a rule-based manner, and $\Pr(R_{ij} \text{ ref sent } b|\text{Type}(R_{ij}), i)$ is estimated from the corpus (recall that we distinguish between sentence classes, rather than specific sentences).

4 Evaluation

We first describe our experimental set-up, followed by our results.

4.1 Experimental set-up

We conducted a web-based survey to collect a corpus comprising multi-sentence requests. To this effect, we presented participants with a scenario where they are in a meeting room, and they ask a robot to fetch something from their office. The idea is that if people cannot see a scene, their instructions will be more segmented than if they can view the scene. The participants were free to decide which object to fetch, and what was in the office. There were no restrictions on vocabulary or grammatical form for the requests.

We collected 115 sets of instructions mostly from different participants (a few people did the survey more than once).[6] The sentence sequences in our corpus contain between 1 and 9 sentences, with 74% of the sequences comprising 1 to 3 sentences. Many of the sentences had grammatical requirements which exceeded the capabilities of our system. To be able to use these instruction sets in our evaluation, we made systematic manual changes to produce sentences that meet our system's grammatical restrictions (in the future, we will relax these restrictions, as required by a deployable system). Below are the main types of changes we made.

- Indirect Speech Acts in the form of questions were changed to imperatives. For instance, "Can you get my tea?" was changed to "Get my tea".
- Conjoined verb phrases or sentences were separated into individual sentences.
- Composite verbs were simplified, e.g., "I think I left it on" was changed to "it is on", and out-of-vocabulary composite nouns were replaced by simple nouns or adjectives, e.g., "the diary is A4 size" to "the diary is big".
- Conditional sentences were removed.

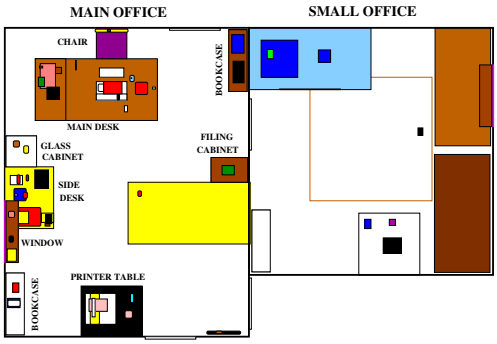


Figure 2: Our virtual environment (top view)

Table 1 shows two original texts compared with the corresponding modified texts (the changed portions in the originals have been italicized).

Our evaluation consists of two experiments: (1) ICGs for sentence pairs, and (2) UCGs for sentence sequences.

Experiment 1. We extracted 106 sentence pairs from our corpus — each pair containing one declarative and one imperative sentence. To evaluate the ICGs, we constructed a virtual environment comprising a main office and a small office (Figure 2). Furniture and objects were placed in a manner compatible with what was mentioned in the requests in our corpus; distractors were also placed in the virtual space. In total, our environment contains 183 instantiated concepts (109 office and household objects, 43 actions and 31 relations). The (x, y, z) coordinates, colour and dimensions of these objects were stored in a knowledge base. Since we have two sentences and their mode is known, no corpus-based information is used for this experiment, and hence no training is required.

[6] We acknowledge the modest size of our corpus compared to that of some publicly available corpora, e.g., ATIS. However, we must generate our own corpus since our task differs in nature from the tasks where these large corpora are used.

Original	Get my book *"The Wizard of Oz"* from my office. It's green *and yellow*. *It has a picture of a dog and a girl on it*. It's in my *desk* drawer on the right *side* of my desk*, the second drawer down*. *If it's not there, it's somewhere on my shelves that are on the left side of my office as you face the window.*
Modified	Get my book from my office. It's green. It's in my drawer on the right of my desk.
Original	*DORIS, I left* my mug in my office *and I want a coffee*. *Can you* go into my office *and* get my mug. It is on top of the cabinet *that is* on the left *side* of my desk.
Modified	My mug is in my office. Go into my office. Get my mug. It is on top of the cabinet on the left of my desk.

Table 1: Original and modified text

Experiment 2. Since UCGs contain only syntactic information, no additional setup was required. However, for this experiment we need to train our mode classifier (Section 2.2), and estimate the probability distribution of referring to a particular sentence in a sequence (Section 2.3). Owing to the small size of our corpus, we use leave-one-out cross validation.

For both experiments, *Scusi?* was set to generate up to 300 sub-interpretations (including parse trees, UCGs and ICGs) for each sentence in the test-set; on average, it took less than 1 second to go from a text to a UCG. An interpretation was deemed successful if it correctly represented the speaker's intention, which was represented by an imperative Gold ICG for the first experiment, and a sequence of imperative Gold UCGs for the second experiment. These Gold interpretations were manually constructed by the authors through consensus-based annotation (Ang et al., 2002). As mentioned in Section 2, we evaluated only imperative ICGs and UCGs, as they contain the actions the robot is expected to perform.

4.2 Results

Table 2 summarizes our results. Column 1 shows the type of outcome being evaluated (ICGs in Experiment 1, and UCG sequences and individual UCGs in Experiment 2). The next two columns display how many sentences had Gold interpretations whose probability was among the top-1 and top-3 probabilities. The average *rank* of the Gold interpretation appears in Column 4 ("not found" Gold interpretations are excluded from this rank). The rank of an interpretation is its position in a list sorted in descending order of probability (starting from position 0), such that all equiprobable interpretations have the same position. Columns 5 and 6 respectively show the median and 75%-ile rank of the Gold interpretation. The number of Gold interpretations that were not found appears in Column 7, and the total number of requests/UCGs is shown in the last column.

Experiment 1. As seen in the first row of Table 2, the Gold ICG was top ranked in 75.5% of the cases, and top-3 ranked in 85.8%. The average rank of 2.17 is mainly due to 7 outliers, which together with the "not-found" Gold ICG, are due to PP-attachment issues, e.g., for the sentence pair "Fetch my phone from my desk. It is near the keyboard.", the top parses and resultant UCGs have "near the keyboard" attached to "the desk" (instead of "the phone"). Nonetheless, the top-ranked interpretation correctly identified the intended object and action in 5 of these 7 cases. Median and 75%-ile results confirm that most of the Gold ICGs are top ranked.

Experiment 2. As seen in the second row of Table 2, the Gold UCG sequence was top ranked for 51.3% of the requests, and top-3 ranked for 53.0% of the requests. The third row shows that 62.4% of the individual Gold UCGs were top-ranked, and 65.4% were top-3 ranked. This indicates that when *Scusi?* cannot fully interpret a request, it can often generate a partially correct interpretation. As for Experiment 1, the average rank of 3.14 for the Gold UCG sequences is due to outliers, several of which were ranked above 30. The median and 75%-ile results show that when *Scusi?* generates the correct interpretation, it tends to be highly ranked.

Unlike Experiment 1, in Experiment 2 there is little difference between the top-1 and top-3 results. A possible explanation is that in Experiment 1, the top-ranked UCG may yield several probable ICGs, such that the Gold ICG is not top ranked — a phenomenon that is not observable at the UCG stage.

Even though Experiment 2 reaches only the

Table 2: *Scusi?*'s interpretation performance

	# Gold interps. with prob. in top 1	with prob. in top 3	Average rank	Median rank	75%-ile rank	Not found	Total #
ICGs	80 (75.5%)	91 (85.8%)	2.17	0	0	1 (0.9%)	106 reqs.
UCG seqs.	59 (51.3%)	61 (53.0%)	3.14	0	1	36 (31.3%)	115 reqs.
UCGs	146 (62.4%)	153 (65.4%)	NA	NA	NA	55 (23.5%)	234 UCGs

UCG stage, *Scusi?*'s performance for this experiment is worse than for Experiment 1, as there are more grounds for uncertainty. Table 2 shows that 31.3% of Gold UCG sequences and 23.5% of Gold UCGs were not found. Most of these cases (as well as the poorly ranked UCG sequences and UCGs) were due to (1) imperatives with object specifications (19 sequences), (2) wrong anaphora resolution (6 sequences), and (3) wrong PP-attachment (6 sequences). In the near future, we will refine the merging process to address the first problem. The second problem occurs mainly when there are multiple anaphoric references in a sequence. We propose to include this factor in our estimation of the probability of referring to a sentence. We intend to alleviate the PP-attachment problem, which also occurred in Experiment 1, by interleaving semantic and pragmatic interpretation of prepositional phrases as done in (Brick and Scheutz, 2007). The expectation is that this will improve the rank of candidates which are pragmatically more plausible.

5 Related Research

This research extends our mechanism for interpreting stand-alone utterances (Zukerman et al., 2008) to the interpretation of sentence sequences. Our approach may be viewed as an *information state* approach (Larsson and Traum, 2000; Becker et al., 2006), in the sense that sentences may update different informational aspects of other sentences, without requiring a particular "legal" set of dialogue acts. However, unlike these information state approaches, ours is probabilistic.

Several researchers have investigated probabilistic approaches to the interpretation of spoken utterances in dialogue systems, e.g., (Pfleger et al., 2003; Higashinaka et al., 2003; He and Young, 2003; Gorniak and Roy, 2005; Hüwel and Wrede, 2006). Pfleger *et al.* (2003) and Hüwel and Wrede (2006) employ modality fusion to combine hypotheses from different analyzers (linguistic, visual and gesture), and apply a scoring mechanism to rank the resultant hypotheses. They disambiguate referring expressions by choosing the first object that satisfies a 'differentiation criterion', hence their system does not handle situations where more than one object satisfies this criterion. He and Young (2003) and Gorniak and Roy (2005) use Hidden Markov Models for the ASR stage. However, these systems do not handle utterance sequences. Like *Scusi?*, the system developed by Higashinaka *et al.* (2003) maintains multiple interpretations, but with respect to dialogue acts, rather than the propositional content of sentences. All the above systems employ semantic grammars, while *Scusi?* uses generic, statistical tools, and incorporates semantic- and domain-related information only in the final stage of the interpretation process. This approach is supported by the findings reported in (Knight et al., 2001) for relatively unconstrained utterances by users unfamiliar with the system, such as those expected by *DORIS*.

Our mechanism is also well suited for processing replies to clarification questions (Horvitz and Paek, 2000; Bohus and Rudnicky, 2005), since a reply can be considered an additional sentence to be incorporated into top-ranked UCG sequences. Further, our probabilistic output can be used by a utility-based dialogue manager (Horvitz and Paek, 2000).

6 Conclusion

We have extended *Scusi?*, our spoken language interpretation system, to interpret sentence sequences. Specifically, we have offered a procedure that combines the interpretations of the sentences in a sequence, and presented a formalism for estimating the probability of the merged interpretation. This formalism supports the comparison of interpretations comprising different numbers of UCGs obtained from different mergers.

Our empirical evaluation shows that *Scusi?* performs well for textual input corresponding to (modified) sentence pairs. However, we still need

to address some issues pertaining to the integration of UCGs for sentence sequences of arbitrary length. Thereafter, we propose to investigate the influence of speech recognition performance on *Scusi?*'s performance. In the future, we intend to expand *Scusi?*'s grammatical capabilities.

Acknowledgments

This research was supported in part by grant DP0878195 from the Australian Research Council.

References

J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke. 2002. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *ICSLP'2002 – Proceedings of the 7th International Conference on Spoken Language Processing*, pages 2037–2040, Denver, Colorado.

T. Becker, P. Poller, J. Schehl, N. Blaylock, C. Gerstenberger, and I. Kruijff-Korbayová. 2006. The SAMMIE system: Multimodal in-car dialogue. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 57–60, Sydney, Australia.

D. Bohus and A. Rudnicky. 2005. Constructing accurate beliefs in spoken dialog systems. In *ASRU'05 – Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 272–277, San Juan, Puerto Rico.

T. Brick and M. Scheutz. 2007. Incremental natural language processing for HRI. In *HRI 2007 – Proceedings of the 2nd ACM/IEEE International Conference on Human-Robot Interaction*, pages 263–270, Washington, D.C.

P. Gorniak and D. Roy. 2005. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *ICMI'05 – Proceedings of the 7th International Conference on Multimodal Interfaces*, pages 138–143, Trento, Italy.

Y. He and S. Young. 2003. A data-driven spoken language understanding system. In *ASRU'03 – Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 583–588, St. Thomas, US Virgin Islands.

R. Higashinaka, M. Nakano, and K. Aikawa. 2003. Corpus-Based discourse understanding in spoken dialogue systems. In *ACL-2003 – Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 240–247, Sapporo, Japan.

E. Horvitz and T. Paek. 2000. DeepListener: Harnessing expected utility to guide clarification dialog in spoken language systems. In *ICSLP'2000 – Proceedings of the 6th International Conference on Spoken Language Processing*, pages 229–229, Beijing, China.

S. Hüwel and B. Wrede. 2006. Spontaneous speech understanding for robust multi-modal human-robot communication. In *Proceedings of the COLING/ACL Main Conference Poster Sessions*, pages 391–398, Sydney, Australia.

S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. 2001. Comparing grammar-based and robust approaches to speech understanding: A case study. In *Proceedings of Eurospeech 2001*, pages 1779–1782, Aalborg, Denmark.

S. Lappin and H.J. Leass. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20:535–561.

S. Larsson and D. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340.

C. Leacock and M. Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*, pages 265–285. MIT Press.

H.T. Ng, Y. Zhou, R. Dale, and M. Gardiner. 2005. A machine learning approach to identification and resolution of one-anaphora. In *IJCAI-05 – Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 1105–1110, Edinburgh, Scotland.

N. Pfleger, R. Engel, and J. Alexandersson. 2003. Robust multimodal discourse processing. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue*, pages 107–114, Saarbrücken, Germany.

J.F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.

I. Zukerman, E. Makalic, M. Niemann, and S. George. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, pages 581–592, Hanoi, Vietnam.

G. Zweig, D. Bohus, X. Li, and P. Nguyen. 2008. Structured models for joint decoding of repeated utterances. In *Proceedings of Interspeech 2008*, pages 1157–1160, Brisbane, Australia.

Participant Subjectivity and Involvement as a Basis for Discourse Segmentation

John Niekrasz and Johanna Moore
Human Communication Research Centre
School of Informatics
University of Edinburgh
{jniekras,jmoore}@inf.ed.ac.uk

Abstract

We propose a framework for analyzing episodic conversational activities in terms of expressed relationships between the participants and utterance content. We test the hypothesis that linguistic features which express such properties, e.g. tense, aspect, and person deixis, are a useful basis for automatic intentional discourse segmentation. We present a novel algorithm and test our hypothesis on a set of intentionally segmented conversational monologues. Our algorithm performs better than a simple baseline and as well as or better than well-known lexical-semantic segmentation methods.

1 Introduction

This paper concerns the analysis of conversations in terms of communicative activities. Examples of the kinds of activities we are interested in include relating a personal experience, making a group decision, committing to future action, and giving instructions. The reason we are interested in these kinds of events is that they are part of participants' common-sense notion of the goals and accomplishments of a dialogue. They are part of participants' subjective experience of what happened and show up in summaries of conversations such as meeting minutes. We therefore consider them an ideal target for the practical, common-sense description of conversations.

Activities like these commonly occur as cohesive *episodes* of multiple turns within a conversation (Korolija, 1998). They represent an intermediate level of dialogue structure – greater than a single speech act but still small enough to have a potentially well-defined singular purpose. They have a temporal granularity of anywhere from a few seconds to several minutes.

Ultimately, it would be useful to use descriptions of such activities in automatic summarization technologies for conversational genres. This would provide an activity-oriented summary describing what 'happened' that would complement one based on information content or what the conversation was 'about'. Part of our research goal is thus to identify a set of discourse features for segmenting, classifying, and describing conversations in this way.

1.1 Participant subjectivity and involvement

The approach we take to this problem is founded upon two basic ideas. The first is that the activities we are interested in represent a coarse level of the *intentional structure* of dialogue (Grosz and Sidner, 1986). In other words, each activity is unified by a common purpose that is shared between the participants. This suggests there may be linguistic properties which are shared amongst the utterances of a given activity episode.

The second idea concerns the properties which distinguish different activity types. We propose that activity types may be usefully distinguished according to two complex properties of utterances, both of which concern relationships between the participants and the utterance: ***participant subjectivity*** and ***participant involvement***. Participant subjectivity concerns attitudinal and perspectival relationships *toward* the dialogue content. This includes properties such as whether the utterance expresses the private mental state of the speaker, or the participants' temporal relationship to a described event. Participant involvement concerns the roles participants play *within* the dialogue con-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 54–61,
Queen Mary University of London, September 2009.

tent, e.g., as the agent of a described event.

1.2 Intentional segmentation

The hypothesis we test in this paper is that the linguistic phenomena which express participant-relational properties may be used as an effective means of intentional discourse segmentation. This is based on the idea that if adjacent discourse segments have different activity types, then they are distinguishable by participant-relational features. If we can reliably extract such features, then this would allow segmentation of the dialogue accordingly.

We test our hypothesis by constructing an algorithm and examining its performance on an existing set of intentionally segmented conversational monologues (i.e., one person speaks while another listens) (Passonneau and Litman, 1997, henceforth P&L). While our long term goal is to apply our techniques to multi-party conversations (and to a somewhat coarser-grained analysis), using this dataset is a stepping-stone toward that end which allows us to compare our results with existing intentional segmentation algorithms.

An example dialogue extract from the dataset is shown in Dialogue 1. Two horizontal lines indicate a segment boundary which was identified by at least 3 of 7 annotators. A single horizontal line indicates a segment boundary which was identified by 2 or fewer annotators. In the example, there are three basic types of discourse activity distinguishable according to the properties of participant subjectivity and participant involvement. The segments beginning at 22.1 and 26.2 share the use of the historical present tense – a type of participant subjectivity – in a narrative activity type. Utterances 24.1 and 25.1, on the other hand, are about the prior perceptions of the speaker, a type of participant involvement in a past event. The segment beginning at 24.2 is a type of generic description activity, exhibiting its own distinct configuration of participant relational features, such as the generic *you* and present tense.

PearStories-09 (Chafe, 1980)

21.2	okay.
22.1	Meanwhile,
22.2	there are three little boys,
22.3	up on the road a little bit,
22.4	and they see this little accident.
23.1	And u-h they come over,
23.2	and they help him,
23.3	and you know,
23.4	help him pick up the pears and everything.
24.1	A-nd the one thing that struck me about the- three little boys that were there,
24.2	is that one had ay uh I don't know what you call them,
24.3	but it's a paddle,
24.4	and a ball-,
24.5	is attached to the paddle,
24.6	and you know you bounce it?
25.1	And that sound was really prominent.
26.1	Well anyway,
26.2	so- u-m tsk all the pears are picked up,
26.3	and he's on his way again,

Dialogue 1: An example dialogue extract showing intentional segment boundaries.

We structure the rest of the paper as follows. First, we begin by describing related and supporting theoretical work. This is followed by a test of our main hypothesis. We then follow this with a similar experiment which contextualizes our work both theoretically and in practical terms with respect to the most commonly studied segmentation task: topic segmentation. We finish with a general discussion of the implications of our experiments.

2 Background and Related Work

The influential work of Grosz and Sidner (1986) provides a helpful starting point for understanding our approach. Their theory suggests that intentions (which equate to the goals and purposes of a dialogue) are a foundation for the structure of discourse. The individual discourse purposes that emerge in a dialogue relate directly to the natural aggregation of utterances into discourse segments. The attentional state of the dialogue, which contains salient objects and relations and allows for the efficient generation and interpretation of utterances, is then dependent upon this interrelated intentional and linguistic structure in the emerging dialogue.

Grosz and Sidner's theory suggests that attentional state is parasitic upon the underlying intentional structure. This implication has informed many approaches which relate referring expressions (an attentional phenomenon) to discourse structure. One example is Centering theory (Grosz et al., 1995), which concerns the relationship of referring expressions to discourse coherence. Another is P&L, who demonstrated that co-reference and inferred relations between noun phrases are a useful basis for automatic intentional segmentation.

Our approach expands on this by highlighting

the fact that objects that are in focus within the attentional state have an important quality which may be exploited: they are focused upon *by* the participants from particular *points of view*. In addition, the objects may in fact *be* the participants themselves. We would expect the linguistic features which express such relationships (e.g., aspect, subjectivity, modality, and person deixis) to therefore correlate with intentional structure, and to do so in a way which is important to participants' subjective experience of the dialogue.

This approach is supported by a theory put forth by Chafe (1994), who describes how speakers can express ideas from alternative perspectives. For example, a subject who is recounting the events of a movie of a man picking pears might say "the man was picking pears", "the man picks some pears", or "you see a man picking pears." Each variant is an expression of the same idea but reflects a different perspective toward, or manner of participation in, the described event. The linguistic variation one sees in this example is in the properties of tense and aspect in the main clause (and in the last variant, a perspectival superordinate clause which uses the generic *you*). We have observed that discourse coheres in these perspectival terms, with shifts of perspective usually occurring at intentional boundaries.

Wiebe (1994; 1995) has investigated a phenomenon closely related to this: point-of-view and subjectivity in fictional narrative. She notes that paragraph-level blocks of text often share a common *objective* or *subjective* context. That is, sentences may or may not be conveyed from the point-of-view of individuals, e.g., the author or the characters within the narrative. Sentences continue, resume, or initiate such contexts, and she develops automatic methods for determining when the contexts shift and whose point-of-view is being taken. Her algorithm provides a detailed method for analyzing written fiction, but has not been developed for conversational or non-narrative genres.

Smith's (2003) analysis of texts, however, draws a more general set of connections between the content of sentences and types of discourse segments. She does this by analyzing texts at the level of short passages and determines a non-exhaustive list of five basic "discourse modes" occurring at that level: narrative, description, report, information, and argument. The mode of a passage is determined by the type of situations described in the text (e.g., event, state, general stative, etc.) and the temporal progression of the situations in the discourse. Situation types are in turn organized according to the perspectival properties of aspect and temporal location. A narrative passage, for example, relates principally specific events and states, with dynamic temporal advancement of narrative time between sentences. On the other hand, an information passage relates primarily general statives with atemporal progression.

3 Automatic Segmentation Experiment

The analysis described in the previous sections suggests that participant-relational features correlate with the intentional structure of discourse. In this section we describe an experiment which tests the hypothesis that a small set of such features, i.e., tense, aspect, and first- and second-person pronouns, are a useful basis for intentional segmentation.

3.1 Data

Our experiment uses the same dataset as P&L, a corpus of 20 spoken narrative monologues known as the Pear Stories (Chafe, 1980). Chafe asked subjects to view a silent movie and then summarize it for a second person. Their speech was then manually transcribed and segmented into prosodic phrases. This resulted in a mean 100 phrases per narrative and a mean 6.7 words per phrase. P&L later had each narrative segmented by seven annotators according to an informal definition of communicative intention. Each prosodic phrase boundary was a possible discourse segment boundary. Using Cochran's Q test, they concluded that an appropriate gold standard could be produced by using the set of boundaries assigned by at least three of the seven annotators. This is the gold standard we use in this paper. It assigns a boundary at a mean 16.9% ($\sigma = 4.5\%$) of the possible boundary sites in each narrative. The result is a mean discourse segment length of 5.9 prosodic phrases, ($\sigma = 1.4$ across the means of each narrative).

3.2 Algorithm

The basic idea behind our algorithm is to distinguish utterances according to the type of activity in which they occur. To do this, we identify a set of utterance properties relating to par-

ticipant subjectivity and participant involvement, according to which activity types may be distinguished. We then develop a routine for automatically extracting the linguistic features which indicate such properties. Finally, the dialogue is segmented at locations of high discontinuity in that feature space. The algorithm works in four phases: pre-processing, feature extraction, similarity measurement, and boundary assignment.

3.2.1 Pre-processing

For pre-processing, disfluencies are removed by deleting repeated strings of words and incomplete words. The transcript is then parsed (Klein and Manning, 2002), and a collection of typed grammatical dependencies are generated (de Marneffe et al., 2006). The TTT2 chunker (Grover and Tobin, 2006) is then used to perform tense and aspect tagging.

3.2.2 Feature extraction

Feature extraction is the most important and novel part of our algorithm. Each prosodic phrase (the corpus uses prosodic phrases as sentence-like units, see Data section) is assigned values for five binary features. The extracted features correspond to a set of utterance properties which were identified manually through corpus analysis. The first four relate directly to individual activity types and are therefore mutually exclusive properties.

first-person participation [1P] – helps to distinguish meta-discussion between the speaker and hearer (e.g., "Did I tell you that?")

generic second-person [2P-GEN] – helps to distinguish narration told from the perspective of a generic participant (e.g., "You see a man picking pears")

third-person stative/progressive [3P-STAT] – helps to distinguish narrative activities related to "setting the scene" (e.g., "[There is a man | a man is] picking pears")

third-person event [3P-EVENT] – helps to distinguish event-driven third-person narrative activities (e.g. "The man drops the pears")

past/non-past [PAST] – helps to distinguish narrative activities by temporal orientation (e.g. "The man drops the pears" vs. "The man dropped the pears")

Feature extraction works by identifying the linguistic elements that indicate each utterance property. First, prosodic phrases containing a first- or second-person pronoun in grammatical subject or object relation to any clause are identified (common fillers like *you know*, *I think*, and *I don't know* are ignored). Of the identified phrases, those with first-person pronouns are marked for 1P, while the others are marked for 2P-GEN. For the remaining prosodic phrases, those with a matrix clause are identified. Of those identified, if either its head verb is *be* or *have*, it is tagged by TTT2 as having progressive aspect, or the prosodic phrase contains an existential *there*, then it is marked for 3P-STAT. The others are marked for 3P-EVENT. Finally, if the matrix clause was tagged as past tense, the phrase is marked for PAST. In cases where no participant-relational features are identified (e.g., no matrix clause, no pronouns), the prosodic phrase is assigned the same features as the preceding one, effectively marking a continuation of the current activity type.

3.2.3 Similarity measurement

Similarity measurement is calculated according to the cosine similarity $cos(v_i, c_i)$ between the feature vector v_i of each prosodic phrase i and a weighted sum c_i of the feature vectors in the preceding context. The algorithm requires a parameter l to be set for the desired mean segment length. This determines the window $w = floor(l/2)$ of preceding utterances to be used. The weighted sum representing the preceding context is computed as $c_i = \sum_{j=1}^{w}((1 + w - j)/w)v_{i-j}$, which gives increasingly greater weight to more recent phrases.

3.2.4 Boundary assignment

In the final step, the algorithm assigns boundaries where the similarity score is lowest, namely prior to prosodic phrases where cos is less than the first $1/l$ quantile for that discourse.

3.3 Experimental Method and Evaluation

Our experiment compares the performance of our novel algorithm (which we call NM09) with a naive baseline and a well-known alternative method – P&L's co-reference based NP algorithm. To our knowledge, P&L is the only existing publication describing algorithms designed specifically for intentional segmentation of dialogue. Their NP algorithm exploits annotations of direct and

inferred relations between noun phrases in adjacent units. Inspired by Centering theory (Grosz et al., 1995), these annotations are used in a computational account of discourse focus to measure coherence. Although adding pause-based features improved results slightly, the NP method was the clear winner amongst those using a single feature type and produced very good results.

The NP algorithm requires co-reference annotations as input, so to create a fully-automatic version (NP-AUTO) we have employed a state-of-the-art co-reference resolution system (Poesio and Kabadjov, 2004) to generate the required input. We also include results based on P&L's original human co-reference annotations (NP-HUMAN).

For reference, we include a baseline that randomly assigns boundaries at the same mean frequency as the gold-standard annotations, i.e., a sequence drawn from the Bernoulli distribution with success probability $p = 0.169$ (this probability determines the value of the target segment length parameter l in our own algorithm). As a top-line reference, we calculate the mean of the seven annotators' scores with respect to the three-annotator gold standard.

For evaluation we employ two types of measure. On one hand, we use $P(k)$ (Beeferman et al., 1999) as an error measure designed to accommodate near-miss boundary assignments. It is useful because it estimates the probability that two randomly drawn points will be assigned incorrectly to either the same or different segments. On the other hand, we use Cohen's Kappa (κ) to evaluate the precise placement of boundaries such that each potential boundary site is considered a binary classification. While κ is typically used to evaluate inter-annotator agreement, it is a useful measure of classification accuracy in our experiment for two reasons. First, it accounts for the strong class bias in our data. Second, it allows a direct and intuitive comparison with our inter-annotator top-line reference. We also provide results for the commonly-used IR measures F_1, recall, and precision. These are useful for comparing with previous results in the literature and provide a more widely-understood measure of the accuracy of the results. Precision and recall are also helpful in revealing the effects of any classification bias the algorithms may have.

The results are calculated for 18 of the 20 narratives, as manual feature development involved the use of two randomly selected narratives as development data. The one exception is NP-HUMAN, which is evaluated on the 10 narratives for which there are manual co-reference annotations.

Table 1: Mean results for the 18 test narratives.

	$P(k)$	κ	F_1	Rec.	Prec.
Human	.21	.58	.65	.64	.69
NP-HUMAN	.35	.38	.40	.52	.46
NM09	.44	.11	.24	.23	.28
NP-AUTO	.52	.03	.27	.71	.17
Random	.50	.00	.15	.14	.17

3.4 Results

The mean results for the 18 narratives, calculated in comparison to the three-annotator gold standard, are shown in Table 1. NP-HUMAN and NM09 are both superior to the random baseline for all measures ($p \leq 0.05$). NP-AUTO, however, is only superior in terms of recall and F_1 ($p \leq 0.05$).

3.5 Discussion

The results indicate that the simple set of features we have chosen can be used for intentional segmentation. While the results are not near human performance, it is encouraging that such a simple set of easily extractable features achieves results that are 19% (κ), 24% ($P(k)$), and 18% (F_1) of human performance, relative to the random baseline.

The other notable result is the very high recall score of NP-AUTO, which helps to produce a respectable F_1 score. However, a low κ reveals that when accounting for class bias, this system is actually not far from the performance of a high recall random classifier.

Error analysis showed that the reason for the problems with NP-AUTO was the lack of reference chains produced by the automatic co-reference system. While the system seems to have performed well for direct co-reference, it did not do well with bridging reference. Inferred relations were an important part of the reference chains produced by P&L, and it is now clear that these play a significant role in the performance of the NP algorithm. Our algorithm is not dependent on this difficult processing problem, which typically requires world knowledge in the form of training on large datasets or the use of large lexical resources.

4 Topic vs. Intentional Segmentation

It is important to place our experiment on intentional segmentation in context with the most commonly studied automatic segmentation task: topic-based segmentation. While the two tasks are distinct, the literature has drawn connections between them which can at times be confusing. In this section, we attempt to clarify those connections by pointing out some of their differences and similarities. We also conduct an experiment comparing our algorithm to well-known topic-segmentation algorithms and discuss the results.

4.1 Automatic segmentation in the literature

One of the most widely-cited discourse segmentation algorithms is TextTiling (Hearst, 1997). Designed to segment texts into multi-paragraph subtopics, it works by operationalizing the notion of lexical cohesion (Halliday and Hasan, 1976). TextTiling and related algorithms exploit the collocation of semantically related lexemes to measure coherence. Recent improvements to this method include the use of alternative lexical similarity metrics like LSA (Choi et al., 2001) and alternative segmentation methods like the minimum cut model (Malioutov and Barzilay, 2006) and ranking and clustering (Choi, 2000). Recently, Bayesian approaches which model topics as a lexical generative process have been employed (Purver et al., 2006; Eisenstein and Barzilay, 2008). What these algorithms all share is a focus on the semantic content of the discourse.

Passonneau and Litman (1997) is another of the most widely-cited articles on discourse segmentation. Their overall approach combines an investigation of prosodic features, cue words, and entity reference. As described above, their approach to using entity reference is motivated by Centering theory (Grosz et al., 1995) and the hypothesis that intentional structure is exhibited in the attentional relationships between discourse referents.

Hearst and P&L try to achieve different goals, but their tasks are nonetheless related. One might reasonably hypothesize, for example, that either lexical similarity or co-reference could be useful to either type of segmentation on the grounds that the two phenomena are clearly related. However, there are also clear differences of intent between the two studies. While there is an obvious difference in the dataset (written expository text vs. spoken narrative monologue), the annotation instructions reflect the difference most clearly. Hearst instructed naive annotators to mark paragraph boundaries "where the *topics* seem to change," whereas P&L asked naive annotators to mark prosodic phrases where the speaker had begun a new *communicative task*.

The results indicate that there is a difference in granularity between the two tasks, with intentional segmentation relating to finer-grained structure. Hearst's segments have a mean of about 200 words to P&L's 40. Also, two hierarchical topic segmentations of meetings (Hsueh, 2008; Gruenstein et al., 2008) have averages above 400 words for the smallest level of segment.

To our knowledge, P&L is the only existing study of automatic intention-based segmentation. However, their work has been frequently cited as a study of topic-oriented segmentation, e.g., (Galley et al., 2003; Eisenstein and Barzilay, 2008). Also, recent research in conversational genres (Galley et al., 2003; Hsueh and Moore, 2007) analyze events like discussing an agenda or giving a presentation, which resemble more intentional categories. Interestingly, these algorithms demonstrate the benefit of including non-lexical, non-semantic features. The results imply that further analysis is needed to understand the links between different types of coherence and different types of segmentation.

4.2 Experiment 2

We have extended the above experiment to compare the results of our novel algorithm with existing topic segmentation methods. We employ Choi's implementations of C99 (Choi, 2000) and TEXTTILING (Hearst, 1997) as examples of well-known topic-oriented methods. While we acknowledge that there are newer algorithms which improve upon this work, these were selected for being well studied and easy to apply out-of-the-box. Our method and evaluation is the same as in the previous experiment.

The mean results for the 18 narratives are shown in Table 2, with the human and baseline score reproduced from the previous table. All three automatic algorithms are superior to the random baseline in terms of $P(k)$, κ, and F_1 ($p \leq 0.05$). The only statistically significant difference ($p \leq 0.05$) between the three automatic methods is between NM09 and TEXTTILING in terms of F_1. The observed difference between NM09 and TEXTTILING in terms of κ is only moderately significant

Table 2: Results comparing our method to topic-oriented segmentation methods.

NP-auto	$P(k)$	κ	F_1	Rec.	Prec.
Human	.21	.58	.65	.64	.69
NM09	.44	.11	.24	.24	.28
C99	.44	.08	.22	.20	.24
TEXTTILING	.41	.05	.18	.16	.21
Random	.50	.00	.15	.14	.17

($p \leq 0.08$). The observed differences between between NM09 and C99 are minimally significant ($p \leq 0.24$) .

4.3 Discussion

The comparable performance achieved by our simple perspective-based approach in comparison to lexical-semantic approaches suggests two main points. First, it validates our novel approach in practical applied terms. It shows that perspective-oriented features, being simple to extract and applicable to a variety of genres, are potentially very useful for automatic discourse segmentation systems.

Second, the results show that the teasing apart of topic-oriented and intentional structure may be quite difficult. Studies of coherence at the level of short passages or episodes (Korolija, 1998) suggest that coherence is established through a complex interaction of topical, intentional, and other contextual factors. In this experiment, the major portion of the dialogues are oriented toward the basic narrative activity which is the premise of the Pear Stories dataset. This means that there are many times when the activity type does not change at intentional boundaries. At other times, the activity type changes but neither the topic nor the set of referents is significantly changed. The different types of algorithms we have tried (i.e., topical, referential, and perspectival) seem to be operating on somewhat orthogonal bases, though it is difficult to say quantitatively how this relates to the types of "communicative task" transitions occurring at the boundaries. In a sense, we have proposed an algorithm for performing "activity type cohesion" which mimics the methods of lexical cohesion but is based upon a different dimension of the discourse. The results indicate that these are both related to intentional structure.

5 General Discussion and Future Work

Future work in intentional segmentation is needed. Our ultimate goal is to extend this work to more conversational domains (e.g., multi-party planning meetings) and to define the richer set of perspectives and related deictic features that would be needed for them. For example, we hypothesize that the different uses of second-person pronouns in conversations (Gupta et al., 2007) are likely to reflect alternative activity types. Our feature set and extraction methods will therefore need to be further developed to capture this complexity.

The other question we would like to address is the relationship between various types of coherence (e.g., topical, referential, perspectival, etc.) and different types (and levels) of discourse structure. Our current approach uses a feature space that is orthogonal to most existing segmentation methods. This has allowed us to gain a deeper understanding of the relationship between certain linguistic features and the underlying intentional structure, but more work is needed.

In terms of practical motivations, we also plan to address the open question of how to effectively combine our feature set with other feature sets which have also been demonstrated to contribute to discourse structuring and segmentation.

References

Doug Beeferman, Adam Berger, and John D. Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

Wallace L. Chafe, editor. 1980. *The Pear Stories: Cognitive, Cultural, and Linguistic Aspects of Narrative Production*, volume 3 of *Advances in Discourse Processes*. Ablex, Norwood, NJ.

Wallace L. Chafe. 1994. *Discourse, Consciousness, and Time: The Flow and Displacement of Conscious Experience in Speaking and Writing*. University of Chicago Press, Chicago.

Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. 2001. Latent semantic analysis for text segmentation. In *Proc. EMNLP*, pages 109–117.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proc. NAACL*, pages 26–33.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. LREC*, pages 562–569.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proc. EMNLP*, pages 334–343.

Michel Galley, Kathleen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. In *Proc. ACL*, pages 562–569.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Barbara J. Grosz, Aravind Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Claire Grover and Richard Tobin. 2006. Rule-based chunking and reusability. In *Proc. LREC*.

Alexander Gruenstein, John Niekrasz, and Matthew Purver. 2008. Meeting structure annotation: Annotations collected with a general purpose toolkit. In L. Dybkjaer and W. Minker, editors, *Recent Trends in Discourse and Dialogue*, pages 247–274.

Surabhi Gupta, John Niekrasz, Matthew Purver, and Daniel Jurafsky. 2007. Resolving "you" in multiparty dialog. In *Proc. SIGdial*, pages 227–230.

M. A. K. Halliday and Ruqayia Hasan. 1976. *Cohesion in English*. Longman, New York.

Marti Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Pei-Yun Hsueh and Johanna D. Moore. 2007. Combining multiple knowledge sources for dialogue segmentation in multimedia archives. In *Proc. ACL*, pages 1016–1023.

Pei-Yun Hsueh. 2008. *Meeting Decision Detection: Multimodal Information Fusion for Multi-Party Dialogue Understanding*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Dan Klein and Christopher D. Manning. 2002. Fast exact inference with a factored model for natural language parsing. In *NIPS 15*.

Natascha Korolija. 1998. *Episodes in talk: Constructing coherence in multiparty conversation*. Ph.D. thesis, Linköping University, The Tema Institute, Department of Communications Studies.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proc. COLING-ACL*, pages 25–32.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139.

Massimo Poesio and Mijail A. Kabadjov. 2004. A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In *Proc. LREC*.

Matthew Purver, Konrad Körding, Thomas Griffiths, and Joshua Tenenbaum. 2006. Unsupervised topic modelling for multi-party spoken discourse. In *Proc. COLING-ACL*, pages 17–24.

Carlota S. Smith. 2003. *Modes of Discourse*. Cambdrige University Press, Cambridge.

Janyce M. Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Janyce M. Wiebe. 1995. References in narrative text. In Judy Duchan, Gail Bruder, and Lynne Hewitt, editors, *Deixis in Narrative: A Cognitive Science Perspective*, pages 263–286.

Genre-Based Paragraph Classification for Sentiment Analysis

Maite Taboada
Department of Linguistics
Simon Fraser University
Burnaby, BC, Canada
mtaboada@sfu.ca

Julian Brooke
Department of Computer Science
University of Toronto
Toronto, ON, Canada
jbrooke@cs.toronto.edu

Manfred Stede
Institute of Linguistics
University of Potsdam
Potsdam, Germany
stede@ling.uni-potsdam.de

Abstract

We present a taxonomy and classification system for distinguishing between different types of paragraphs in movie reviews: formal vs. functional paragraphs and, within the latter, between description and comment. The classification is used for sentiment extraction, achieving improvement over a baseline without paragraph classification.

1 Introduction

Much of the recent explosion in sentiment-related research has focused on finding low-level features that will help predict the polarity of a phrase, sentence or text. Features, widely understood, may be individual words that tend to express sentiment, or other features that indicate not only sentiment, but also polarity. The two main approaches to sentiment extraction, the semantic or lexicon-based, and the machine learning or corpus-based approach, both attempt to identify low-level features that convey opinion. In the semantic approach, the features are lists of words and their prior polarity, (e.g., the adjective *terrible* will have a negative polarity, and maybe intensity, represented as -4; the noun *masterpiece* may be a 5). Our approach is lexicon-based, but we make use of information derived from machine learning classifiers.

Beyond the prior polarity of a word, its local context obviously plays an important role in conveying sentiment. Polanyi and Zaenen (2006) use the term 'contextual valence shifters' to refer to expressions in the local context that may change a word's polarity, such as intensifiers, modal verbs, connectives, and of course negation. Further beyond the local context, the overall structure and organization of the text, influenced by its genre, can help the reader determine how the evaluation is expressed, and where it lies. Polanyi and Zaenen (2006) also cite genre constraints as relevant factors in calculating sentiment.

Among the many definitions of genre, we take the view of Systemic Functional Linguistics that genres are purposeful activities that develop in stages, or parts (Eggins and Martin, 1997), which can be identified by lexicogrammatical properties (Eggins and Slade, 1997). Our proposal is that, once we have identified different stages in a text, the stages can be factored in the calculation of sentiment, by weighing more heavily those that are more likely to contain evaluation, an approach also pursued in automatic summarization (Seki et al., 2006).

To test this hypothesis, we created a taxonomy of stages specific to the genre of movie reviews, and annotated a set of texts. We then trained various classifiers to differentiate the stages. Having identified the stages, we lowered the weight of those that contained mostly description. Our results show that we can achieve improvement over a baseline when classifying the polarity of texts, even with a classifier that can stand to improve (at 71.1% accuracy). The best performance comes from weights derived from the output of a linear regression classifier.

We first describe our inventory of stages and the manual annotation (Section 2), and in Section 3 turn to automatic stage classification. After describing our approach to sentiment classification of texts in Section 4, we describe experiments to improve its performance with the information on stages in Section 5. Section 6 dis-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 62–70,
Queen Mary University of London, September 2009.

cusses related work, and Section 7 provides conclusions.

2 Stages in movie reviews

Within the larger *review* genre, we focus on movie reviews. Movie reviews are particularly difficult to classify (Turney, 2002), because large portions of the review contain description of the plot, the characters, actors, director, etc., or background information about the film.

Our approach is based on the work of Bieler et al. (2007), who identify formal and functional zones (stages) within German movie reviews. Formal zones are parts of the text that contribute factual information about the cast and the credits, and also about the review itself (author, date of publication and the reviewer's rating of the movie). Functional zones contain the main gist of the review, and can be divided roughly into *description* and *comment*. Bieler et al. showed that functional zones could be identified using 5-gram SVM classifiers built from an annotated German corpus.

2.1 Taxonomy

In addition to the basic Describe/Comment distinction in Bieler et al., we use a Describe+Comment label, as in our data it is often the case that both description and comment are present in the same paragraph. We decided that a paragraph could be labeled as Describe+Comment when it contained at least a clause of each, and when the comment part could be assigned a polarity (i.e., it was not only subjective, but also clearly positive or negative).

Each of the three high-level tags has a subtag, a feature also present in Bieler et al.'s manual annotation. The five subtags are: overall, plot, actors/characters, specific and general. 'Specific' refers to one particular aspect of the movie (not plot or characters), whereas 'general' refers to multiple topics in the same stage (special effects and cinematography at the same time). Outside the Comment/Describe scale, we also include tags such as Background (discussion of other movies or events outside the movie being reviewed), Interpretation (subjective but not opinionated or polar), and Quotes. Altogether, the annotation system includes 40 tags, with 22 formal and 18 functional zones. Full lists of zone/stage labels are provided in Appendix A.

2.2 Manual annotation

We collected 100 texts from rottentomatoes.com, trying to include one positive and one negative review for the same movie. The reviews are part of the "Top Critics" section of the site, all of them published in newspapers or on-line magazines. We restricted the texts to "Top Critics" because we wanted well-structured, polished texts, unlike those found in some on-line review sites. Future work will address those more informal reviews.

The 100 reviews contain 83,275 words and 1,542 paragraphs. The annotation was performed at the paragraph level. Although stages may span across paragraphs, and paragraphs may contain more than one stage, there is a close relationship between paragraphs and stages. The restriction also resulted in a more reliable annotation, performed with the PALinkA annotation tool (Orasan, 2003).

The annotation was performed by one of the authors, and we carried out reliability tests with two other annotators, one another one of the authors, who helped develop the taxonomy, and the third one a project member who read the annotation guidelines[1], and received a few hours' training in the labels and software. We used Fleiss' kappa (Fleiss, 1971), which extends easily to the case of multiple raters (Di Eugenio and Glass, 2004). We all annotated four texts. The results of the reliability tests show a reasonable agreement level for the distinction between formal and functional zones (.84 for the 3-rater kappa). The lowest reliability was for the 3-way distinction in the functional zones (.68 for the first two raters, and .54 for the three raters). The full kappa values for all the distinctions are provided in Appendix B. After the reliability test, one of the authors performed the full annotation for all 100 texts. Table 1 shows the breakdown of high-level stages for the 100 texts.

Stage	Count
Describe	347
Comment	237
Describe+Comment	237
Background	51
Interpretation	22
Quote	2
Formal	646

Table 1. Stages in 100 text RT corpus

[1]Available from http://www.sfu.ca/~mtaboada/nserc-project.html

3 Classifying stages

Our first classification task aims at distinguishing the two main types of functional zones, Comment and Describe, vs. Formal zones.

3.1 Features

We test two different sets of features. The first, following Bieler et al. (2007), consists of 5-grams (including unigrams, bigrams, 3-grams and 4-grams), although we note in our case that there was essentially no performance benefit beyond 3-grams. We limited the size of our feature set to n-grams that appeared at least 4 times in our training corpus. For the 2 class task (no formal zones), this resulted in 8,092 binary features, and for the 3 and 4 class task there were 9,357 binary n-gram features.

The second set of features captures different aspects of genre and evaluation, and can in turn be divided into four different types, according to source. With two exceptions (features indicating whether a paragraph was the first or last paragraph in text), the features were numerical (frequency) and normalized to the length of the paragraph.

The first group of genre features comes from Biber (1988), who attempted to characterize dimensions of genre. The features here include frequency of first, second and third person pronouns; demonstrative pronouns; place and time adverbials; intensifiers; and modals, among a number of others.

The second category of genre features includes discourse markers, primarily from Knott (1996), that indicate contrast, comparison, causation, evidence, condition, and similar relations.

The third type of genre features was a list of 500 adjectives classified in terms of Appraisal (Martin and White, 2005) as indicating Appreciation, Judgment or Affect. Appraisal categories have been shown to be useful in improving the performance of polarity classifiers (Whitelaw et al., 2005).

Finally, we also include text statistics as features, such as average length of words and sentences and position of paragraphs in the text.

3.2 Classifiers

To classify paragraphs in the text, we use the WEKA suite (Witten and Frank, 2005), testing three popular machine learning algorithms: Naïve Bayes, Support Vector Machine, and Linear Regression (preliminary testing with Decision Trees suggests that it is not appropriate for this task). Training parameters were set to default values.

In order to use Linear Regression, which provides a numerical output based on feature values and derived feature weights, we have to conceive of Comment/Describe/Describe+Comment not as nominal (or ordinal) classes, but rather as corresponding to a Comment/Describe ratio, with "pure" Describe at one end and "pure" Comment at the other. For training, we assign a 0 value (a Comment ratio) to all paragraphs tagged Describe and a 1 to all Comment paragraphs; for Describe+Comment, various options (including omission of this data) were tested. The time required to train a linear regression classifier on a large feature set proved to be prohibitive, and performance with smaller sets of features generally quite poor, so for the linear regression classifier we present results only for our compact set of genre features.

3.3 Performance

Table 2 shows the performance of classifier/feature-set combinations for the 2-, 3-, and 4-class tasks on the 100-text training set, with 10-fold cross-validation, in terms of precision (P), recall (R) and F-measure [2]. SVM and Naïve Bayes provide comparable performance, although there is considerable variation, particularly with respect to the feature set; the SVM is a significantly ($p<0.05$) better choice for our genre features [3], while for the n-gram features the Bayes classification is generally preferred. The SVM-genre classifier significantly outperforms the other classifiers in the 2-class task; these genre features, however, are not as useful as 5-grams at identifying Formal zones (the n-gram classifier, by contrast, can make use of words such as *cast*). In general, formal zone classification is fairly straightforward, whereas identification of Describe+Comment is quite difficult, and the SVM-genre classifier, which is more sensitive to frequency bias, elects to (essentially) ignore this category in order to boost overall accuracy.

To evaluate a linear regression (LR) classifier, we calculate correlation coefficient ρ, which reflects the goodness of fit of the line to the data. Table 3 shows values for the classifiers built from the corpus, with various Comment ratios

[2] For the 2- and 3-way classifiers, Describe+Comment paragraphs are treated as Comment. This balances the numbers of each class, ultimately improving performance.

[3] All significance tests use chi-square (χ^2).

Classifier	Comment			Describe			Formal			Desc+Comm			Overall Accuracy
	P	R	F	P	R	F	P	R	F	P	R	F	
2-class-5-gram-Bayes	.66	**.79**	.72	.70	.55	.62	-	-	-	-	-	-	68.0
2-class-5-gram-SVM	.53	.63	.64	.68	.69	**.69**	-	-	-	-	-	-	66.8
2-class-genre-Bayes	.66	.75	.70	.67	.57	.61	-	-	-	-	-	-	66.2
2-class-genre-SVM	**.71**	.76	**.74**	**.71**	.65	.68	-	-	-	-	-	-	**71.1**
3-class-5-gram-Bayes	.69	.49	.57	.66	**.78**	**.71**	**.92**	**.97**	**.95**	-	-	-	**78.1**
3-class-5-gram-SVM	.64	.63	.63	**.68**	.65	.65	.91	**.97**	.94	-	-	-	77.2
3-class-genre-Bayes	**.68**	.68	.66	.67	.46	.55	.84	.96	.90	-	-	-	74.0
3-class-genre-SVM	.66	**.71**	**.68**	.67	.56	.61	.90	.94	.92	-	-	-	76.8
4-class-5-gram-Bayes	**.46**	.35	.38	**.69**	.47	.56	**.92**	**.97**	**.95**	**.42**	**.64**	**.51**	69.0
4-class-5-gram-SVM	.43	**.41**	.44	.59	.62	.60	.91	**.97**	.94	.45	.41	.42	**69.6**
4-class-genre-Bayes	.38	.31	.34	.66	.30	.41	.86	**.97**	.90	.33	.60	.42	62.3
4-class-genre-SVM	.46	.32	.38	.53	**.82**	.65	.87	.94	.90	.26	.03	.06	67.4

Table 2. Stage identification performance of various categorical classifiers

(C) assigned to paragraphs with the Describe+Comment tag, and with Describe+Comment paragraphs removed from consideration.

Classifier	ρ
LR, Des+Com C = 0	.37
LR, Des+Com C = 0.25	.44
LR, Des+Com C = 0.5	.47
LR, Des+Com C = 0.75	.46
LR, Des+Com C = 1	.43
LR, No Des+Com	.50

Table 3. Correlation coefficients for LR classifiers

The drop in correlation when more extreme values are assigned to Describe+Comment suggests that Describe+Comment paragraphs do indeed belong in the middle of the Comment spectrum. Since there is a good deal of variation in the amount of comment across Describe+Comment paragraphs, the best correlation comes with complete removal of these somewhat unreliable paragraphs. Overall, these numbers indicate that variations in relevant features are able to predict roughly 50% of the variation in Comment ratio, which is fairly good considering the small number and simplistic nature of the features involved.

4 Sentiment detection: SO-CAL

In this section, we outline our semantic orientation calculator, SO-CAL. SO-CAL extracts words from a text, and aggregates their semantic orientation value, which is in turn extracted from a set of dictionaries. SO-CAL uses five dictionaries: four lexical dictionaries with 2,257 adjectives, 1,142 nouns, 903 verbs, and 745 adverbs, and a fifth dictionary containing 177 intensifying expressions. Although the majority of the entries are single words, the calculator also allows for multiword entries written in regular expression-like language.

The SO-carrying words in these dictionaries were taken from a variety of sources, the three largest a corpus of 400 reviews from Epinions.com, first used by Taboada and Grieve (2004), a 100 text subset of the 2,000 movie reviews in the Polarity Dataset (Pang and Lee, 2004), and words from the General Inquirer dictionary (Stone, 1997). Each of the open-class words were given a hand-ranked SO value between 5 and -5 (neutral or zero-value words are not included in the dictionary) by a native English speaker. The numerical values were chosen to reflect both the prior polarity and strength of the word, averaged across likely interpretations. For example, the word *phenomenal* is a 5, *nicely* a 2, *disgust* a -3, and *monstrosity* a -5. The dictionary was later reviewed by a committee of three other researchers in order to minimize the subjectivity of ranking SO by hand.

Our calculator moves beyond simple averaging of each word's semantic orientation value, and implements and expands on the insights of Polanyi and Zaenen (2006) with respect to contextual valence shifters. We implement negation by shifting the SO value of a word towards the opposite polarity (*not terrible*, for instance, is calculated as -5+4 = -1). Intensification is modeled using percentage modifiers (*very engaging*: 4x125% = 5). We also ignore words appearing within the scope of *irrealis* markers such as certain verbs, modals, and punctuation, and decrease the weight of words which appear often in the text. In order to counter positive linguistic

bias (Boucher and Osgood, 1969), a problem for lexicon-based sentiment classifiers (Kennedy and Inkpen, 2006), we increase the final SO of any negative expression appearing in the text.

The performance of SO-CAL tends to be in the 76-81% range. We have tested on informal movie, book and product reviews and on the Polarity Dataset (Pang and Lee, 2004). The performance on movie reviews tends to be on the lower end of the scale. Our baseline for movies, described in Section 5, is 77.7%. We believe that we have reached a ceiling in terms of word- and phrase-level performance, and most future improvements need to come from discourse features. The stage classification described in this paper is one of them.

5 Results

The final goal of a stage classifier is to use the information about different stages in sentiment classification. Our assumption is that descriptive paragraphs contain less evaluative content about the movie being reviewed, and they may include noise, such as evaluative words describing the plot or the characters. Once the paragraph classifier had assigned labels we used those labels to weigh paragraphs.

5.1 Classification with manual tags

Before moving on to automatic paragraph classification, we used the 100 annotated texts to see the general effect of weighting paragraphs with the "perfect" human annotated tags on sentiment detection, in order to show the potential improvements that can be gained from this approach.

Our baseline polarity detection performance on the 100 annotated texts is 65%, which is very low, even for movie reviews. We posit that formal movie reviews might be particularly difficult because full plot descriptions are more common and the language used to express opinion less straightforward (metaphors are common). However, if we lower the weight on non-Comment and mixed Comment paragraphs (to 0, except for Describe+Comment, which is maximized by a 0.1 weight), we are able to boost performance to 77%, an improvement which is significant at the $p<0.05$ level. Most of the improvement (7%) is due to disregarding Describe paragraphs, but 2% comes from Describe+Comment, and 1% each from Background, Interpretation, and (all) Formal tags. There is no performance gain, however, from the use of aspect tags (e.g., by increasing the weight on Overall paragraphs), justifying our decision to ignore subtags for text-level polarity classification.

5.2 Categorical classification

We evaluated all the classifiers from Table 2, but we omit discussion of the worst performing. The evaluation was performed on the Polarity Dataset (Pang and Lee, 2004), a collection of 2,000 online movie reviews, balanced for polarity. The SO performance for the categorical classifiers is given in Figure 1. When applicable, we always gave Formal Zones (which Table 2 indicates are fairly easy to identify) a weight of 0, however for Describe paragraphs we tested at 0.1 intervals between 0 and 1. Testing all possible values of Describe+Comment was not feasible, so we set the weights of those to a value halfway between the weight of Comment paragraphs (1) and the weight of the Describe paragraph.

Most of the classifiers were able to improve performance beyond the 77.7% (unweighted) baseline. The best performing model (the 2-class-genre-SVM) reached a polarity identification accuracy of 79.05%, while the second best (the 3-class 5-gram-SVM) topped out at 78.9%. Many of the classifiers showed a similar pattern with respect to the weight on Describe, increasing linearly as weight on Describe was decreased before hitting a maximum in the 0.4-0.1 range, and then dropping afterwards (often precipitously). Only the classifiers which were more conservative with respect to Describe, such as the 4-class-5-gram-Bayes, avoided the drop, which can be attributed to low precision Describe identification: At some point, the cost associated with disregarding paragraphs which have been mistagged as Describe becomes greater that the benefit of disregarding correctly-labeled ones. Indeed, the best performing classifier for each class option is exactly the one that has the highest precision for identification of Describe, regardless of other factors. This suggests that improving precision is key, and, in lieu of that, weighting is a better strategy than simply removing parts of the text.

In general, increasing the complexity of the task (increasing the number of classes) decreases performance. One clear problem is that the identification of Formal zones, which are much more common in our training corpus than our test corpus, does not add important information, since most Formal zones have no SO valued words. The delineation of an independent Describe+Comment class is mostly ineffective,

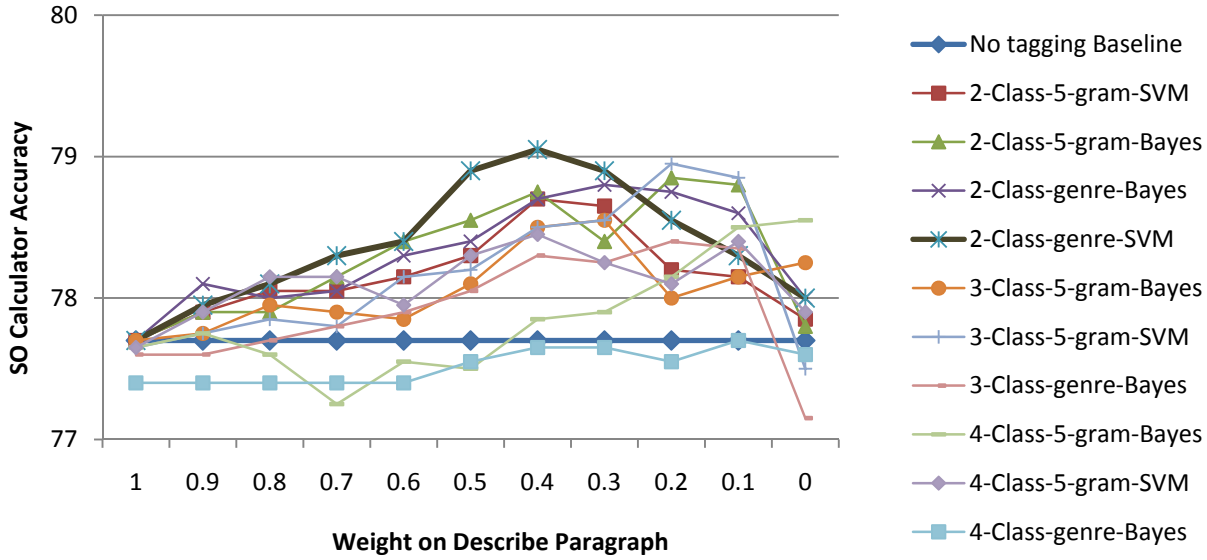


Figure 1. SO Performance with various paragraph tagging classifiers, by weight on Describe

probably because this class is not easily distinguishable from Describe and Comment (nor in fact should it be).

We can further confirm that our classifier is properly distinguishing Describe and Comment by discounting Comment paragraphs rather than Describe paragraphs (following Pang and Lee 2004). When Comment paragraphs tagged by the best performing classifier are ignored, SO-CAL's accuracy drops to 56.65%, just barely above chance.

5.3 Continuous classification

Table 4 gives the results for the linear regression classifier, which assigns a Comment ratio to each paragraph used for weighting.

Model	Accuracy
LR, Des+Com C = 0	78.75
LR, Des+Com C = 0.25	**79.35**
LR, Des+Com C = 0.5	79.00
LR, Des+Com C = 0.75	78.90
LR, Des+Com C = 1	78.95
LR, No Des+Com	79.05

Table 4. SO Performance with linear regression

The linear regression model trained with a 0.25 comment ratio on Describe+Comment paragraphs provides the best performance of all classifiers we tested (an improvement of 1.65% from baseline). The correlation coefficients noted in Table 4 are reflected in these results, but the spike at C = 0.25 is most likely related to a general preference for low (but non-zero) weights on Describe+Comment paragraphs also noted when weights were applied using the manual tags; these paragraphs are unreliable (as compared to pure Comment), but cannot be completely discounted. There were some texts which had only Describe+Comment paragraphs.

Almost a third of the tags assigned by the 2-class genre feature classifier were different than the corresponding n-gram classifier, suggesting the two classifiers might have different strengths. However, initial attempts to integrate the various high performing classifiers—including collapsing of feature sets, metaclassifiers, and double tagging of paragraphs—resulted in similar or worse performance. We have not tested all possible options (there are simply too many), but we think it unlikely that additional gains will be made with these simple, surface feature sets. Although our testing with human annotated texts and the large performance gap between movie reviews and other consumer reviews both suggest there is more potential for improvement, it will probably require more sophisticated and precise models.

6 Related work

The bulk of the work in sentiment analysis has focused on classification at either the sentence level, e.g., the subjectivity/polarity detection of Wiebe and Riloff (2005), or alternatively at the level of the entire text. With regards to the latter, two major approaches have emerged: the use of machine learning classifiers trained on n-grams

or similar features (Pang et al., 2002), and the use of sentiment dictionaries (Esuli and Sebastiani, 2006; Taboada et al., 2006). Support Vector Machine (SVM) classifiers have been shown to out-perform lexicon-based models within a single domain (Kennedy and Inkpen, 2006); however they have trouble with cross-domain tasks (Aue and Gamon, 2005), and some researchers have argued for hybrid classifiers (Andreevskaia and Bergler, 2008).

Pang and Lee (2004) attempted to improve the performance of an SVM classifier by identifying and removing objective sentences from the texts. Results were mixed: The improvement was minimal for the SVM classifier (though the performance of a naïve Bayes classifier was significantly boosted), however testing with parts of the text classified as subjective showed that the eliminated parts were indeed irrelevant. In contrast to our findings, they reported a drop in performance when paragraphs were taken as the only possible boundary between subjective and objective text spans.

Other research that has dealt with identifying more or less relevant parts of the text for the purposes of sentiment analysis include Taboada and Grieve (2004), who improved the performance of a lexicon-based model by weighing words towards the end of the text; Nigam and Hurst (2006), who detect polar expressions in topic sentences; and Voll and Taboada (2007), who used a topic classifier and discourse parser to eliminate potentially off-topic or less important sentences.

7 Conclusions

We have described a genre-based taxonomy for classifying paragraphs in movie reviews, with the main classification being a distinction between formal and functional stages, and, within those, between mainly descriptive vs. comment stages. The taxonomy was used to annotate 100 movie reviews, as the basis for building classifiers.

We tested a number of different classifiers. Our results suggest that a simple, two-way or continuous classification using a small set of linguistically-motivated features is the best for our purposes; a more complex system is feasible, but comes at the cost of precision, which seems to be the key variable in improving sentiment analysis.

Ultimately, the goal of the classification was to improve the accuracy of SO-CAL, our semantic orientation calculator. Using the manual annotations, we manage to boost performance by 12% over the baseline. With the best automatic classifier, we still show consistent improvement over the baseline. Given the relatively low accuracy of the classifiers, the crucial factor involves using fine-grained weights on paragraphs, rather than simply ignoring Describe-labeled paragraphs, as Pang and Lee (2004) did for objective sentences.

An obvious expansion to this work would involve a larger dataset on which to train, to improve the performance of the classifier(s). We would also like to focus on the syntactic patterns and verb class properties of narration, aspects that are not captured with simply using words and POS labels. Connectives in particular are good indicators of the difference between narration (temporal connectives) and opinion (contrastive connectives). There may also be benefit to combining paragraph- and sentence-based approaches. Finally, we would like to identify common sequences of stages, such as plot and character descriptions appearing together, and before evaluation stages. This generic structure has been extensively studied for many genres (Eggins and Slade, 1997).

Beyond sentiment extraction, our taxonomy and classifiers can be used for searching and information retrieval. One could, for instance, extract paragraphs that include mostly comment or description. Using the more fine-grained labels, searches for comment/description on actors, directors, or other aspects of the movie are possible.

Acknowledgements

This work was supported by SSHRC (410-2006-1009) and NSERC (261104-2008) grants to Maite Taboada.

References

Andreevskaia, Alina & Sabine Bergler. 2008. When specialists and generalists work together: Domain dependence in sentiment tagging. Proceedings of 46th Annual Meeting of the Association for Computational Linguistics (pp. 290-298). Columbus, OH.

Aue, Anthony & Michael Gamon. 2005. Customizing sentiment classifiers to new domains: A case study. Proceedings of the International Conference on Recent Advances in Natural Language Processing. Borovets, Bulgaria.

Biber, Douglas. 1988. Variation across Speech and Writing. Cambridge: Cambridge University Press.

Bieler, Heike, Stefanie Dipper & Manfred Stede. 2007. Identifying formal and functional zones in film reviews. Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue (pp. 75-78). Antwerp, Belgium.

Boucher, Jerry D. & Charles E. Osgood. 1969. The Pollyanna hypothesis. Journal of Verbal Learning and Verbal Behaviour, 8: 1-8.

Di Eugenio, Barbara & Michael Glass. 2004. The kappa statistic: A second look. Computational Linguistics, 30(1): 95-101.

Eggins, Suzanne & James R. Martin. 1997. Genres and registers of discourse. In Teun A. van Dijk (ed.), Discourse as Structure and Process. Discourse Studies: A Multidisciplinary Introduction (pp. 230-256). London: Sage.

Eggins, Suzanne & Diana Slade. 1997. Analysing Casual Conversation. London: Cassell.

Esuli, Andrea & Fabrizio Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. Proceedings of 5th International Conference on Language Resources and Evaluation (LREC) (pp. 417-422). Genoa, Italy.

Fleiss, Joseph L. 1971. Measuring nominal scale agreement among many raters. Psychological Bulletin, 76: 378-382.

Kennedy, Alistair & Diana Inkpen. 2006. Sentiment classification of movie and product reviews using contextual valence shifters. Computational Intelligence, 22(2): 110-125.

Knott, Alistair. 1996. A Data-Driven Methodology for Motivating a Set of Coherence Relations. Edinburgh, UK: University of EdinburghThesis Type.

Martin, James R. & Peter White. 2005. The Language of Evaluation. New York: Palgrave.

Nigam, Kamal & Matthew Hurst. 2006. Towards a robust metric of polarity. In Janyce Wiebe (ed.), Computing Attitude and Affect in Text: Theory and Applications (pp. 265-279). Dordrecht: Springer.

Orasan, Constantin. 2003. PALinkA: A highly customizable tool for discourse annotation. Proceedings of 4th SIGdial Workshop on Discourse and Dialog (pp. 39 – 43). Sapporo, Japan.

Pang, Bo & Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Proceedings of 42nd Meeting of the Association for Computational Linguistics (pp. 271-278). Barcelona, Spain.

Pang, Bo, Lillian Lee & Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using Machine Learning techniques. Proceedings of Conference on Empirical Methods in NLP (pp. 79-86).

Polanyi, Livia & Annie Zaenen. 2006. Contextual valence shifters. In James G. Shanahan, Yan Qu & Janyce Wiebe (eds.), Computing Attitude and Affect in Text: Theory and Applications (pp. 1-10). Dordrecht: Springer.

Seki, Yohei, Koji Eguchi & Noriko Kando. 2006. Multi-document viewpoint summarization focused on facts, opinion and knowledge. In Janyce Wiebe (ed.), Computing Attitude and Affect in Text: Theory and Applications (pp. 317-336). Dordrecht: Springer.

Stone, Philip J. 1997. Thematic text analysis: New agendas for analyzing text content. In Carl Roberts (ed.), Text Analysis for the Social Sciences. Mahwah, NJ: Lawrence Erlbaum.

Taboada, Maite, Caroline Anthony & Kimberly Voll. 2006. Creating semantic orientation dictionaries. Proceedings of 5th International Conference on Language Resources and Evaluation (LREC) (pp. 427-432). Genoa, Italy.

Taboada, Maite & Jack Grieve. 2004. Analyzing appraisal automatically. Proceedings of AAAI Spring Symposium on Exploring Attitude and Affect in Text (AAAI Technical Report SS-04-07) (pp. 158-161). Stanford University, CA.

Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proceedings of 40th Meeting of the Association for Computational Linguistics (pp. 417-424).

Voll, Kimberly & Maite Taboada. 2007. Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. Proceedings of the 20th Australian Joint Conference on Artificial Intelligence (pp. 337-346). Gold Coast, Australia.

Whitelaw, Casey, Navendu Garg & Shlomo Argamon. 2005. Using Appraisal groups for sentiment analysis. Proceedings of ACM SIGIR Conference on Information and Knowledge Management (CIKM 2005) (pp. 625-631). Bremen, Germany.

Wiebe, Janyce & Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. Proceedings of Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005). Mexico City, Mexico.

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

Appendix A: Full lists of formal and functional zones

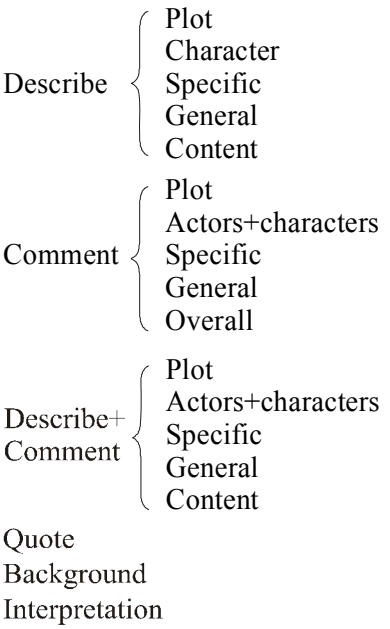


Figure A1. Functional zones

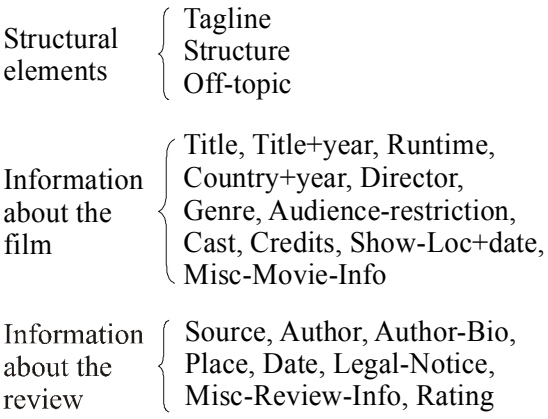


Figure A2. Formal zones

Appendix B: Kappa values for annotation task

Classes	**2-rater kappa**	**3-rater kappa**
Describe/Comment/Describe+Comment/Formal	.82	.73
Describe/Comment/Formal	.92	.84
Describe/Comment/Describe+Comment	.68	.54
Describe/Comment	.84	.69

Table B1. Kappa values for stage annotations

Detecting the Noteworthiness of Utterances in Human Meetings

Satanjeev Banerjee
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
banerjee@cs.cmu.edu

Alexander I. Rudnicky
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
air@cs.cmu.edu

Abstract

Our goal is to make note-taking easier in meetings by automatically detecting noteworthy utterances in verbal exchanges and suggesting them to meeting participants for inclusion in their notes. To show feasibility of such a process we conducted a Wizard of Oz study where the Wizard picked automatically transcribed utterances that he judged as noteworthy, and suggested their contents to the participants as notes. Over 9 meetings, participants accepted 35% of these suggestions. Further, 41.5% of their notes at the end of the meeting contained Wizard-suggested text. Next, in order to perform noteworthiness detection automatically, we annotated a set of 6 meetings with a 3-level noteworthiness annotation scheme, which is a break from the binary "in summary"/ "not in summary" labeling typically used in speech summarization. We report Kappa of 0.44 for the 3-way classification, and 0.58 when two of the 3 labels are merged into one. Finally, we trained an SVM classifier on this annotated data; this classifier's performance lies between that of trivial baselines and inter-annotator agreement.

1 Introduction

We regularly exchange information verbally with others over the course of meetings. Often we need to access this information afterwards. Typically we record the information we consider important by taking notes. Note taking at meetings is a difficult task, however, because the participant must summarize and write down the information in a way such that it is comprehensible afterwards, while paying attention to and participating in the ongoing discussion. Our goal is to make note-taking easier by automatically extracting noteworthy items from spoken interactions in real time, and proposing them to the humans for inclusion in their notes.

Judging which pieces of information in a meeting are noteworthy is a very subjective task. The subjectivity of this task is likely to be more acute than even that of meeting summarization, where low inter-annotator agreement is typical e.g. (Galley, 2006), (Liu & Liu, 2008), (Penn & Zhu, 2008), etc – whether a piece of information should be included in a participant's notes depends not only on its importance, but also on factors such as the participant's need to remember, his perceived likelihood of forgetting, etc. To investigate whether it is feasible even for a human to predict what someone else might find noteworthy in a meeting, we conducted a Wizard of Oz-based user study where a human suggested notes (with restriction) to meeting participants during the meeting. We concluded from this study (presented in section 2) that this task appears to be feasible for humans.

Assuming feasibility, we then annotated 6 meetings with a 3-level noteworthiness scheme. Having 3 levels instead of the typical 2 allows us to explicitly separate utterances of middling noteworthiness from those that are definitely noteworthy or not noteworthy, and allows us to encode more human knowledge than a 2-level scheme. We describe this annotation scheme in more detail in section 3, and show high inter-annotator agreement compared to that typically reported in the summarization literature. Finally in sections 4 and 5 we use this annotated data to train and test a simple Support Vector Machine-based predictor of utterance noteworthiness.

2 Can Humans Do this Task?

As mentioned in the introduction, given the degree of subjectivity involved in identifying note-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 71–78,
Queen Mary University of London, September 2009.

worthy utterances, it is reasonable to ask whether the notes-suggestion task can be accomplished by humans, let alone by automatic systems. That is, we ask the question: Is it possible for a human to identify noteworthy utterances in a meeting such that

(a) For at least some fraction of the suggestions, one or more meeting participants agree that the suggested notes should indeed be included in their notes, and
(b) The fraction of suggested notes that meeting participants find noteworthy is high enough that, over a sequence of meetings, the meeting participants do not learn to simply ignore the suggestions.

Observe that this task is more restricted than that of generic note-taking. While a human who is allowed to summarize discussions and produce to-the-point notes is likely to be useful, we assume here that our system will not be able to create such abstractive summaries. Rather, our goal here is to explore the feasibility of an extractive summarization system that simply picks noteworthy utterances and suggests their contents to the participants. To answer this question, we conducted a Wizard of Oz-based pilot user study, as follows.

2.1 Wizard of Oz Study Design

We designed a user study in which a human Wizard listened to the utterances being uttered during the meeting, identified noteworthy utterances, and suggested their contents to one or more participants for inclusion in their notes. In order to minimize differences between the Wizard and the system (except for the Wizard's human-level ability to judge noteworthiness), we restricted the Wizard in the following ways:

(a) The Wizard was allowed to only suggest the contents of individual utterances to the participants, and not summarize the contents of multiple utterances.
(b) The Wizard was allowed to listen to the meeting speech, but when suggesting the contents of an utterance to the participants, he was restricted to using a real-time automatic transcription of the utterance. (He was allowed to withhold suggestions because they were too erroneously transcribed.)
(c) In order to be closer to a system that has little or no "understanding" of the meetings, we chose a human (to play the role of the Wizard) who had not participated in the meetings before, and thus had little prior knowledge of the meetings' contents.

2.2 Notes Suggestion Interface

In order to suggest notes to meeting participants during a meeting – either automatically or through a Wizard – we have modified the SmartNotes system, whose meeting recording and note-taking features have been described earlier in (Banerjee & Rudnicky, 2007). Briefly, each meeting participant comes to the meeting with a laptop running SmartNotes. At the beginning of the meeting, each participant's SmartNotes client connects to a server, authenticates the participant and starts recording and transmitting his speech to the server. In addition, SmartNotes also provides meeting participants with a note-taking interface that is split into two major panes. In the "notes" pane the participant types his notes that are then recorded for research purposes. In the "suggestions" pane, Wizard-suggested notes are displayed. If at any time during the meeting a participant double-clicks on one of the suggested notes in the "suggestions" pane, its text gets included in his notes in the "notes" pane. The Wizard uses a different application to select real-time utterance transcriptions, and insert them into each participant's "suggestions" pane. (While we also experimented with having the Wizard target his suggestions at individual participants, we do not report on those experiments here; those results were similar to the ones presented below.)

2.3 Results

We conducted the Wizard of Oz study on 9 meetings that all belonged to the same sequence. That is, these meetings featured a largely overlapping group of participants who met weekly to discuss progress on a single project. The same person played the role of the Wizard in each of these 9 meetings. The meetings were on average 33 minutes long, and there were 3 to 4 participants in each meeting. Although we have not evaluated the accuracy of the speech recognizer on these particular meetings, the typical average word error rate for these speakers is around 0.4 – i.e., 4 out of 10 words are incorrectly transcribed.

On average, the Wizard suggested the contents of 7 utterances to the meeting participants, for a total of 63 suggestions across the 9 meetings. Of these 63 suggestions, 22 (34.9%) were accepted by the participants and included in their notes. Thus on average, about 2.5 Wizard-suggested notes were accepted and included in participants' notes in each meeting. On average, meeting participants took a total of 5.9 lines of notes per

meeting; thus, 41.5% of the notes in each meeting were Wizard-suggested.

It cannot be ascertained if the meeting participants would have written the suggested notes on their own if they weren't suggested to them. However the fact that some Wizard-suggested notes *were* accepted implies that the participants probably saw some value in including those suggestions in their notes. Further, there was no drop-off in the fraction of meeting notes that was Wizard-suggested: the per-meeting average percentage of notes that was Wizard-suggested was around 41% for both the first 4 meetings, as well as the last 5. This implies that despite a seemingly low acceptance rate (35%), participants did not "give up" on the suggestions, but continued to make use of them over the course of the 9-meeting meeting sequence. We conclude that an extractive summarization system that detects noteworthy utterances and suggests them to meeting participants can be perceived as useful by the participants, if the detection of noteworthy utterances is "accurate enough".

3 Meeting Data Used in this Paper

Assuming the feasibility of an extraction-based notes suggestion system, we turn our attention to developing a system that can automatically detect the noteworthiness of an utterance. Our goal here is to learn to do this task over a sequence of related meetings. Towards this end, we have recorded sequences of natural meetings – meetings that would have taken place even if they weren't being recorded. Meetings in each sequence featured largely overlapping participant sets and topics of discussion. For each meeting, we used SmartNotes (Banerjee & Rudnicky, 2007) (described in section 2 above) to record both the audio from each participant as well as his notes. The audio recording and the notes were both time stamped, associated with the participant's identity, and uploaded to the meeting server. After the meeting was completed the audio was manually segmented into utterances and transcribed both manually and using a speech recognizer (more details in section 5.2).

In this paper we use a single sequence of 6 meetings held between April and June of 2006. (These were separate from the ones used for the Wizard of Oz study above.) The meetings were on average 28 minutes and 43 seconds long (± 3 minutes and 48 seconds standard error) counting from the beginning of the first recorded utterance to the end of the last one. On average each meeting had 28 minutes and 38 seconds of speech – this includes overlapped speech when multiple participants spoke on top of each other. Across the 6 meetings there were 5 unique participants; each meeting featured between 2 and 4 of these participants (average: 3.5 ± 0.31).

The meetings had, on average, 633.67 (± 85.60) utterances each, for a total of 3,796 utterances across the 6 meetings. (In this paper, these 3,796 utterances form the units of classification.) As expected, utterances varied widely in length. On average, utterances were 2.67 ± 0.18 seconds long and contained 7.73 (± 0.44) words.

4 Multilevel Noteworthiness Annotation

In order to develop approaches to automatically identify noteworthy utterances, we have manually annotated each utterance in the meeting data with its degree of "noteworthiness". While researchers in the related field of speech summarization typically use a binary labeling – "in summary" versus "out of summary" (e.g. (Galley, 2006), (Liu & Liu, 2008), (Penn & Zhu, 2008), etc) – we have observed that there are often many utterances that are "borderline" at best, and the decision to label them as "in summary" or "out" is arbitrary. Our approach instead has been to create three levels of noteworthiness. Doing so allows us to separate the "clearly noteworthy" utterances from the "clearly not noteworthy", and to label the rest as being between these two classes. (Of course, arbitrary choices must still be made between the edges of these three classes. However, having three levels preserves more information in the labels than having two, and it is always possible to create two labels from the three, as we do in later sections.)

These multilevel noteworthiness annotations were done by two annotators. One of them – denoted as "annotator 1" – had attended each of the meetings, while the other – "annotator 2" – had not attended any of the meetings. Although annotator 2 was given a brief overview of the general contents of the meetings, his understanding of the meeting was expected to be lower than that of the other annotator. By using such an annotator, our aim was to identify utterances that were "obviously noteworthy" even to a human being who lacks a deep understanding of the context of the meetings. (In section 5.2 we describe how we merge the two sets of annotations.)

The annotators were asked to make a 3-level judgment about the relative noteworthiness of each utterance. That is, for each utterance, the

annotators were asked to decide whether a note-suggestion system should "definitely show" the contents of the utterance to the meeting participants, or definitely not show (labeled as "don't show"). Utterances that did not quite belong to either category were asked to be labeled as "maybe show". Utterances labeled "definitely show" were thus at the highest level of noteworthiness, followed by those labeled "maybe show" and those labeled "don't show". Note that we did not ask the annotators to label utterances directly in terms of noteworthiness. Anecdotally, we have observed that asking people to label utterances with their noteworthiness leaves the task insufficiently well defined because the purpose of the labels is unclear. On the other hand, asking users to identify utterances they would have included in their notes leads to annotators taking into account the difficulty of writing particular notes, which is also not desirable for this set of labels. Instead, we asked annotators to directly perform (in some sense) the task that the eventual notes-assistance system will perform.

In order to gain a modicum of agreement in the annotations, the two annotators discussed their annotation strategies after annotating each of the first two meetings (but not after the later meetings). A few general annotation patterns emerged, as follows: Utterances labeled "definitely show" typically included:

(a) Progress on action items since the last week.
(b) Concrete plans of action for the next week.
(c) Announcements of deadlines.
(d) Announcements of bugs in software, etc.

In addition, utterances that contained the *crux* of any seemingly important discussion were labeled as "definitely show". On the other hand, utterances that contained no information worth including in the notes (by the annotators' judgment) were labeled as "don't show". Utterances that did contain some additional elaborations of the main point, but without which the main point could still be understood by future readers of the notes were typically labeled as "maybe show".

Table **1** shows the distribution of the three labels across the full set of 3,796 utterances in the dataset for both annotators. Both annotators labeled only a small percentage of utterances as "definitely show", a larger fraction as "maybe show" and most utterances as "don't show". Although the annotators were not asked to shoot for a certain distribution, observe that they both labeled a similar fraction of utterances as "definitely show". On the other hand, annotator 2, who did not attend the meetings, labeled 50% more utterances as "maybe show" than annotator 1 who did attend the meetings. This difference is likely due to the fact that annotator 1 had a better understanding of the utterances in the meeting, and was more confident in labeling utterances as "don't show" than annotator 2 who, not having attended the meetings, was less sure of some utterances, and thus more inclined to label them as "maybe show".

Annotator #	Definitely show	Maybe show	Don't show
1	13.5%	24.4%	62.1%
2	14.9%	38.8%	46.3%

Table 1: Distribution of Labels for Each Annotator

4.1 Inter-Annotator Kappa Agreement

To gauge the level of agreement between the two annotators, we compute the Kappa score. Given labels from different annotators on the same data, this metric quantifies the difference between the observed agreement between the labels and the expected agreement, with larger values denoting stronger agreement.

For the 3-way labeling task, the two annotators achieve a Kappa agreement score of 0.44 (± 0.04). This seemingly low number is typical of agreement scores obtained in meeting summarization. (Liu & Liu, 2008) reported Kappa agreement scores between 0.11 and 0.35 across 6 annotators while (Penn & Zhu, 2008) with 3 annotators achieved Kappa of 0.383 and 0.372 on casual telephone conversations and lecture speech. (Galley, 2006) reported inter-annotator agreement of 0.323 on data similar to ours.

To further understand where the disagreements lie, we converted the 3-way labeled data into 2 different 2-way labeled datasets by merging two labels into one. First we evaluate the degree of agreement the annotators have in separating utterances labeled "definitely show" from the other two levels. We do so by re-labeling all utterances not labeled "definitely show" with the label "others". For the "definitely show" versus "others" labeling task, the annotators achieve an inter-annotator agreement of 0.46. Similarly we compute the agreement in separating utterances labeled "do not show" from the two other labels – in this case the Kappa value is 0.58. This implies that it is easier to agree on the separation between "do not show" and the other classes, than between "definitely show" and the other classes.

4.2 Inter-Annotator Accuracy, Prec/Rec/F

Another way to gauge the agreement between the two sets of annotations is to compute accuracy, precision, recall and f-measure between them. That is, we can designate one annotator's labels as the "gold standard", and use the other annotator's labels to find, for each of the 3 labels, the number of utterances that are true positives, false positives, and false negatives. Using these numbers we can compute precision as the ratio of true positives to the sum of true and false positives, recall as the ratio of true positives to the sum of true positives and false negatives, and f-measure as the harmonic mean of precision and recall. (Designating the other annotator's labels as "gold standard" simply swaps the precision and recall values, and keeps f-measure the same). Accuracy is the number of utterances that have the same label from the two annotators, divided by the total number of utterances.

Table **2** shows the evaluation over the 6-meeting dataset using annotator 1's data as "gold standard". The standard error for each cell is less than 0.08. Observe in Table **2** that while both the "definitely show" and "maybe show" classes have nearly equal f-measure, the precision and recall values for the "maybe show" class are much farther apart from each other than those for the "definitely show" class. This is due to the fact that while both annotators label a similar number of utterances as "definitely show", they label very different numbers of utterances as "maybe show". If the same accuracy, precision, recall and f-measure scores are computed for the "definitely show" vs. "others" split, the accuracy jumps to 87%, possibly because of the small size of the "definitely show" category. The accuracy remains at 78% for the "don't show" vs. "others" split.

	Definitely show	Maybe show	Don't show
Precision	0.57	0.70	0.70
Recall	0.53	0.46	0.93
F-measure	0.53	0.54	0.80
Accuracy	69%		

Table 2 Inter-Annotator Agreement using Accuracy Etc.

4.3 Inter-Annotator Rouge Scores

Annotations can also be evaluated by computing the ROUGE metric (Lin, 2004). ROUGE, a popular metric for summarization tasks, compares two summaries by computing precision, recall and f-measure over ngrams that overlap between them. Following previous work on meeting summarization (e.g. (Xie, Liu, & Lin, 2008), (Murray, Renals, & Carletta, 2005), etc), we report evaluation using ROUGE-1 F-measure, where the value "1" implies that overlapping *unigrams* are used to compute the metric. Unlike previous research that had one summary from each annotator per meeting, our 3-level annotation allows us to have 2 different summaries: (a) the text of all the utterances labeled "definitely show" and, (b) the text of all the utterances labeled either "definitely show" or "maybe show". On average (across both annotators over the 6 meetings) the "definitely show" utterance texts are 18.72% the size of the texts of all the utterances in the meetings, while the "definitely or maybe show" utterance texts are 61.6%. Thus, these two texts represent two distinct points on the compression scale. The average R1 F-measure score is 0.62 over the 6 meetings when comparing the "definitely show" texts of the two annotators. This is twice the R1 score – 0.3 – of the trivial baseline of simply labeling every utterance as "definitely show". The inter-annotator R1 F-measure for the "definitely or maybe show" texts is 0.79, marginally higher than the trivial "all utterances" baseline of 0.71. In the next section, we compare the scores achieved by the automatic system against these inter-annotator and trivial baseline scores.

5 Automatic Label Prediction

So far we have presented the annotation of the meeting data, and various analyses thereof. In this section we present our approach for the automatic prediction of these labels. We apply a classification based approach to the problem of predicting the noteworthiness level of an utterance, similar to (Banerjee & Rudnicky, 2008). We use leave-one-meeting-out cross validation: for each meeting *m*, we train the classifier on manually labeled utterances from the other 5 meetings, and test the classifier on the utterances of meeting *m.* We then average the results across the 6 meetings. Given the small amount of data, we do not test on separate data, nor do we perform any tuning.

Using the 3-level annotation described above, we train a 3-way classifier to label each utterance with one of the multilevel noteworthiness labels. In addition, we use the two 2-way merged-label annotations – "definitely show" vs. others and "don't show" vs. others – to train two more 2-way classifiers. In each of these classification

problems we use the same set of features and the same classification algorithms described below.

5.1 Features Used

Ngram features: As has been shown by (Banerjee & Rudnicky, 2008), the strongest features for noteworthiness detection are ngram features, i.e. features that capture the occurrence of ngrams (consecutive occurrences of one or more words) in utterances. Each ngram feature represents the presence or absence of a single specific ngram in an utterance. E.g., the ngram feature "action item" represents the occurrence of the bigram "action item" in a given utterance. Unlike (Banerjee & Rudnicky, 2008) where each ngram feature captured the *frequency* of a specific ngram in an utterance, in this paper we use boolean-valued ngram features to capture the presence/absence of ngrams in utterances. We do so because in tests on separate data, boolean-valued features out-performed frequency-based features, perhaps due to data sparseness. Before ngram features are extracted, utterances are normalized: partial words, non-lexicalized filler words (like "umm", "uh"), punctuations, apostrophes and hyphens are removed, and all remaining words are changed to upper case. Next, the vocabulary of ngrams is defined as the set of ngrams that occur at least 5 times in the entire dataset of meetings, for ngram sizes of 1 through 6 word tokens. Finally, the occurrences of each of these vocabulary ngrams in an utterance are recorded as the feature vector for that utterance. In the dataset used in this paper, there are 694 unique unigrams that occur at least 5 times across the 6 meetings, 1,582 bigrams, 1,065 trigrams, 1,048 4-grams, 319 5-grams and 102 6-grams. In addition to these ngram features, for each utterance we also include the number of Out of Vocabulary ngram – ngrams that occur less than 5 times across all the meetings.

Overlap-based Features: We assume that we have access to the text of the agenda of the test meeting, and also the text of the notes taken by the participants in previous meetings (but not those taken in the test meeting). Since these artifacts are likely to contain important keywords we compute two sets of overlaps features. In the first set we compute the number of ngrams that overlap between each utterance and the meeting agenda. That is, for each utterance we count the number of unigrams, bigrams, trigrams, etc that also occur in the agenda of that meeting. Similarly in the second set we compute the number of ngrams in each utterance that also occur in the notes of previous meetings. Finally, we compute the degree of overlap between this utterance and other utterances in the meeting. The motivation for this last feature is to find utterances that are repeats (or near-repeats) of other utterances – repetition may correlate with importance.

Other features: In addition to the ngram and ngram overlap features, we also include term frequency – inverse document frequency (tf-idf) features to capture the information content of the ngrams in the utterance. Specifically we compute the TF-IDF of each ngram (of sizes 1 through 5) in the utterance, and include the maximum, minimum, average and standard deviation of these values as features of the utterance. We also include speaker-based features to capture who is speaking when. We include the identity of the speaker of the current utterance and those of the previous and next utterances as features. Lastly we include the length of the utterance (in seconds) as a feature.

5.2 Evaluation Results

In this paper we use a Support Vector Machines-based classifier, which is a popular choice for extractive meeting summarization, e.g. (Xie, Liu, & Lin, 2008); we use a linear kernel in this paper. In the results reported here we use the output of the Sphinx speech recognizer, using speaker-independent acoustic models, and language models trained on publicly available meeting data. The word error rate was around 44% – more details of the speech recognition process are in (Huggins-Daines & Rudnicky, 2007). For training purposes, we merged the annotations from the two annotators by choosing a "middle or lower ground" for all disagreements. Thus, if for an utterance the two labels are "definitely show" and "don't show", we set the merged label as the middle ground of "maybe show". On the other hand if the two labels were on adjacent levels, we chose the lower one – "maybe show" when the labels were "definitely show" and "maybe show", and "don't show" when the labels were "maybe show" and "don't show". Thus only utterances that *both* annotators labeled as "definitely show" were also labeled as "definitely show" in the merged annotation. We plan to try other merging strategies in the future. For testing, we evaluated against each annotator's labels separately, and averaged the results.

	Definitely show	Maybe show	Don't show
Precision	0.21	0.47	0.72
Recall	0.16	0.40	0.79
F-measure	0.16	0.43	0.75
Accuracy	61.4%		

Table 3 Results of the 3-Way Classification

Table **3** presents the accuracy, precision, recall and f-measure results of the 3-way classification task. (We use the Weka implementation of SVM that internally devolves the 3-way classification task into a sequence of pair-wise classifications. We use the final per-utterance classification here.) Observe that the overall accuracy of 61.4% is only 11% lower relative to the accuracy obtained by comparing the two annotators' annotations (69%, Table **2**). However, the precision, recall and f-measure values for the "definitely show" class are substantially lower for the predicted labels than the agreement between the two annotators. The numbers are closer for the "maybe show" and the "don't show" classes. This implies that it is more difficult to accurately detect utterances labeled "definitely show" than it is to detect the other classes. One reason for this difference is the size of each utterance class. Utterances labeled "definitely show" are only around 14% of all utterances, thus there is less data for this class than the others. We also ran the algorithm using manually transcribed data, and found improvement in only the "Definitely show" class with an f-measure of 0.21. This improvement is perhaps because the speech recognizer is particularly prone to getting names and other technical terms wrong, which may be important clues of noteworthiness.

Table **4** presents the ROUGE-1 F-measure scores averaged over the 6 meetings. (ROUGE is described briefly in section 4.3 and in detail in (Lin, 2004)). Similar to the inter-annotator agreement computations, we computed ROUGE between the text of the utterances labeled "definitely show" by the system against that of utterances labeled "definitely show" by the two annotators. (We computed the scores separately against each of the annotators in turn and then averaged the two values.) We did the same thing for the set of utterances labeled either "definitely show" or "maybe show". Observe that the R1-F score for the "definitely show" comparison is nearly 50% relative higher than the trivial baseline of labeling every utterance as "definitely show". However the score is 30% lower than the corresponding inter-annotator agreement. The corresponding R1-Fmeasure score using manual transcriptions is only marginally better – 0.47. The set of utterances labeled either definitely or maybe shows (second row of table 4) does not outperform the all-utterances baseline when using automatic transcriptions, but does so with manual transcriptions, whose R1-F value is 0.74.

Comparing What	R1-Fmeasure
Definitely show	0.43
Definitely or maybe show	0.63

Table 4 ROUGE Scores for the 3-Way Classification

These results show that while the detection of definitely show utterances is better than the trivial baselines even when using automatic transcriptions, there is a lot of room for improvement, as compared to human-human agreement. Although direct comparisons to other results from the meeting summarization literature are difficult because of the difference in the datasets, numerically it appears that our results are similar to those obtained previously. (Xie, Liu, & Lin, 2008) uses Rouge-1 F-measure solely, and achieve scores between 0.6 to 0.7. (Murray, Renals, & Carletta, 2005) also achieve Rouge-1 scores in the same range with manual transcripts.

The trend in the results for the two 2-way classifications is similar to the trend for the inter annotator agreements. Just as inter-annotator accuracy increased to 87% for the "definitely show" vs. "others" classification, so does accuracy of the predicted labels increase to 88.3%. The f-measure for the "definitely show" class falls to 0.13, much lower than the inter-annotator f-measure of 0.53. For the "don't show" vs. "others" classification, the automatic system achieves an accuracy of 66.6%. For the "definitely plus maybe" class, the f-measure is 0.59, which is 22% relatively lower than the inter-annotator f-measure for that class. (As with the 3-way classification, these results are all slightly worse than those obtained using manual transcriptions.)

5.3 Useful Features

In order to understand which features contribute most to these results, we used the Chi-Squared test of association to find features that are most strongly correlated to the 3 output classes. The best features are those that measure word overlaps between the utterances and the text in the agenda labels and the notes in previous meetings. This is not a surprising finding – the occurrence of an ngram in an agenda label or in a previous note is highly indicative of its importance, and

consequently that of the utterances that contain that ngram. Max and average TF-IDF scores are also highly ranked features. These features score highly for utterances with seldom-used words, signifying the importance of those utterances. Domain independent ngrams such as "action item" are strongly correlated with noteworthiness, as are a few domain *dependent* ngrams such as "time shift problem". These latter features represent knowledge that is transferred from earlier meetings to latter ones in the same sequence. The identity of the speaker of the utterance does not seem to correlate well with the utterance's noteworthiness, although this finding could simply be an artifact of this particular dataset.

6 Related Work

Noteworthiness detection is closely related to meeting summarization. Extractive techniques are popular, e.g. (Murray, Renals, & Carletta, 2005), and many algorithms have been attempted including SVMs (Xie, Liu, & Lin, 2008), Gaussian Mixture Models and Maximal Marginal Relevance (Murray, Renals, & Carletta, 2005), and sequence labelers (Galley, 2006). Most approaches use a mixture of ngram features, and other structural and semantic features – a good evaluation of typical features can be found in (Xie, Liu, & Lin, 2008). Different evaluation techniques have also been tried, with ROUGE often being shown as at least adequate (Liu & Liu, 2008). Our work is an application and extension of the speech summarization field to the problem of assistive note-taking.

7 Conclusions and Future Work

In our work we investigated the problem of detecting the noteworthiness of utterances produced in meetings. We conducted a Wizard-of-Oz-based user study to establish the usefulness of extracting the text of utterances and suggesting these as notes to the meeting participants. We showed that participants were willing to accept about 35% of these suggestions over a sequence of 9 meetings. We then presented a 3-level noteworthiness annotation scheme that breaks with the tradition of 2-way "in/out of summary" annotation. We showed that annotators have strong agreement for separating the highest level of noteworthiness from the other levels. Finally we used these annotations as labeled data to train a Support Vector Machine-based classifier which performed better than trivial baselines but not as well as inter-annotator agreement levels.

For future work, we plan to use automatic noteworthiness predictions to suggest notes to meeting participants during meetings. We are also interested in training the noteworthiness detector directly from the notes that participants took in previous meetings, thus reducing the need for manually annotated data.

Reference

Banerjee, S, and A. I. Rudnicky. "Segmenting Meetings into Agenda Items by Extracting Implicit Supervision from Human Note-Taking." Proceedings of the International Conference on Intelligent User Interfaces. Honolulu, HI, 2007.

Banerjee, Satanjeev, and A. I. Rudnicky. "An Extractive-Summarization Baseline for the Automatic Detection of Noteworthy Utterances in Multi-Party Human-Human Dialog." IEEE Workshop on Spoken Language Technology. Goa, India, 2008.

Galley, Michel. "A Skip-Chain Conditional Random Field for Ranking Meeting Utterances by Importance." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Sydney, Australia, 2006.

Huggins-Daines, David, and A. I. Rudnicky. "Implicitly Supervised Language Model Adaptation for Meeting Transcription." Proceedings of the HLT-NAACL. Rochester, NY. 2007.

Lin, Chin-Yew. "ROUGE: A Package for Automatic Evaluation of Summaries." Proceedings of the ACL-04 Workshop: Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics, 2004. 74-81.

Liu, Feifan, and Y. Liu. "Correlation between ROUGE and Human Evaluation of Extractive Meeting Summaries." Proceedings of ACL-HLT. Columbus, OH, 2008.

Murray, Gabriel, S. Renals, and J. Carletta. "Extractive Summarization of Meeting Recordings." Proceedings of Interspeech. Lisbon, Portugal, 2005.

Penn, Gerald, and X. Zhu. "A Critical Reassessment of Evaluation Baselines for Speech Summarization." Proceedings of ACL-HLT. Columbus, OH, 2008.

Xie, Shasha, Y. Liu, and H. Lin. "Evaluating the Effectiveness of Features and Sampling in Extractive Meeting Summarization." IEEE Workshop on Spoken Language Technology. Goa, India, 2008.

A: An Experimental Investigation into...
B: ...Split Utterances

Christine Howes, Patrick G.T. Healey and Gregory J. Mills
Queen Mary University of London
Interaction, Media and Communication Research Group, London, E1 4NS
{chrizba, ph, gj}@dcs.qmul.ac.uk

Abstract

A distinguishing feature of dialogue is that more that one person can contribute to the production of an utterance. However, until recently these 'split' utterances have received relatively little attention in models of dialogue processing or of dialogue structure. Here we report an experiment that tests the effects of artificially introduced speaker switches on groups of people engaged in a task-oriented dialogue. The results show that splits have reliable effects on response time and on the number of edits involved in formulating subsequent turns. In particular we show that if the second half of an utterance is 'misattributed' people take longer to respond to it. We also show that responses to utterances that are split across speakers involve fewer deletes. We argue that these effects provide evidence that: a) speaker switches affect processing where they interfere with expectations about who will speak next and b) that the pragmatic effect of a split is to suggest to other participants the formation of a coalition or sub-'party'.

1 Introduction

Split utterances, defined simply as utterances which are split between speakers[1], are known to occur in dialogue, as evidenced by Conversational Analysis (CA) studies, based on the analysis of naturally occuring dialogues. In addition to numerous analyses of split utterances in generic English dialogues, there are cross linguistic studies, and observations of conversations with aphasics. In Finnish, split utterances within a single clause conform to the strict syntactic constraints of the language (which has a rich inflectional morphology), despite the change in speaker (Helasvuo, 2004). Similarly, in Japanese, a verb-final language, speakers also engage in "co-participant completions" (Hayashi, 1999; Lerner and Takagi, 1999). There is also evidence of split utterances in conversations with aphasics (Oelschlaeger and Damico, 1998), demonstrating that the phenomenon is pervasive in dialogue. However, with the possible exception of Szczepek (2000) who analysed some 200 splits from 40 hours of recorded English conversation, these studies tend to be unconcerned with frequencies of occurrence; that split utterances occur at all renders them worthy of study.

Split utterances are a clear and canonical example of coordination in dialogue. In order for one person to continue an utterance which has been begun by another person requires the hearer to have coordinated with the initial speaker up to the point at which they take over the role of producer[2].

Analysis of split utterances, when they can or cannot occur and what effects they have on the coordination of agents in dialogue, is therefore an area of interest not only for conversational analysts wishing to characterise sytematic interactions in dialogue, but also linguists trying to formulate grammars of dialogue, and psychologists interested in alignment mechanisms in dialogue.

[1] What we call split utterances have been variously referred to as *collaborative turn sequences* (Lerner, 1996; Lerner, 2004), *collaborative completions* (Clark, 1996) *co-constructions* (Helasvuo, 2004), *co-participant completions* (Hayashi, 1999; Lerner and Takagi, 1999) *collaborative productions* (Szczepek, 2000) and *anticipatory completions* (Fox and others, 2007) amongst others.

[2] Note that this says nothing about whether such a continuation is the same as the initial speakers intended continuation.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 79–86,
Queen Mary University of London, September 2009.

In this regard, studies of split utterances, in both spontaneous dialogues and experimentally, as below, provide a complementary way of studying structural alignment to the traditional experimental set up exemplified by Branigan and colleagues (Branigan et al., 2000; Branigan et al., 2003; Branigan et al., 2006). Indeed, Poesio and Rieser (In preparation) claim that "[c]ollaborative completions . . . are among the strongest evidence yet for the argument that dialogue requires *coordination* even at the sub-sentential level" (italics original).

Broadly speaking, there have been two types, or levels, of explanations of split utterances offered; pragmatic accounts and processing accounts. Pragmatic accounts are favoured by Conversational Analysts, with various aspects of split utterances analysed. However, in line with CA assumptions, these analyses are almost exclusively concerned with the conditions under which split utterances can occur. Lerner (1991), for example, identifies a number of 'compound' turn-constructional units, such as the IF-THEN construction (whereby the second participant is in some sense licensed to provide the THEN part of the structure). However, Lerner's insistence on identifying the circumstances in which split utterances usually occur misses the important generalisation that, syntactically, they can be anywhere in a string (his *opportunistic completions*). His claim that an *anticipatory completion* is ordinarily "designed as a syntactic continuation of the utterance part it follows at the point of onset", seems to hold for all split utterances.

The occurrence of split utterances also has implications for the organisation of turn-taking, as outlined in Sacks et al. (1974). According to Schegloff (1995), turn-taking operates, not on individual conversational participants, but on 'parties'. For example, if a couple are talking to a third person, they may organise their turns as if they are one 'party', rather than two separate individuals. Lerner (1991) suggests that split utterances can clarify the formation of such parties; "collaboratively produced sentences reveal a relationship between syntax and social organisation. It provides evidence of how syntax can be mobilised to organise participants into "groups"."

The processing approach towards split utterances is exemplified by the interactive alignment model of Pickering and Garrod (2004). They claim that;

> . . . it should be more-or-less as easy to complete someone else's sentence as one's own, and this does appear to be the case.
> (Pickering and Garrod, 2004, p186)

According to this model, speaker and listener ought to be interchangeable at any point, and this is also the stance taken by the grammatical framework of Dynamic Syntax (Cann et al., 2005). In Dynamic Syntax (DS), parsing and production are taken to use exactly the same mechanisms, leading to a prediction that split utterances ought to be strikingly natural (Purver et al., 2006). Additionally, for a third person to process an utterance that appears to come from two separate speakers ought not be more difficult than processing the same utterance from a single speaker, regardless of where in a string the changeover occurs.

According to Poesio and Rieser (In preparation), "the study of sentence completions can shed light on a number of central issues. . . this type of data may be used to compare competing claims about coordination – i.e. whether it is best explained with an intentional model like Clark's. . . or with a model based on simpler alignment models like Pickering and Garrod's." As they see intentions as crucial to dialogue management, they conclude that a model which accounts for intentions (such as their PTT account) better captures their task specific split utterance data (See Poncin and Rieser (2006) for details of the German data they are modelling).

If this is the case, it ought to be more difficult to process an utterance that appears to be split between speakers, as opposed to one that comes from one source, because the intentions of the two different agents have to be considered in arriving at an interpretation, and they may appear to have formed a 'party' with respect to the subject of the utterance. Additionally it ought to be more disruptive to the conversation if the utterance is attributed to someone other than the person who genuinely contributed it, because the hearer would falsely attribute intentions to the wrong interlocutor. This ought to be especially clear in cases where the 'conversational momentum' appears to be with the 'wrong' interlocutor. Contrarily, if a processing model such as the interactive alignment model is correct, then no such differences should

be observed[3].

To test these predictions, an experiment was set up to alter genuine single-turn utterances into split utterances at an arbitrary point in the string. Different types of intervention were introduced, in a 2 x 2 factorial design, in order to separate out the effects of an utterance appearing to come from two different participants from effects caused by an apparent change of floor.

2 Method

The effects of seeing an utterance split between speakers or not were tested using the Dialogue Experimentation Toolkit (DiET) chat tool, as described in Healey et al. (2003), which enables dialogues to be experimentally manipulated.

The DiET chat tool allows interventions to be introduced into a dialogue in real time, thus causing a minimum of disruption to the natural 'flow' of the conversation. In this case, a number of genuine turns in a three way conversation were artificially split into two sections, with both parts either appearing to originate from the genuine source, or one or both parts being falsely attributed to another participant.

2.1 Materials

2.1.1 The Balloon Task

The *balloon task* is an ethical dilemma requiring agreement on which of three passengers should be thrown out of a hot air balloon that will crash, killing all the passengers, if one is not sacrificed. The choice is between a scientist, who believes he is on the brink of discovering a cure for cancer, a 7 months pregnant woman, and her husband, the pilot. This task was chosen on the basis that it should stimulate discussion, leading to dialogues of a sufficient length to enable an adequate number of interventions.

2.1.2 The DiET Chat Tool

The DiET chat tool itself is a custom built java application consisting of two main components, which will be outlined in turn; the user interface, and the server console.

[3]This is, of course, an oversimplification, and note that in contrast to pragmatic accounts, no claims are made regarding higher level discourse effects of the split utterance, as the focus is on the mechanisms which allow split utterances to occur. Additional mechanisms could of course be posited in processing models to account for any such differences.

2.1.3 User interface

The user interface is designed to look and feel like instant messaging applications e.g. Microsoft Messenger. It consists of a display split into two windows, with a status bar, indicating whether any other participant(s) are actively typing, between them (see figure 1). The ongoing dialogue, consisting of both the nickname of the contributor and their transmitted text, is shown in the upper window. In the lower window, participants type and revise their contributions, before sending them to their co-participants. All key presses are time-stamped and stored by the server.

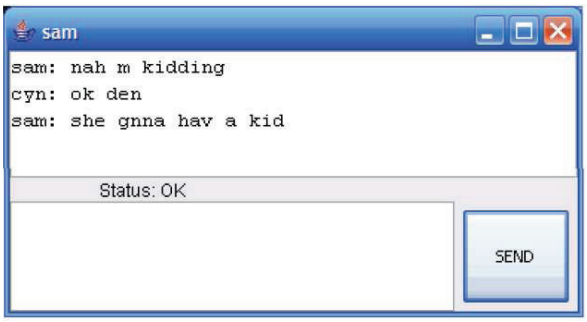


Figure 1: The user interface chat window (as viewed by participant 'sam')

2.1.4 Server Console

All text entered is passed to the server, from where it is relayed to the other participants, not relayed directly between participants. Prior to being relayed, some turns are altered by the server to create fake split utterances.

This is carried out automatically such that a genuine single-person turn is split around a space character near the centre of the string. The part of the turn before the space is relayed first, followed by a short delay during which no other turns may be sent. This is followed by the part of the turn after the space, as if they were in fact two quite separate, consecutive turns. In every case, the server produces two variants of the split utterance, relaying different information to both recipients. Each time an intervention is triggered, one of the two recipients receives both parts from the *actual* source of the utterance (henceforth referred to as an *AA*-split). The other recipient receives one of three, more substantial, manipulations; the first half could appear to be from the actual origin with the second part of the split appearing to originate from the other recipient (an *AB*-split), or

the inverse could be the case (a *BA*-split), or both parts could be wrongly attributed to the other participant (a *BB*-split). This design was in order to separate the effects of a change in conversational momentum (floor change) from the effects of splitting per se, hence the inclusion of the BB condition where who apparently has the floor is altered without the utterance being attributable to different participants. This contrast is shown in table 1.

Table 1: Comparison of split types

A types:		
Should we start now		
B sees (AA intervention):		
A: Should we A: start now		
C sees (one of):		
AB intervention:	BA intervention:	BB intervention:
A: Should we B: start now	B: Should we A: start now	B: Should we B: start now

The intervention is triggered every 10 turns, and restricted such that the participant who receives the non AA-split is rotated (to ensure that each participant only sees any of the more substantially manipulated interventions every 30 turns). Which of the three non AA-splits they see (AB, BA or BB) is, however, generated randomly.

2.2 Subjects

41 male and 19 female native English speaking undergraduate students were recruited for the experiment, in groups of three to ensure that they were familiar with each other. All had previous experience of internet chat software such as Microsoft Messenger and each was paid £7.00 for their participation.

2.3 Procedure

Each of the triad of subjects was sat in front of a desktop computer in separate rooms, so that they were unable to see or hear each other. Subjects were asked to follow the on screen instructions, and input their e-mail address and their username (the nickname that would identify their contributions in the chat window). When they had entered these, a blank chat window appeared, and they were given a sheet of paper with the task description on. Participants were instructed to read this carefully, and begin discussing the task with their colleagues via the chat window once they had done so. They were told that the experiment was investigating the differences in communication when conducted using a text only interface as opposed to face-to-face. Additionally, subjects were informed that the experiment would last approximately 20-30 minutes, and that all turns would be recorded anonymously for later analysis. Once all three participants had been logged on, the experimenter went to sit at the server machine, a fourth desktop PC out of sight of all three subjects, and made no further contact with them until at least 20 minutes of dialogue had been carried out.

3 Results

A post experimental questionnaire and debriefing showed that participants felt the conversations went as smoothly as face-to-face dialogue. With the exception of one subject, who had taken part in a previous chat tool experiment and was therefore aware that interventions may occur, none of the participants reported awareness of any interventions.

As production and receipt of turns sometimes occurs in overlap in text chat, it is not possible to say definitively when one turn is made in direct response to another[4]. We therefore chose two separate measures; *next turn* – the first turn, by the first recipient to start and complete a response, after receipt of the intervention, and *global* – all the turns produced by both recipients between the most recent intervention and the next intervention, averaged to produce one data point per recipient per intervention. This means that in the next turn condition, only one datapoint is analysed for each intervention, despite two different people seeing an intervention (and both usually producing a response). This was to try and isolate the initial response to an intervention; for the other person who saw a split but did not respond first, it is not clear if they are responding to the split utterance, or to

[4] In online chat, participants can compose their next turns simultaneously, and turns under construction when another is received can be subsequently revised, prior to transmission. This means that a genuine response to a split utterance might have a negative start time. However, the inclusion of cases where the whole turn was constructed after receiving the split (an arbitrary cut-off point, which would catch some turns that were responses to earlier turns in the dialogue, and miss some which were begun before the intervention was received and subsequently revised) should impose the same level of noise in all cases.

the person who *already* responded to the split utterance. In the global condition, in contrast, there are two datapoints for each intervention (one for each of the participants who saw a split utterance).

Of the 253 interventions to which at least one recipient responded, 89 were AA/AB splits, 99 were AA/BA splits and 65 AA/BB splits. Table 2 shows the n values in each case.

Both next turn and global measures were analysed according to two factors in a 2 x 2 factorial design; *split* – whether both parts of the utterance had appeared to come from the same person, or from different sources ([AA and BB] vs [AB and BA]), and *floor change* – who appeared to have produced the second part of the split, the genuine source, or the other participant ([AA and BA] vs [AB and BB]).

Measures selected for analysis were *typing time of turn* (The time, in milliseconds, between the first key press in a turn and sending the turn to the other participants by hitting the return key) and *length of turn in characters* as measures of production; *deletes per character* (The number of keyed deletes plus one (to prevent null values) divided by the total number of characters) as a measure of revisions; and *typing time per character* as a measure of speed. Data in tables are displayed in the original scale of measurement. However, as inspection of the data showed that they were not normally distributed, logarithmic transformations (using log_e) were applied to the data prior to all formal analyses.

2 x 2 ANOVAs show a main effect of floor change on the typing time of turn (see table 2). This holds for next turns ($F_{(3,249)} = 7.13$, $p < 0.05$) and globally ($F_{(3,486)} = 3.78$, $p < 0.05$), with participants taking longer over their turns in the AB and BB conditions. There was no main effect of split, and no effect of interaction. This effect is greater locally than globally, with participants who respond first after seeing a floor change condition taking more than 40% longer over their turns than those who saw a non-floor change condition. Globally the difference is in the order of 10%.

There was a main effect of split on the number of *deletes per character* , which also held both in the next turn condition ($F_{(3,249)} = 6.26, p < 0.05$) and globally ($F_{(3,486)} = 9.23$, $p < 0.05$), with subjects seeing a split condition (AB or BA) using *fewer* deletes per character than those seeing a non-split condition (see table 3). There was no main effect of floor change or interaction effect. This effect is also stronger in the next turn condition, with those not seeing a cross-person split using over 50% more deletes. In the global condition, this difference is still 40%, though the overall proportion of deletes is approximately 25% lower, from 0.334 per character in the next turn condition to 0.244 globally.

Table 2: Typing time of turn by type of intervention

Condition		Mean (s.d.)		N (poss N)	
Next Turn	AA	9475.54	(12258.5)	136	(253)
	AB	14560.70	(18863.9)	37	(89)
	BA	6968.24	(6437.0)	51	(99)
	BB	14812.59	(20367.8)	29	(65)
Global	AA	11122.27	(14413.5)	246	(253)
	AB	12500.98	(10944.6)	89	(89)
	BA	9800.77	(8810.3)	92	(99)
	BB	11561.67	(10138.4)	63	(65)

Table 3: Deletes per character by type of intervention

Condition		Mean (s.d.)	
Next Turn	AA	0.435	(1.63)
	AB	0.152	(0.30)
	BA	0.202	(0.25)
	BB	0.324	(0.61)
Global	AA	0.288	(0.83)
	AB	0.192	(0.28)
	BA	0.145	(0.18)
	BB	0.287	(0.37)

Additional analyses showed an effect of floor change on *length of turn in characters* (table 4) in the next turn condition ($F_{(3,249)} = 5.57$, $p < 0.05$) such that turns are longer in the AB and BB conditions (note that though this might be thought to be confounded by the typing time of turn, as you would expect longer turns to take longer to type, there are no significant effects when ANOVAs are performed on *typing time per character*). There is no main effect of split, or interaction effect. In the global condition, however, there is a main effect of split ($F_{(3,486)} = 4.08, p < 0.05$) such that turns are longer after seeing an utterance that appears to be split between two different people (AB and BA conditions). There is no main effect of floor change, and no effect of interaction.

As the experiment was looking for generic effects of splitting on coordination, the location of the splits was random. A post-hoc analysis was therefore carried out to ascertain whether the standalone coherence (as judged by the authors) of the two separate parts of the utterance was a possible confounding factor. Examples of coherence judgements are shown in table 5.

Table 4: Length of turn in characters by type of intervention

Condition		Mean (s.d.)	
Next Turn	AA	23.95	(22.0)
	AB	37.76	(34.9)
	BA	23.92	(18.4)
	BB	26.52	(21.5)
Global	AA	26.41	(20.4)
	AB	32.12	(23.9)
	BA	28.27	(18.4)
	BB	25.78	(13.6)

Table 5: Examples of standalone coherence judgement examples

Part of Split		Coherent	
First	Second	1st	2nd
what the hell	is that	Y	N
the woman is pregnant	she should stay	Y	Y
these people said	you did something	N	Y
I think this is also	the wish of the doctor	N	N

2 x 2 ANOVAs showed that in the next turn condition, there are no main effects of first or second part coherence, but there was an interaction effect of first part coherence by second part coherence on deletes ($F_{(3,249)} = 4.05$, $p < 0.05$), such that if *both* parts are independently coherent, or if *neither* part is independently coherent, there are fewer deletes used in the turn immediately following the intervention (see table 6). There are no significant global effects.

Table 6: Deletes per character by first and second part standalone coherence (next turn condition)

Coherence		Mean (s.d.)	
1st	2nd		
Y	Y	0.198	(0.38)
	N	0.651	(2.26)
N	Y	0.304	(0.66)
	N	0.206	(0.30)

Running a 2 x 2 x 2 x 2 ANOVA with these additional factors does not alter the main effects observed for floor change or split, as detailed above. There are no additional interaction effects on any of the measures.

4 Discussion

As this is the first experimental study into split utterances using the DiET chat tool, what follows is necessarily exploratory. This discussion presents our current hypotheses as to how best to interpret the data, as summarised in table 7, below.

Table 7: Summary of significant effects

Effect of	Condition	on and direction
Floor Change	Next Turn and Global	Typing Time $(AB \wedge BB) > (AA \wedge BA)$
Floor Change	Next Turn	Number of Chars $(AB \wedge BB) > (AA \wedge BA)$
Split	Next Turn and Global	Deletes $(AA \wedge BB) > (AB \wedge BA)$
Split	Global	Number of Chars $(AB \wedge BA) > (AA \wedge BB)$

Taking longer over the production of a turn (independently of typing speed) indicates a lack of confidence in the conversation (misattributing the second part of the utterance thus reducing confidence), and is also indicative of local organisation of turn-taking. If a participant who has seen a floor change intervention (Participant C) responds first, then they may be taking longer over their turns because there is less pressure on them to take a turn. This is because of the C's expectations. They will falsely believe that the fake source (Participant B) has just completed a turn, and will therefore not expect them to take the floor, and the genuine source (Participant A) will not be taking the floor because they have just completed a turn (though C does not know this). It is probable that in the turn immediately following a floor change intervention both these factors are at play, whereas globally it is the weaker effect of generic confidence loss that is observed. This compounding of effects in the next turn condition would also help explain the divergent effects on the length of turn in characters in next turn and global conditions.

Regardless of the precise reasons for it, this effect of floor change on typing time clearly demonstrates that changing the apparent speaker is disruptive, perhaps because it alters the forward mo-

mentum of the conversation.

More interestingly, independently of a change of floor, seeing an utterance that appears to be split between speakers also has an impact on the conversation, seen in the amount of revision undertaken in formulating a response (deletes). One reason why participants might worry less about precisely formulating their turns following a cross-person split is that the production of a cross-person split could have the effect on the recipient of suggesting that the two other participants have formed a 'party' (Schegloff, 1995) with respect to the decision of who to throw out of the balloon. This might be understood as signalling the formation of a strong coalition between the other two participants, therefore making the recipient behave as though they are resigned to the decision of this coalition. This is not the same as the effect on the typing time of turn, whereby participants are less rushed when seeing a change of floor. Deletes, on the other hand, demonstrate how carefully participants are constructing their turns. Excerpt 1, taken from the transcripts shows an example where this appears to be the case.

Excerpt 1 AB-Split showing apparent coalition between 'Bhups' and 'Dan' ('fake' part of split shown in bold)

Bhups:	and he can tell his formula
Dan:	**to tom and susie**

If we take split utterances as an indicator of coordination then it is likely that if we believe our two conversational partners to be in coordination, we will worry less about precisely formulating our own contributions. This also backs up the idea that people are not interchangeable.

The interaction of first and second part coherence also underlines the effect of split on revisions as outlined above. In the case were both parts of the split could potentially stand as independent utterances, they are treated as such and the number of deletes per character is in line with the global average (i.e. they are treated as normal dialogue). In the other non ambiguous case, where neither part could be interpreted as an utterance on its own, there are also fewer deletes, in line with the result that there are fewer deletes in strong split cases. Interestingly, the most disruptive case is that where the first part could have been a standalone utterance, but the second part could not. This could be seen as analogous to a garden path effect, and provides some indication that that the building up of interpretations is incremental, and not concerned with who supplies the input.

These results do not, of course, prejudice the claim that, at a purely mechanistic level, people could anticipate the structures needed to complete a turn, as the interactive alignment model suggests, because they are not concerned with the actual production of a split utterance, rather on the effect it has on the conversation. They do indicate that in terms of the effects of seeing split utterances, the pragmatic approach offers a more feasible level of analysis. For example, if we wish to treat a jointly produced split utterance as signalling especially strong alignment, then we need to account for more than simply syntax.

There is an issue with the design of the experiment which means that the floor change effects might be caused by a confounding variable; in essence, because one of the recipients always received an AA-split, in the cases which have been labelled as cases of floor change, the two recipients will have been left with the impression that a different person made the final contribution. This means that there may well be a an effect of confounded listener expectation (though see Schober and Brennan (2003) for discussion), although it should be noted that this does not have any bearing on the observed differences after an utterance split between speakers. It is also possible that split utterances might be particularly marked in a chat environment, though preliminary results of a corpus study show that, perhaps surprisingly, split utterances also occur naturally and as frequently in text-based chat (Eshghi, in prep) as they do in face-to-face dialogue (Purver et al., 2009). Because of these issues, and the already noted potential problems of linearity in text-based chat, a follow-up study using a character-by-character chat tool interface is underway. This more directly enforces turn-taking, as it does not allow participants to formulate their turn before communicating it; each character is transmitted as and when it is entered.

5 Conclusions

The experiment reported here offers clues towards an understanding of split utterances as an example of dialogue phenomena, and provides evidence

that speaker switches affect processing where they interfere with expectations about who will speak next and that the pragmatic effect of a split is to suggest to other participants the formation of a coalition or sub-'party'. It also clearly demonstrates that this type of experiment provides a fruitful line of future research in the ongoing attempt to adequately characterise dialogue, though further developments are needed.

References

H. Branigan, M. Pickering, and A. Cleland. 2000. Syntactic co-ordination in dialogue. *Cognition*, 75(2):13–25.

H. Branigan, M. Pickering, J. Pearson, J. McLean, and C. Nass. 2003. Syntactic alignment between computers and people: The role of belief about mental states. In *Proceedings of the Twenty-fifth Annual Conference of the Cognitive Science Society*.

H. Branigan, M. Pickering, J. McLean, and A. Stewart. 2006. The role of local and global syntactic structure in language production: Evidence from syntactic priming. *Language and cognitive processes*, 21(7-8):974–1010.

R. Cann, R. Kempson, and L. Marten. 2005. *The Dynamics of Language*. Elsevier, Oxford.

H. Clark. 1996. *Using Language*. Cambridge University Press.

A. Eshghi. in prep. *Uncommon ground: the distribution of dialogue contexts*. Ph.D. thesis, Department of Computer Science, Queen Mary University of London.

A. Fox et al. 2007. Principles shaping grammatical practices: an exploration. *Discourse Studies*, 9(3):299.

M. Hayashi. 1999. Where Grammar and Interaction Meet: A Study of Co-Participant Completion in Japanese Conversation. *Human Studies*, 22(2):475–499.

P. G. T. Healey, M. Purver, J. King, J. Ginzburg, and G. J. Mills. 2003. Experimenting with clarification in dialogue. In *Proceedings of the 25th Annual Meeting of the Cognitive Science Society*.

M. Helasvuo. 2004. Shared syntax: the grammar of co-constructions. *Journal of Pragmatics*, 36(8):1315–1336.

G. Lerner and T. Takagi. 1999. On the place of linguistic resources in the organization of talk-in-interaction: A co-investigation of English and Japanese grammatical practices. *Journal of Pragmatics*, 31(1):49–75.

G. Lerner. 1991. On the syntax of sentences-in-progress. *Language in Society*, pages 441–458.

G. Lerner. 1996. On the semi-permeable character of grammatical units in conversation: Conditional entry into the turn space of another speaker. In E. Ochs, E. A. Schegloff, and S. A. Thompson, editors, *Interaction and grammar*, pages 238–276. Cambridge University Press.

G. Lerner. 2004. Collaborative turn sequences. In *Conversation analysis: Studies from the first generation*, pages 225–256. John Benjamins.

M. Oelschlaeger and J. Damico. 1998. Joint productions as a conversational strategy in aphasia. *Clinical linguistics & phonetics*, 12(6):459–480.

M. Pickering and S. Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226.

M. Poesio and H. Rieser. In preparation. Completions, coordination, and alignment in dialogue. to appear.

K. Poncin and H. Rieser. 2006. Multi-speaker utterances and co-ordination in task-oriented dialogue. *Journal of Pragmatics*, 38(5):718–744.

M. Purver, R. Cann, and R. Kempson. 2006. Grammars as parsers: Meeting the dialogue challenge. *Research on Language and Computation*, 4(2-3):289–326.

M. Purver, C. Howes, P. G. Healey, and E. Gregoromichelaki. 2009. Split utterances in dialogue: a corpus study. In *SigDial 2009 workshop proceedings*.

H. Sacks, E. Schegloff, and G. Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, pages 696–735.

E. Schegloff. 1995. Parties and talking together: Two ways in which numbers are significant for talk-in-interaction. *Situated order: Studies in the social organization of talk and embodied activities*, pages 31–42.

M. Schober and S. Brennan. 2003. Processes of interactive spoken discourse: The role of the partner. *Handbook of discourse processes*, pages 123–64.

B. Szczepek. 2000. Formal Aspects of Collaborative Productions in English Conversation. *Interaction and Linguistic Structures (InLiSt), `http://www.uni-potsdam.de/u/inlist/issues/17/index.htm`*.

Interactive Gesture in Dialogue: a PTT Model

Hannes Rieser
Bielefeld University
Hannes.Rieser@uni-Bielefeld.de

Massimo Poesio
Università di Trento/University of Essex
poesio@essex.ac.uk

Abstract

Gestures are usually looked at in isolation or from an intra-propositional perspective essentially tied to one speaker. The Bielefeld multi-modal Speech-And-Gesture-Alignment (SAGA) corpus has many interactive gestures relevant for the structure of dialogue (Rieser 2008, 2009). To describe them, a dialogue theory is needed which can serve as a speech-gesture interface. PTT (Poesio and Traum 1997, Poesio and Rieser submitted a) can do this job in principle, how this can be achieved is the main topic of this paper. As a precondition, the empirical research procedure from systematic corpus annotation via gesture typology to a partial ontology for gestures is described. It is then explained how PTT is extended to provide an incremental modelling of speech plus gesture in an assertion-acknowledgement adjacency pair where grounding between dialogue participants is obtained through gesture.

1 Introduction and Overview

We present work combining experimental methods, body-movement tracking techniques, corpus linguistics and theoretical modelling in order to investigate the role of iconic gesture in dialogue. We propose to map speech meaning and gesture meaning into a single compositional meaning which is then used in grounding and up-dating of information states in discourse, using PTT (Poesio & Traum 1997, Poesio & Rieser submitted 2009a) to account for the speech-gesture interface. We argue that several design features of PTT are essential for this purpose, such as accepting sub-propositional inputs, extracting information from linguistic surface, using dynamic semantics, basing the dialogue engine on a theory of groundedness and grounding, and allowing for the resolution of anaphora across turns.

The structure of the paper is as follows. Section 2 looks at the Bielefeld Speech-and-Gesture-Alignment corpus SAGA from which the data comes. Section 3 then deals with multi-modal acts using one example from SAGA (Dial 1 p.**??**). In section 4 a short introduction into PTT is provided. Sections 5 and 6 explain how a gesture typology and a partial ontology can be extracted from the annotated data. Both (see Appendix) serve as the basis for the integration of gesture meaning and verbal meaning. In section 7 PTT is developed as an interface for verbal and gestural meaning. First a PTT description of Dial 1 is provided using (Poesio and Rieser submitted b, Poesio to appear) dealing *inter alia* with anaphora resolution (7.1). Secondly, PTTs interface properties are detailed (7.2), the semantic defaults for combining speech and gesture meaning are set up (7.3), and a gestural dialogue act is described (7.4). Section 8 contains some preliminary insights into the grounding of multi-modal content.

2 The Multi-modal SAGA Corpus

The SAGA corpus contains 25 route-description dialogues taken from three camera perspectives using body tracking technologies.[1] The setting comes with a Router "riding on a car" through a virtual landscape passing five landmarks. The landmarks are connected by streets. Fig. 1a in Appendix B shows the Router, Fig. 1b the site, Fig.

[1] cf. Bergmann, K. *et al.* (2007, 2008)

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 87–96,
Queen Mary University of London, September 2009.

1c the town hall. After the ride the Router reports his trip in detail to a Follower. We collected audio and body movement data as well as eye-tracking data from the Router. The dialogues have all been annotated, use of functional predicates like INDEXING, MODELLING, SHAPING[2] etc. was rated.

3 An Example from the SAGA corpus

In the dialogue passage (Dial 1) the Router uses gestures to explain the looks of the town-hall. We'll focus on the numbered utterances in this paper; utterances omitted in the reconstruction are reported in italics, omitted phrases in brackets.

DIAL 1 [ROUTER:] [. . .] *und [du] folgst dann dem*
[. . .] *and [you] follow then the*
Straßenverlauf einfach nur bis du ah vor nem
street simply until you ah before a
größeren Gebäude stehst.
larger building stand.

(2.1) Das ist dann das Rathaus.
That is then the townhall.

(2.2) [Ahm] das ist ein U-förmiges Gebäude.
[Ahm] that is a U-shaped building.

(2.3) Du blickst [praktisch] da rein.
You look [practically] into it.

(2.4) [Das heißt] es hat vorne zwei
[That is] it has to the front two
Buchtungen und geht hinten
bulges and closes in the rear
zusammen dann.
then.
[FOLLOWER:] OK.

In (Dial 1) Router's gestures first come with two BEATS.[3] Shortly after, the BEATs extend into an ICONIC gesture overlapping *town hall* in (2.1)(stills in Apendix B), cf. the still Two-Handed-Prism-Segment-1. Then the Router's DRAWN U-*shaped* gesture (still One-Handed-U-Shape) intersects the word U-shaped. Next his SHAPING the sides of a prism (still Two-Handed-U-Shaped-Prism-Segment) aligns with *[look pactically] into it.* The gesture following is two-handed: one hand SHAPES the U's left branch and the other both the U's right branch and its rear bend linking up to the left branch (stills Two-Handed-Prism-Segment-2A and 2B). The STROKE overlaps with the words *and closes in the rear.* The Follower copies the two-handed *town hall* gesture of the Router in his acknowledgement (still Two-Handed-Prism-Segment-3). In other words: the Follower's gesture is aligned to the Router's. Being copies of each other, the semantics of the Router's and the Follower's gesture can enter the common ground (*cf.* 7.4 and 8). In the reconstruction we will use the translation with the English word order standardised.[4]

4 A Short Introduction to PTT

Explanation of dialogue rests on three things: making clear how the succession of speakers' contributions emerges, stating what the impact of contributions on speakers' minds is and specifying how information is extracted incrementally from the contributions. Turning to emerging structure, PTT assumes that participants perform (often fragmentary) contributions, discourse units (DUs), which are dynamic propositions (DRSs in the sense of (Muskens, 1996)). They contain locutionary acts, conversational events/dialogue acts plus their propositional contents/DRSs. DUs may be sub-propositional micro-conversational events. Dialogue acts are either core speech acts or grounding acts. Core speech acts can be related to the present like *assert*, towards the past like *accept* or towards the future like *commit*. Grounding acts are *acknowledge* or *repair* (Traum 2009). Putting the distinctions above to work, we obviously can already model adjacency pairs. For the problems at issue we do not need more, cf. (Dial 1).

Which attitudes are assumed in current PTT and which changes of participants' minds are accounted for? Agents can have individual and private or common and public intentions. All sorts of actions, verbal or domain ones, are as a rule intended, at the outset of changes we have individual intentions. Common intentions are for example needed in order to explain completions and repairs (Poesio and Rieser submitted a). Most of the cooperation facts investigated in Clark (1996) need common intentions, most prominently, the intention to carry out a communicative task felicitously. Frequently, the vehicle for these types of intentions are (partial) plans. Plans can also be individual or shared. In (Dial 1) for example, the Router has an individual plan how to best map out his ride and the intention to communicate it to the

[2] Annotation PREDICATES are written in capital letters. Cf. also fn. 5.

[3] BEATS largely rest on supra-segmentals and would demand a paper of their own.

[4] We will end up with a mixture of German gesture and English wording here. However, for didactic purposes (sketch the main ideas) this seems acceptable. Sometimes we will simulate German constructions in English.

Follower. The Follower in turn intends to let the Router control her beliefs. Both have the collective intention to enable the Follower to follow the Router's route. Information presupposed or generated is contained in the discourse situation which, in PTT, is just a normal situation with objects and events, i.e., a DRS.

Conversational participants have command over information states. An information state is updated whenever a new event is perceived, including events such as sub-sentential utterances, and non-verbal events such as gestures or nods. Hence the possibility is already implemented in PTT to model accumulation of information due to gesture. Information common to the dialogue participants can be considered as grounded by default. This assumption connects PTT with other dialogue theories, for example Clark's (cf. Clark and Marshall, 1981, Clark and Schaefer, 1989) and Traum's (Traum 2009). Acknowledged information is at the heart of the grounding process. What is grounded is mutually believed *ceteris paribus.* Therefore, grounded information is part of the pragmatic machinery driving a dialogue forward (Rieser 2009). Grounding acts are taken as meta-discoursive devices and not included in discourse units proper. Besides beliefs and intentions we have obligations as mental attitudes. In PTT every conversational action induces an obligation on the participant indicated to address that action.

Information states raise the question of how changes of information are brought about on the basic grammatical level, viz. the interpretation of incrementally produced locutionary acts. The grammar in which syntactic and semantic interpretation is implemented is LTAG (Abeilleé & Rambow (eds), 2000). LTAG is a tree-grammar encoding syntactic projections which do the duty of, say, HPSGs rules, principles and constraints. Nodes and projecting leaves are decorated with semantic information based on Compositional DRT as developed in (Muskens, 1996, 2001). A specific trait of PTT is working with semantic non-monotonicity at all compositional levels: PTT hypothesizes that semantic computation is the result of defeasible inferences over DRSs obtained concatenating updates of single contributions. These default inference rules have the effect of semantic composition rules. Due to the impact of interpreted LTAG one can say that PTT is well founded in a bottom up fashion. Especially the default mechanism of PTT is used to make it a workable interface for speech and gesture (*cf.* 7.2 - 7.4).

5 Setting up the Speech-gesture Interface: Typology and Partial Ontology

As mentioned, this paper is based on the systematic annotation of SAGA carried out over the years 2007-2009 (Rieser 2009). Like many gesture researchers we assume that the semantic and pragmatic centre of a gesture is its **stroke**. The stroke overlaps as a rule with part of a complex constituent, for example the head or the logical subject. The range of speech-gesture overlap usually marks the functional position where the gestures meaning has to be merged into the speech content. Technically, the annotation is an ELAN-grid. From the annotation, a set of gesture types has been factored out in the following way (Rieser 2009). *AGENCY*[5] is installed as a root feature dominating the role features *ROUTER* and *FOLLOWER*. Next come the Router's and the Follower's *LEFT* and *RIGHT HAND* and *BOTH* their *HANDS*. *HANDEDNESS* in turn is mapped onto single annotation features like HANDSHAPE, WRISTMOVEMENT, PATHOFWRISTMOVEMENT etc. Bundles of features make feature *CLUSTERs* which yield classes of objects like curved, straight etc. entities. These build up *SHAPES* of different dimensions:[6] *ABSTRACT OBJECTs* of *0 DIMENSION* and *LINEs,* one-dimensional entities of different curvature. Among the two dimensional entities are *LOCATIONs, RECTANGLEs, CIRCLEs*[7] etc. Then three dimensional sorts come up: *CUBOIDSs CYLINDERs, PRISMs* and so on. In the end we get *COMPOSITEs* of *SHAPES,* for example a *BENT LINE* in a *SPHERE,* and S*EQUENCES OF COMPOSITEs.*[8] The central issue of 'How does a gesture acquire meaning'? is answered in the following way: A gesture type is mapped onto a partial ontology description, a stipulation encoding the content attributed to a gesture by raters. As a rule, gesture content is underspec-

[5]Gesture types, organised in an inheritance hierarchy working with defaults (cf. Rieser 2009), are written in *CAPITAL ITALICS*.

[6]In the following *geometry terms* are used mnemonically.

[7]*SHAPEs* can in general be fully developed or come in *SEGMENTs*. We do not deal with *SEGMENTs* here.

[8]*SEQUENCES* encode evolution of *SHAPEs* in time.

ified and will be completed to some extent when interfacing with verbal meaning. As an example of a gesture type and its partial ontology, see e.g. *TwoHandedPrismSegment1* and *'Partial OntologyTwoHandedPrismSegment1'* in Appendix A.

6 Setting up the Speech-gesture Interface: Levels of Interaction

Our starting point is the hypothesis detailed in (Rieser 2008) that a genuine understanding of dialogues like (Dial 1) requires integration of multimodal meaning at different levels of discourse, from fine grained lexical definitions up to rhetorical relations. In the rest of the paper, we will specify how information from spoken utterances merges with information from gestures, using (Dial 1) as an example. Omitting the two BEATS on *that is [then],* we have the following gestures on the Router's side (see stills in Appendix B):

6.1 the *PRISM SEGMENT* covering *the town hall;* cf. still Two-Handed-Prism-Segment

6.2 the DRAWN U-shape overlapping the adjective *U-shaped;* still One-Handed-U-Shape

6.3 the *PRISM SEGMENT* affiliated to [practically] *look into it;* still Two-Handed-U-Shaped-Prism-Segment

6.4 the two-handed U-shaped *PRISM SEGMENT* going with *and closes in the rear;* stills Two-Handed-Prism-Segment-2A and 2B.

The Follower uses a variant of

6.5 the Router's *PRISM SEGMENT* in (6.4) followed by OK; still Two-Handed-Prism-Segment-3.

The key observation from Rieser (2009) is that gestures interact with verbal contributions at different levels. (6.1) to (6.4) must be integrated at the level of the semantic interpretation of LTAG. (6.3) is involved since the stroke covers three constituents in the German wording, the modal adverb *[practically],* the pronoun it, and the separable prefix *da rein/into* of the verb *blickst/you look.* We will develop a simplified solution here using the "verb" *look-into.* Similarly, in (6.4) the gesture contains information relevant for *closes in the rear,* i.e. for the whole VP. The gesture information has to be integrated into the Router's dialogue acts at the interface points mentioned. Therefrom several side issues arise, for example the treatment of anaphora across Router's or Follower's contributions. In (Dial 1) the Follower uses gestural information only to acknowledge. It is a multimodal example of acknowledging by imitating the Router's multi-modal acts. Her gesture and the OK form a kind of "complex acknowledgement". This way the Router's contributions (6.2) to (6.4) and the Follower's contribution (6.5) show the interactive role of gesture, more specifically, gesture content in its use for grounding. We will briefly comment upon that in section 8.

7 Using PTT as an Interface for Verbal Meaning and Gestural Meaning

7.1 The verbal part of (Dial 1)

According to PTT, the discourse situation after the verbal updates brought about by (Dial 1) would be as follows.[9] (We only represent one aspect of the content of the initial utterances of (Dial 1).):

[DU0, DU1, DU2, DU3, DU4, DU5 |
DU0 **is** [. . . K1, . . . |
K1 **is** [b1 | **building**(b1), **large**(b1)],
. . .]
DU1 **is** [u2.1, K2, ce2.1 |
u2.1: **utter**(Router,"Das ist das Rathaus"),
sem(u2.1) is K2,
K2 is [th1, tnhl |
th1 **is** ιy1. K1; [| y1 is b1],
tnhl **is** ιu. [| **town hall** (u)],
th1 **is** tnhl,
ce2.1: **assert**(Router, Follower, K2),
generate(u2.1, ce2.1)],
DU2 **is** [u2.2, K4, ce2.2 |
u2.2: **utter**(Router, "das ist ein
U-förmiges Gebäude."),
sem(u2.2) **is** K4,
K4 **is** [th2 | th2 **is** ιy2. K5; [|s: y2 **is** b1],
building(th2), U-shaped(th2),
K5 is K1],
ce2.2: **assert**(Router, Follower, K4),
generate(u2.2, ce2.2)],
DU3 **is** [u2.3, K7, ce2.3|
u2.3: **utter**(Router, "Du blickst da rein"),
sem(u2.3) **is** K7,
K7 **is** [th3, s1 | th3 **is** ιy3. K8; [|s: y3 **is** b1],
s1: look-into(Follower, th3),[10]
K8 **is** K4],
ce2.3: **assert**(Router, Follower, K7),
generate(u2.3, ce2.3)],
DU4 **is** [u2.4, K9, ce2.4|
u4: **utter**(Router, "es hat vorne
zwei Buchtungen und geht hinten zus. dann"),
sem(u2.4) **is** K9,
K9 **is** [th4,bu1,bu2,s2,s3,s4,s5,s6,
re1,re2 |
th4 **is** ιy4. K10;
[| y4 **is** th3], K10 **is**
K7,
bulge(bu1), bulge(bu2),
s2: has(th4, bu1),

[9] Abbreviations used in the PTT-fragment: The prefixes are usually followed by a number n $\geq$ 0. DU = discourse unit, ce = conversational event, K = DRS, u = utterance, sem = semantic function, x, y, z . . . = DRs, e: event, s: = situation. In the DRSs ',' stands for conjunction an ';' between DRSs for composition of DRSs.

s3: has(th4, bu2),
to-front-of(bu1, th4),
to-front-of(bu2, th4),
rear(re1), s4: has(bu1,
re1), rear(re2)[11],
s5:has(bu2, re2),
s6: meet(re1, re2)],
ce2.44: **assert**(Router, Follower, K9),
generate(u2.4, ce2.4)].

The model of anaphora resolution accounting for the anaphoric cases is developed in (Poesio and Rieser, submitted 2009 b). The anaphoric *Das/this* in DU1 depends on the discourse entity *a larger building* introduced at the beginning of the conversation in DRS K1: K1 is the resource situation for the anaphoric definite. The second *das/this* still depends on the same resource situation. The pronouns, however, behave differently: Pronoun *da/there* in DU3 takes up the antecedent a *U-shaped building,* whereas the *es/it* in DU4 in turn refers to the *it* in DU3. Observe that the verbal part of (Dial 1) alone would already specify the interpretation completely: nothing essential is missing. As it will become clear below, what gestures do in this example is to add details to the verbally determined models and restrict the model set.

7.2 Tying in Gestures with Utterances

What we have got so far is a PTT-representation of the verbal part of (Dial 1). We now move on to how the information coming from the Router's gestures gets integrated with the verbal information – in particular, how this integration can take place below the sentential level. Our account builds on two key ideas from PTT. First of all, gestures are part of the discourse situation – i.e., the occurrence of gestures is recorded in the information state's representation of the discourse situation. Second, every occurrence of a sentence constituent counts as a conversational event – a MICRO CONVERSATIONAL EVENT (MCE). With these assumptions in place, the interaction of speech meaning and gesture meaning – how the two types of meanings combine to specify the overall meaning of a contribution – can be specified using the same mechanisms that specify the meaning of MCEs: i.e., with (prioritized) defaults in the sense of (Reiter, 1980, Brewka 1989). One example of a default specifying semantic composition is the BINARY SEMANTIC COMPOSITION (BSC) developed in (Poesio to appear, Poesio and Rieser submitted a) to specify the default way in which MCEs meanings can be derived from the meanings of their constituents. (We use the notation $>$ to indicate defeasible inference, $\uparrow$ to indicate 'dominated by'.)

BSC: u1$\uparrow$u, u2$\uparrow$u, sem(u1) **is** $\alpha_{\langle\sigma\iota\rangle}$ sem(u2) **is** β_σ, **complete**(u,u1,u2)
$>$ sem(u) **is** $\alpha(\beta)$

BSC can however be overridden in a number of circumstances: most notably, when anaphora interpretation processes identify a referent for a definite description like u_{NPI}: **utter**("the building"), in which case **sem**(u_{NPI}) will be the referent as opposed to a set of properties; or in cases of metonymy such as those studied by Nunberg (2004), in which the meaning of a MCE may be derived even more indirectly. We hypothesize that the integration of utterance meaning and gesture meaning is specified by **interface defaults** that may override the general meaning in a similar way by enriching the normal meaning of MCEs. We provide several examples of interface defaults below. For reasons of space, we only specify the results of default inference, without providing full derivations of the multi-modal meanings. For the gestures only the semantics[12] is specified, abstracted from the description of the partial ontology (cf. Appendix A for details). Utterance meaning then operates on the partial ontology information. MM abbreviates "multi-modal"; "lex-entry" means the word-form at stake, "lex-definition" means an explict dictionary definition for the word, for example in the style of the OED, cast into PL1.

7.3 The Interface Defaults

The general heuristic strategy for setting up interface defaults designed to combine verbal meaning and gesture meaning is to probe into the PTT structure as deep as you need in order to fit in the gestural content properly. Gestures may be relevant at any level of discourse, as shown in (Rieser, 2008) and demonstrated below; this means that sometimes gestural content has to be stored "deep

[10] Observe that the town-hall and the U-shaped building are the same.

[11] Observe that the gesture dynamically shapes two rears which meet.

[12] This is due to the fact that we do not integrate gestures into the discourse situation here. If these are integrated one will use their type description as syntax in AVM format. Gestures do not have the normal category syntax.

in" the lexical definition of a word, at other times one has to remain on the top level of semantic composition or even follow up the contributions produced so far. The interface defaults mostly follow the general schema:

λ-prefix mentioning the open parameters + lexicon definition + open parameters applied to iconic meaning = λ-abstracted partial ontology description where the λ-bound parameters secure binding.

An exception to this is (7.3.5.1) which uses the notion of satisfaction (see stills in Appendix B).

7.3.1 The PRISM SEGMENT aligned with *[the] town hall* (6.1). To begin with, gestural meaning can enrich the meaning of a nominal utterance. The interface default allowing this is called **Noun meaning extended (NMExt)**[13]

> **NMExt**: Noun(u), sem(u) **is** λx **lex-definition**(x), u$\uparrow$u', N'(u'), u overlaps g, gesture(g), iconic-meaning(g) **is** λp partial ontology(p)
> $>$sem(u') is λx (lex-definition(x)) iconic-meaning(g)(x)

For instance in the dialogue under consideration **lex-definition** is the predicate 'large building used for the administration of local government' abbreviated as 'λPλx [[|s: large building(x), used for the administration of local government(x)]; P(x)]' and the Partial Ontology *TwoHandedPrismSegment1* from the Appendix A, resulting in the following meaning for the utterance of 'town hall' accompanied by the gesture:

> (7.3.1.1) λx [ls, rs, loc|s: large building(x), used for the administration of local government(x), side(ls, x), left(ls, Router), side(rs, x), right(rs, Router), location(loc, x)]

Observe that the fine-grained local information is provided by the gesture.

7.3.2 The DRAWN U-shape overlapping the adjective *U-shaped* is an example of gesture enriching an adjectival meaning through the interface default **Adjective meaning extended (AdjMExt)**

> **AdjMExt**: Adjective(u), sem(u) **is** λPλx [|lex-entry(x), P(x)], u$\uparrow$u', N'(u'), u overlaps g, gesture(g), iconic-meaning(g) **is** λp partial ontology(p)
> $>$ **sem**(u') **is** λPλQλx([|lex-entry(x), P(x)]; Q(x)) iconic- meaning(g)(x).

[13] 'λp partial ontology (p)' in NMExt and the following defaults is used in the following way: The expression 'partial ontology' refers to information from the partial ontology list in the Appendix A. What has to be chosen can be seen from the application of the default below.

Using **AdjMExt** and the meaning of the gesture *OneHanded-U-shape* in the Partial Ontology we obtain (7.3.2.1) as an enriched meaning for "U-shaped", '$\oplus$' denoting mereological composition:

> (7.3.2.1) λQλx([|U-shaped(x), λus(strai- ght-line(lr, us), arc(lb, us), straight-line(ll, us), us = lr $\oplus$ lb $\oplus$ ll)(x)]; Q(x))

After fitting in the noun modified by the multimodal content into position 'Q', the DRs will have to be correctly bound.

Observe that we could apply (NMext) and (AdjMext) iteratively to arrive at a complex MM Nom-meaning.

7.3.3 The PRISM SEGMENT affiliated to *[practically] look into it* is computed using the interface default **Verb meaning extended (VMExt)**.

> **VMExt**: VP(u), V(u1), NP(u2), u1$\uparrow$ u, u2$\uparrow$ u, sem(u1) **is** λPλx([|s: lex-definition(x), P(x)], u overlaps g, gesture(g), iconic-meaning(g) **is** λp partial ontology(p)
> $>$ **sem**(u) **is** λPλx([|s: lex-definition(x), P(x)]) iconic-meaning(g)(x)

VMExt gives us, again using the information from the Partial Ontology *TwoHanded-U-shapedPrism* from the Appendix:

> (7.3.3.1) λx([|s: focus(agent, x), space(x), bounded(x), empty(x), λp[hl, ls, lel, fs, hr, rs, ler, d| prism(p), height(hl, ls), left-side(ls, p), front-side(fs, p), left(ls, Router), height(hr, rs), right-side(rs, p), length(ler, rs), right(rs, Router), length(lel, ls), distance(d, ls, lr), lel = ler](x)])

Again we see that fine-grained information is provided by the gesture, especially the pragmatic anchoring of the space looked into from the Router's position.

7.3.4 Finally, the two-handed U-shaped PRISM SEGMENT going with *and closes in the rear* needs a default **VP meaning extended (VPMextended)**. The gesture information is distributed among the verb "closes" and the PP "in the rear", the assumption being that the object closing does so at a particular location which is part of the object itself. So we have:

> **VPMExt**: V(u) $\uparrow$ VP(uph1), P(u) $\uparrow$ PP(uph2), Det(u) $\uparrow$ NP(uph3), Nom(uph4) $\uparrow$ NP(uph3), PP(uph2) $\uparrow$ VP(uph1), **sem**(u) **is** λPλx([|lex-definition(x)]; P(x)), u overlaps g, gesture(g), iconic-meaning(g) **is** λp partial ontology(p)
> $>$ **sem**(uph1) = λPλx([|lex-definition(x), P(x)]; iconic-meaning(g)(x)

The default using Appendix A, Partial Ontology *TwoHanded-U-shapedPrism,* generates the following MM meaning:

(7.3.4.1) λx([|s: close(x), at(s, loc), prism(leftp), prism(rightp), part(leftp, x), part(rightp, x), section(sectl, leftp), leftside(lefts, leftp), length(ll, lefts), left(lefts, Router), section(sectr, rightp), rightside(rights, rightp), frontside(fronts, rightp), bent(rightp), meet(lefts, rights, loc), right(rights, Router), parallel(lefts, rights), distance(d, lfts, rhts)]).

7.3.5 The Follower's U-shaped gesture: So far, gesture meaning constrained word meanings or constituent meanings. In contrast, the Follower's U-shaped gesture invades dialogue structure. The Follower's reply has two steps. Her iconic gesture yields a predicate U-shaped in much the same way as the Router's contribution in DU2 and DU4 does. This is combined with a DR anaphorically linked to the Router's preceding *its* and *thats*. The gesture in turn takes up the Router's U-shapes from DU2 and DU4. So we get an anaphora related to antecedent multi-modal information.[14] Her "OK" then simply acknowledges her own DU5 filled up. Acknowledgement of the Router's contributions is achieved indirectly. In order to model all that, we have to Hook up the Gesture's Content with a DR. This is simply

(7.3.5.1) λp(iconic-meaning(p))DR for some preceding discourse referent DR satisfying iconic-meaning.

The relevant iconic meaning is taken from Partial Ontology *TwoHandedPrismSegment3:* section(sect, p), leftpart(lftp, p), lengthl(lftp), left(leftp, Follower), rightpart(rtp, p), right(rightp, Follower), lengthr(rtp), lftp = rtp, p = lftp ⊕ rtp.

7.4 A Gestural Dialogue Act of Assertion

Concerning dialogue structure, we have concentrated on the verbal part of (Dial 1) in 7.1. In the SAGA corpus there are many data showing how dialogue structure interfaces with gesture meaning. In 7.3.5 a default for the follower's U-shaped gesture was given. Its embedding into the PTT-description of (Dial 1) is shown in DU5 below:

(7.4) DU5 **is** [g1, K10|
g1: gesticulate(Follower, Router, U-shape),
sem(g1) **is** K10,
K10 **is** [|s: th5 is th4, λp(section(sect, p), leftpart(lftp, p), lengthl(lftp), left(leftp, Follower), rightpart(rtp, p), right(rightp, Follower), lengthr(rtp), lftp is rtp, p is lftp ⊕ rtp)(th5))]
ce5: **assert**(Follower, Router, K10),
generate(g1, ce5)],],

[14]These anaphorical relations are not reconstructed here but delegated to a follow-up paper.

[ce6, u6|
u6: **utter**(Follower,"OK"),
ce6: **ack**(Follower, DU5),
textbfgenerate(u6, ce6)]]

In the multi-modal dialogue passage we have 'gesticulate' instead of 'utter'. The semantics, using the default (7.3.5.1) 'Hook up the Gesture's Content with a DR' and material from Appendix A is provided in the standard way by K10. It is assumed that gestural content can be generated and asserted. The Follower's acknowledgement is a sort of self-acknowledgement that percolates up through anaphora.

8 Grounding by Gesture: a Genuine Case of Gestural Alignment

The different defaults, Noun-meaning extended (**NMextended**), Adjective meaning extended (**AdjMextended**), Verb meaning extended (**VMextended**), VP meaning extended (**VPMextended**) and **Hook up the Gesture's Content with a DR**, clearly indicate that integration of gesture meaning has to operate on levels of different grain. Gesture can operate on a sub-word level if one has to attach its meaning to parts of a lexical definition, on the word level, on the level of constituents, and, as a consequence of all that, on specific dialogue acts. Furthermore, we have seen gesture at two inter-propositional levels at work, at the interface among the contributions of one agent (see Router's contributions which are all "united" by communicating the appearance of the town hall) and at the interface among contributions of different agents (Router-Follower).The Follower acknowledges by imitating gestures of the Router; this is a genuine case of gestural alignment. Alternatively, she could also acknowledge verbally, uttering 'U-shaped' but she chooses a gestural content. Obviously, speakers think that this works. Her 'OK' furthermore shows that verbal and gestural means can work in tandem. So, in the end, the U-shape of the town hall is rooted in the common ground by default and the Router can continue with describing the route leading to the next landmark.

Acknowledgements

Support by the SFB 674, Bielefeld University, is gratefully acknowledged. We also want to than three anonymous reviewers for carful reading

and suggestions for improvement. Hannes Rieser wants to thank Florian Hahn for common work on gesture typology starting in 2007.

References

Abeillé, A & Rambow, O. (eds) 2004. *Tree Adjoining Grammars. CSLI Publ.* Stanford, CA.

Bergmann, K., Fröhlich, C., Hahn, F., Kopp, St., Lücking, A. and Rieser, H. June 2007. *Wegbeschreibungsexperiment: Grobannotationsschema.* Univ. Bielefeld, MS.

Bergmann, K., Damm, O., Fröhlich, C., Hahn, F., Kopp, St., Lücking, A., Rieser, H. and Thomas, N. June 2008. *Annotationsmanual zur Gestenmorphologie.* Univ. Bielefeld, MS.

Brewka, G. 1989. *Nonmonotonic reasoning: from theoretical foundation towards efficient computation.* Hamburg, Univ., Diss.

Clark, H. H. 1996. *Using Language.* Cambridge University Press.

Clark, H. H. and Marshall, C. R. 1981. Definite Reference and Mutual Knowledge. In A. K. Joshi, B. Webber, and I. A. Sag (eds.),*Elements of Discourse Understanding.* CUP

Clark, H. H. and Schaefer, E. F. 1989. Contributing to Discourse. *Cognitive Science,* 13, 259-294.

Muskens, R. 1996. Combining Montague Semantics and Discourse Representation. *Linguistics and Philosophy 19,* pp. 143-186.

Muskens, R. 2001. Talking about Trees and Truth-conditions. *Journal of Logic, Language and Information, 10(4),* pp. 417-455.

Nunberg, G. 2004. The Pragmatics of Deferred Interpretation. In: Horn, L.R. and Ward, G.: *The Handbook of Pragmatics.* Blackwell Publishing Ltd.

Poesio, M. to appear. *Incrementality and underspecification in semantic interpretation.* CSLI Publications.

Poesio, M. February 2009. *Grounding in PTT.* Talk given at Bielefeld Univ.

Poesio, M. and Rieser, H. submitted a. *Completions, Coordination, and Alignment in Dialogue.*

Poesio, M. and Rieser, H. submitted b. *Anaphora and Direct Reference: Empirical Evidence from Pointing.*

Poesio, M. and Traum, D. 1997. "Conversational Actions and Discourse Situations", *Computational Intelligence,* v. 13, n.3, 1997, pp.1- 45.

Rieser, H. 2008. Aligned Iconic Gesture in Different Strata of MM Route-description Dialogue. In *Proceedings of LONdial 2008,* pp. 167-174

Rieser, H. 2009. On Factoring out a Gesture Typology from the Bielefeld Speech-And- Gesture-Alignment Corpus. Talk given at the *GW 2009*, ZiF Bielefeld, to appear in the Proceedings of GW 2009.

Traum, D. 2009. Computational Models of Grounding for Human-Computer Dialogue. Talk given at Bielefeld Univ., February 2009

Appendices

Appendix A: Gesture Types and Description of Partial Ontology

Due to limited space gesture types and ontology descriptions are only partially characterised.

TwoHandedPrismSegment 1	
R.G.Left.HandShapeShape	*loose B spread*
R.G.Left.HandPalmDirection	*PDN/PTR*
R.G.Left.BackOfHandDirection	*BAB*
R.G.Left.Practice	*grasping-indexing*
R.G.Left.Perspective	*speaker*
R.G.Right.HandShapeShape	*loose B spread*
R.G.Right.HandPalmDirection	*PDN/PTL*
R.G.Right.BackOfHandDirection	*BAB*
R.G.Right.Practice	*grasping-indexing*
R.G.Right.Perspective	*speaker*
R.Two-handed-configuration	*TT*
R.Movement-relative-to-other-hand	0

Partial Ontology TwoHandedPrismSegment 1	
R.G.Left.HandShapeShape-loose B spread	*side(ls, p)*
R.G.Left.HandPalmDirection-PDN/PTR	*left(ls, Router)*
R.G.Right.HandShapeShape-loose B spread	*side(rs, p)*
R.G.Right.HandPalmDirection-PDN/PTL	*right(rs, Router)*
R.Two-handed-configuration-TT	*location(loc, p)*

OneHanded-U-shape	
R.G.Right.HandShapeShape	*G*
R.G.Right.PalmDirection	*PDN/PTL>PDN/PTB>PDN*
R.G.Right.BackOfHandDirection	*BAB/BTL>BAB/BDN>BAB/BDN/BTL*
R.G.Right.PathOfWristLocation	*ARC*
R.G.Right.WristLocationMovementDirectio	*MR>MF>ML*
R.G.Right.Extent	*MEDIUM*
R.G.Right.Practice	*drawing*
R.G.Right.Pespective	*speaker*

Partial Ontology OneHanded-U-shape	
R.G.Right.PathOfWristLocation-ARC	*U-shape(us)*
R.G.Right.WristLocation MovementDirection-MR>MF>ML	*straight-line(lr, us)* ∧ *bent-line(lb, us)* ∧ *straight-line(ll, us)*

TwoHandedPrismSegment 2	
R.G.Left.HandShapeShape	*B spread*
R.G.Left.HandPalmDirection	*PTR*
R.G.Left.BackOfHandDirection	*BAB/BUP > BAB*
R.G.Left.PathOfWristLocation	*LINE*
R.G.Left.WristLocation MovementDirection	*MF*
R.G.Left.Practice	*shaping-modelling*
R.G.Left.Perspective	*speaker*
R.G.Right.HandShapeShape	*B spread*
R.G.Right.HandPalmDirection	*PTL*
R.G.Right.BackOfHandDirection	*BAB/BUP > BAB*
R.G.Right.PathOfWristLocation	*LINE*
R.G.Right.WristLocation MovementDirection	*MF*
R.G.Right.Practice	*shaping-modelling*
R.G.Right.Perspective	*speaker*
R.Two-handed-configuration	*PF*
R.Movement-relative-to-other-hand	*SYNC*

Partial Ontology TwoHandedPrismSegment 2	
R.G.Left.HandShapeShape-B spread	*hight(hl, ls)*
R.G.Left.HandPalmDirection-PTR	*leftside(ls, p)* ∧ *prism(p)*
R.G.Left.PathOfWristLocation-LINE	*length(lel, ls)*
R.G.Left.WristLocation MovementDirection-MF	*frontside(fs, p)*
R.G.Left.Perspective-speaker	*left(ls, speaker)*
R.G.Right.HandShapeShape-B spread	*hight(hr, rs)*
R.G.Right.HandPalmDirection-PTL	*rightside(rs, p)* ∧ *prism(p)*
R.G.Right.PathOfWristLocation-LINE	*length(ler, rs)*
R.G.Right.WristLocation MovementDirection-MF	*frontside(fs, p)*
R.G.Right.Perspective-speaker	*right(rs, speaker)*
R.Two-handed-configuration-PF	*distance(d, ls, lr)*
R.Movement-relative-to-other-hand-SYNC	*lel = ler*

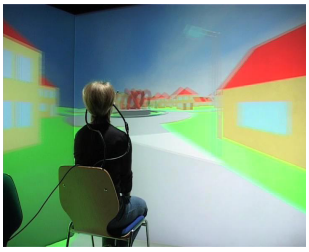
TwoHanded-U-shapedPrism	
R.G.Left.HandShapeShape	*small C*
R.G.Left.HandPalmDirection	*PAB*
R.G.Left.BackOfHandDirection	*BAB/BTR*
R.G.Left.PathOfWristLocation	*LINE*
R.G.Left.WristLocation MovementDirection	*MF*
R.G.Left.Practice	*shaping*
R.G.Left.Perspective	*speaker*
R.G.Right.HandShapeShape	*small C*
R.G.Right.HandPalmDirection	*PAB/PTL> PTL>PTB/PTL*
R.G.Right.BackOfHandDirection	*BAB/BTR> BAB>BAB/BTL*
R.G.Right.PathOfWristLocation	*LINE>LINE*
R.G.Right.WristLocation MovementDirection	*MF>ML*
R.G.Right.Practice	*shaping*
R.G.Right.Perspective speaker	
R.Two-handed-configuration	*BHA*
R.Movement-relative-to-other-hand	*SYNC*

Partial Ontology TwoHanded-U-shapedPrism	
R.G.Left.HandShapeShape-small C	*section(sectl, leftp)*
R.G.Left.PathOfWristLocation-LINE	*leftside(lefts, leftp)*
R.G.Left.WristLocation MovementDirection -MF	*length(ll, lefts)*
R.G.Left.Perspective-speaker	*left(lefts, speaker)*
R.G.Right.HandShapeShape-small	*section(sectr, rightp)*
R.G.Right.PathOfWristLocation-LINE>LINE	*rightside(rights, rightp)* ∧ *frontside(fronts, rightp)*
R.G.Right.WristLocation>ML MovementDirection-MF	*bent(rightp)* ∧ *meet(lefts, rights)*
R.G.Right.Perspective-speaker	*right(rights, speaker)*
R.Movement-relative-to-other-hand-SYNC	*parallel(lefts, rights)* ∧ *distance(d, lefts, rights)*

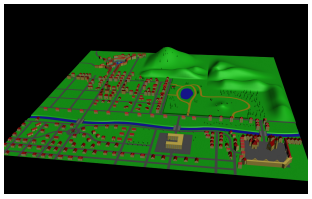
TwoHandedPrismSegment 3	
R.G.Left.HandShapeShape	*C*
R.G.Right.HandPalmDirection	*PDN/PTR>*
	PAB/PUP
R.G.Left.BackOfHandDirection	*BAB>*
	BTL/BUP
R.G.Left.PathOfWristLocation	*ARC*
R.G.Left.WristLocationMovementDirection	*ML>MB*
R.G.Left.Practice	*shaping*
R.G.Left.Perspective	*speaker*
R.G.Right.HandShapeShape	*C*
R.G.Right.HandPalmDirection	*PDN/PTL>*
	PAB/PUP
R.G.Right.BackOfHandDirection	*BAB>*
	BTR/BUP
R.G.Right.PathOfWristLocation	*ARC*
R.G.Right.WristLocationMovementDirection	*MR>MB*
R.G.Right.Practice	*shaping*
R.G.Right.Perspective	*speaker*
R.Two-handed-configuration	*BHA*
R.Movement-relative-to-other-hand	*Mirror-Sagittal*

Partial Ontology TwoHandedPrismSegment 3	
R.G.Left.HandShapeShape	*section(sect, p)*
R.G.Left.PathOfWristLocation	*leftpart(lftp, p)*
R.G.Left.WristLocationMovementDirection	*lengthl(lftp)*
R.G.Left.Perspective	*left(leftp, speaker)*
R.G.Right.HandShapeShape	*section(sect, p)*
R.G.Right.PathOfWristLocation	*rightpart(rtp, p)*
R.G.Right.WristLocationMovementDirection	*lengthr(rtp)*
R.G.Right.Perspective	*speaker*
R.Two-handed-configuration	$lftp = rtp$
R.Movement-relative-to-other-hand	$p = lftp \oplus rtp$

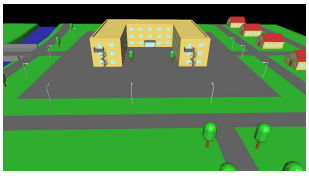
Appendix B: Figure 1



(a) The Router on his trip.



(b) The site traversed by the Router. The U-shaped building is the town hall



(c) Fig. 1c shows the town hall as described and gestured by the Router.

(d) Two-Handed-Prism-Segment-1

(e) One-Handed-U-Shape

(f) Two-Handed-U-Shaped-Prism-Segment

(g) Two-Handed-Prism-Segment-2A

(h) Two-Handed-Prism-Segment-2B

(i) Two-Handed-Prism-Segment-3

Figure 1: The SAGA Setting

Tense, temporal expressions and demonstrative licensing in natural discourse

Iker Zulaica-Hernández
Department of Spanish & Port.
The Ohio State University
Columbus, Ohio 43210
ikerzulaica@gmail.com

Javier Gutiérrez-Rexach
Department of Spanish & Port.
The Ohio State University
Columbus, OH 43210
gutierrez-rexach.1@osu.edu

Abstract

Demonstrative terms are highly context-dependent elements both in deictic and anaphoric uses. When reference is transferred from a visual, three-dimensional context to the textual domain, information-structure factors (i.e. the cognitive status of the antecedent, recency of mention, syntactic structure or the semantic type of the antecedent) have an effect on speaker preferences for selecting demonstrative anaphors over other referring expressions. In certain languages, there seems to be a correlation between demonstratives and tenses in discourse. For example, proximal demonstratives correlate better with present tenses whereas distal demonstratives correlate with past tenses. In this paper, we present a corpus study of Spanish texts that analyzes the ways in which temporal expressions selectively favor the use of specific demonstratives thus confirming the contextual dependency of demonstrative anaphors.

1 Introduction

As referring expressions, adnominal and pronominal demonstratives (this/that) can be used in two basic 'modes' that allow speakers to refer to entities in different discourse situations. In the deictic mode, speakers commonly use demonstratives to refer to physical, concrete entities in the real-world speech situation. Utterance of the demonstrative (very likely accompanied by a pointing gesture) has an important communicative effect, namely, that of focusing the attention of the addressee on a particular entity in the perceptual or visual field. This is accomplished by making the intended entity salient from among a set of (potentially) competing entities. Thus, both speaker and addressee focus their attention on the same element and the speaker's intended communicative goal is achieved. This is the so-called *joint attention* effect in the psycholinguistics literature (Diessel 2006) In these exophoric uses, the role of the pointing gesture (a pointing finger, a gaze or movement of the head) may become essential. It completes the meaning of the demonstrative expression and serves to disambiguate the speaker's reference. Demonstratives can also be used exophorically without an accompanying ostension, but in such cases the entity referred to is already sufficiently salient in the visual field for the interlocutors to have focused their attention on it and, consequently, an accompanying gesture by the speaker would be redundant or irrelevant to achieve the intended communicative goal.

There are certain uses of demonstrative elements that depart from the purely deictic mode. These uses, long and widely recognized, have been characterized as anaphoric or discourse anaphoric many authors (see *inter alia* Asher, 1993; Diessel, 1999; Janssen, 1996; Gundel et al. 2001, 2003; Hegarty et al. 2001). Thus, as demonstrative anaphors, demonstratives like English *this/that* are coreferential with a range of textual elements. From a syntactic point of view, the antecedents of demonstrative anaphors can be of a varied nature: NPs, subordinate clauses, entire sentences or larger textual fragments. Semantically, these antecedents comprise a rich ontology that ranges from individuals or propositions to event and event-types. Whether the anaphoric referring mode is derived from a primary deictic character or not is an issue quite beyond the purpose of this paper, but studies on language acquisition indicate that the deictic

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 97–106,
Queen Mary University of London, September 2009.

features are learned at the earlier stages (Diessel 2006).

The only difference between the referential capabilities of deictic and anaphoric uses of demonstratives lies in the fact that these capabilities have been transferred from a real-world context of utterance common to strict deictic uses to a textual (endophoric) domain in the anaphoric use. The communicative function remains the same. The pointing gesture, absent in demonstrative anaphors, seems to have evolved into derived pragmatic functions in the anaphoric use. A key feature that is common to both deictic and anaphoric uses is their high degree of contextual dependency. This issue will be explored in the next sections.

2 Referential distance

In deictic uses, pronominal and adnominal demonstratives are highly context dependent. In order to be properly used and fully interpreted, they require the aid of contextual parameters such as the speaker, the addressee, the location of the deictic center, the location of the object pointed at, the utterance time, etc. This is not only true of demonstratives like *this* (NP) or *that* (NP) but also of other indexical expressions like *here*, *there*, *I*, *you*, etc.

Demonstrative anaphors also appear to be dependent on contextual factors to a high degree. The most relevant parameter in anaphoric uses is textual context. Elements such as referential distance, lexical clues, or syntactic structure may have an effect on the speaker's preferences for one demonstrative over the other(s), or over other referential expressions. For example, Gundel et al (2001, 2003) examined referential expressions, including demonstratives and the personal pronoun *it*, in different environments and came to the conclusion that several factors, most prominently information structure, have an effect in the way clausally-introduced entities are referred to with these expressions. Different referential expressions (demonstratives, indefinites, the definite article, etc.) have the property of 'marking' the cognitive status of their antecedents. In the Givenness Hierarchy (Gundel et al. 1993), the antecedents of demonstratives are cognitively ACTIVATED whereas those of the unstressed personal pronoun *it* are IN FOCUS. As they point out, "the entities IN FOCUS at a given point in the discourse will be that partially-ordered subset of activated entities which are likely to be continued as topics of subsequent utterances." (2001: 40). A very important factor in determining the status of an entity is syntactic structure. Consider the following discourses:

(1) a. My neighbor 's Bull Mastiff bit a girl on a bike.
    b. **It's**/**That's** the same dog that bit Mary last summer.
(2) a. Sears delivered new siding to my neighbors with the Bull Mastiff.
    b. #**It's**/**That's** the same dog that bit Mary last summer.

In (1)[1], the NP *My neighbor's Bull Mastiff* occupies the subject position of a main clause and it is very likely the discourse topic. This brings the entity into the FOCUS of attention and, therefore, can be indistinctively referred to using personal and demonstrative pronouns. The anaphor '*that'* can be used to refer to the NP for entities IN FOCUS are also ACTIVATED, namely, in working memory. On the other hand, entities introduced in subordinate clauses or prepositional phrases are more likely to be rendered the cognitive status ACTIVATED upon their introduction in discourse. This point is shown in example (2), where the same NP is introduced within a prepositional phrase. This peripheral syntactic position renders the cognitive status of the antecedent ACTIVATED hence banning the use of the personal pronoun *it* and licensing the use of the demonstrative pronoun.

Another contextual factor that has been investigated as bearing clear implications on the use of demonstrative anaphors by language users is referential distance. By referential distance we mean the textual distance between the antecedent and the demonstrative anaphor. Textual or referential distance is commonly quantified as the number of intervening clauses between antecedent and anaphor. Hegarty et al. (2000) observed that English demonstrative pronouns and adjectives (**this**/**that**-(NP)) show a strong preference for their antecedents to be found in the clause immediately preceding the clause holding the demonstrative expression.

In quite the same line, Kirsner et al. (1987) investigate the factors that affect demonstrative (*deze* 'this' vs *die* 'that') choice in written Dutch. One of these determining factors is the magnitude of referential distance. Based on texts from different subcorpora tested on native Dutch speakers and comprising various different dis-

[1] This example appears in Gundel et al. (2003)

course genres, these authors showed that the Dutch proximal demonstrative *deze* ('this') tends to be associated with referential distance ≥ 1 (extrasentential retrieval of a referent) and distal die ('that') tends to be associated with referential distance = 0 (intrasentential retrieval of a referent). Their study showed that only 15% of NPs with *deze* ('this') have referential distance = 0, whereas a 40% of the NPs with *die* ('that') have referential distance = 0.

In our empirical corpus study, we have analyzed the referential distance factor for Spanish demonstrative pronouns with the aim of checking whether this contextual parameter may have an influence on the speakers' preferences for one demonstrative over the others. Let us first briefly characterize Spanish demonstratives. Unlike English, Spanish has a tripartite demonstrative system with three elements (*este*, *ese* and *aquel*) inflected for gender and number. As deictic elements, these demonstratives are commonly characterized as conveying different degrees of distance with respect to the deictic center (the speaker): *este* ('this') is proximal, *ese* ('that') medial, and *aquel* ('that yonder') is the distal demonstrative of the tripartite system. In addition, Spanish has three demonstrative pronouns (*esto*, *eso* and *aquello*), which do not inflect and have been traditionally labelled as neuter demonstrative pronouns in the Spanish grammatical tradition (even though there is not clearly a neuter grammatical gender in this language). Most likely these pronouns have been labelled as neuter for they are used, as in many other languages, as demonstrative anaphors to anaphorically and cataphorically refer to abstract, genderless, higher order entities like events, propositions, etc. in discourse.

2.1 A first corpus study

To test the referential distance (recency) factor, we carried out a corpus search[2] on the three neuter demonstrative pronouns. To this aim, we obtained a sample of 193 cases to be scrutinized. Out of the total 193 occurrences, 82 were instances of demonstrative *esto*, 80 instances of *eso* and 31 of *aquello*. In order to restrict the high number of occurrences of demonstratives in our corpus we searched for combinations of a demonstrative in the subject position immediately followed by a past tense (e.g. *esto ocurrió*… ('this happened…'). The sample is not even (82 vs 80 vs 31 cases) due to the lesser frequency of occurrence of demonstrative *aquello* in Spanish oral and written discourses.

In order to find out the antecedent (and semantic referent) of the demonstrative, we segmented our sample texts into sentences as in (3). Note that each bracketed item corresponds to a discourse segment (a sentence), and each segment has been numbered with a subscript. The demonstrative anaphor is written in bold characters and the most likely antecedent has been underlined. In (3), for example, the antecedent to the demonstrative pronoun can be found in sentence #3 (subscript CL_3), namely, in the third sentence relative to the position of the demonstrative anaphor in clause CL_0. Antecedent and anaphor are subscripted to show coreference. An English translation of the original Spanish text is given below.

(3) [Al fin y al cabo, si ustedes están aquí es porque quieren que les hable de la **<u>Operación Ópera</u>**$_k$, claro.]$_{CL3}$ [¡Me cuesta tanto volver al pasado!]$_{CL2}$ [Ya comprenderán, el tiempo aquí transcurre de otra manera.]$_{CL1}$ [Y todo **aquello**$_k$ sucedió en el 92, hace ya…]$_{CL0}$ ¡Pero sea! No voy a defraudarles. Les contaré la historia tal y como sucedió.

[After all, it is obvious that you are here because you want me to tell you about the **<u>Operation Opera</u>**]$_{CL3}$ [It's so hard for me to go back to the past!]$_{CL2}$ [You see, here times goes by differently.]$_{CL1}$ [And all **that** happened in 1992, about…]$_{CL0}$ But OK! I won't disappoint you. I'll tell you the story exactly as it happened.

The results from our study indicate that demonstrative pronouns in Spanish show, like English, a strong preference for their antecedents to be found immediately prior to the occurrence of the anaphor. The results are shown in Figure 1. There are 7 different categories. Categories labelled 1CL, 2CL, 3CL and 4CL stand for the

[2] The corpus CREA (Corpus de Referencia del Español Actual) has been the source of data used throughout this paper and for all our corpus samples and illustrative examples. The CREA corpus of Spanish is a very large collection of texts. A dedicated search interface allows the user to search the corpus for words and phrases and display the search result as a concordance with limited context (the sufficient amount of context for the purposes of this paper.) The corpus comprises written texts (newspaper, novels, emails, etc.) as well as transcribed spoken discourse (interviews, speeches, etc.). For the purposes of this paper, we have included both written and spoken discourse in our corpus samples. The corpus is accessible at http://corpus.rae.es/creanet.html

sentence ID number where the antecedent was found (where 1CL is the closest to the anaphor). The category ≥5CL comprises all those cases in which the antecedent was found in the fifth clause preceding the anaphor and up. All those cases in which, for some reason, the proper antecedent could not be determined have been gathered under category N/A (e.g. not enough text to locate the antecedent, the antecedent was ambiguous, etc.). Finally, the category labelled CATAPH stands for cases of cataphora where reference via the demonstrative is made to a textual entity, which is subsequent to the appearance of the anaphor. In all instances of cataphoric reference considered, the antecedent was found in the clause immediately following the one holding the anaphor.

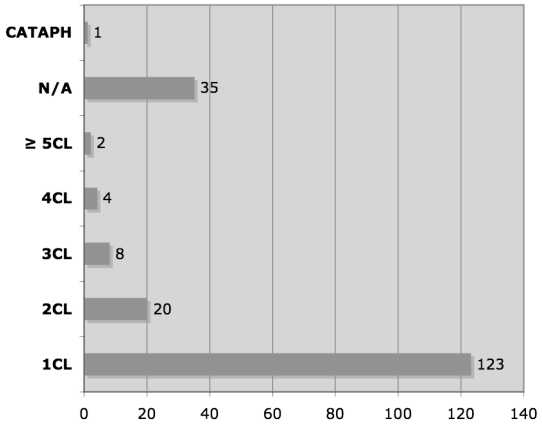


Figure 1: textual distance of antecedent relative to demonstrative anaphor

The results seem to be relatively straightforward: in 123 instances (63.7%) out of the total 193 cases examined, the antecedent was found in the sentence closest to the anaphor (1CL). This percentage increases substantially up to a 77.8% when we leave N/A cases out of the count. On the other hand, there are no relevant differences regarding individual demonstrative pronouns as the three elements obtained pretty similar results: 78% of antecedents of proximal *esto* ('this'), 78.2% of *eso* ('that') and 76.0% of *aquello* ('that yonder') were found in 1CL. Even if we would collapse categories 2CL, 3CL, 4CL and ≥5CL into one single category (with label ≥2CL), the number of antecedents found in the clause closest to the demonstrative anaphor would still be much higher. The results of this study seem to confirm the data by Hegarty et al. (2000) for English, namely that demonstrative pronouns show a strong preference for their clausal antecedents to be introduced in the sentence preceding the one containing the anaphor. Unlike Dutch demonstratives *deze* and *die* (Kirsner et al. (1987), Spanish demonstrative pronouns do not exhibit any individual differences concerning referential distance. This does not entail that Spanish demonstratives do not show any semantic or pragmatic differences. As we will see in the next section, Spanish demonstratives show important differences in the way they are licensed in discourse by certain contextual elements like tenses, even denoting nominals and temporal adverbials.

3 Tense and demonstration in discourse

In the Spanish grammatical tradition, it was noticed that demonstratives and some particular tenses show a correlation in oral and written discourse (Fernández-Ramírez, 1951; cf. also Gutiérrez-Rexach (2002) for a more recent restatement). This parallelism arises when demonstratives are used as discourse anaphors, that is, when reference is intratextual. To date, the alleged correlation tense-demonstration has not been empirically tested. To this aim, we carried out a corpus study to check whether the alleged correlation can be sustained and explained.

The alleged correlation tense-demonstration does not appear to be restricted to Spanish. In Dutch, Kirsner et al. (1987) studied the effect than tense may have on the speaker's choice of demonstratives *deze* ('this') and *die* ('that'). At the intrasentential level, they found that 59% of present tense verb forms co-occur with proximal *deze* and 67% of past tense forms co-occur with distal *die*. These figures are based on the study of 43-*deze* sentences and 42 *die*-sentences containing non-perfect verb forms. Nevertheless, as the authors of the study point out, when context is added, other factors such as referential distance, the degree of detail with which the referent has been described, etc. override the influence of tense on demonstrative choice. It is worth noting that for Kirsner et al.'s the differences in meaning between Dutch demonstratives can be fundamentally explained on the basis of "the degree of attention which the addressee is instructed to give to the referent of the noun" (p. 17).

Regarding Spanish, our initial hypothesis is that contextual clues or elements such as tense, temporal adverbials, event denoting nominals and other temporal expressions favor the use of

certain demonstrative anaphors. In particular, we will hypothesize that distal demonstrative *aquel* ('that yonder'), both in its pronominal and adnominal forms, needs a PAST-denoting contextual element to be licensed in discourse. As a consequence, anaphoric reference with the distal demonstrative *aquel* commonly involves past events, facts, situations, etc. On the other hand, the two other demonstratives of the tripartite system (proximal *este* ('this') and medial *ese* ('that')) do not need any particular contextual configuration to be used as demonstrative anaphors in discourse. In sum, there is a correlation between a PAST trigger and distal *aquel*, whereas *este* and *ese* are found in any context irrespective of their temporal frame in discourse.

We carried out a second corpus study where we analyzed if the alleged correlation tense-demonstration can be sustained on empirical grounds. For this purpose, we searched for occurrences of demonstrative pronouns and past and present tense verb forms where the demonstrative anaphor played the syntactic role of subject of the verb. A sample query is shown in (4). All three demonstrative pronouns were combined with the tensed verb forms: past tense as in (4a) and present tense as in (4b). We limited our search to event predicates ('happen', 'occur' and 'finish'), to ensure the demonstratives in question were referring back in the text to events, which are entities that are commonly referred to via demonstrative anaphors. At the same time, forcing a discourse referential reading for the demonstrative would ensure that we were filtering out other actual uses of Spanish demonstratives as discourse particles.

(4) a. Esto/Eso/Aquello sucedió.
This/That/That yonder happen-3sg.Pret.
'This/That/That yonder happened.'

b. Esto/Eso/Aquello sucede.
This/That/That yonder happen-3sg.Pres.
'This/That/That yonder happen.'

Regarding the tenses scrutinized, we looked for combinations of demonstratives and various PAST (past progressive, Spanish "imperfecto", preterite) and PRESENT[3] tenses (present progressive, simple present and Spanish "pretérito perfecto")[4]. The results are shown in figures 2 and 3. As shown in the chart in 2, only 1 single occurrence of demonstrative *aquello* combined with a present tense was found. Compare these figures with the 573 occurrences of demonstrative pronoun *esto* and 209 occurrences of 'medial' demonstrative *eso*.

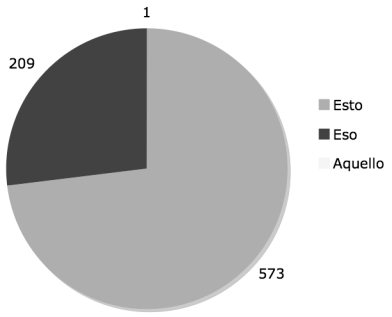


Figure 2: total number of demonstrative plus present tense combinations found in corpus search.

On the other hand, the frequency of tokens of demonstrative *aquello* accompanied by a past tensed verb form increases dramatically as shown in graph 3 (45 occurrences). Thus, out of the total 46 tokens of demonstrative *aquello* found in our corpus search, only 2.17% were cases where the demonstrative was accompanied by a past tense. The other two demonstratives also show a high rate of occurrence along with past tenses in discourse (151 tokens of *eso* vs 290 of proximal *esto*).

The disparity in the total number of occurrences among demonstratives (46 of distal *aquello*; 360 of medial *eso*; 863 of proximal *esto*) clearly indicates that the use of demonstrative *aquel* is in general quite limited in modern Spanish; especially when compared to the overall frequency of use of demonstratives *este* and *ese*. This is a proven fact that equally applies to pronominal and adnominal demonstrative *aquel* when used anaphorically in discourse.

[3] We have adopted in this paper a Reichenbachian view of natural language tense (Reichenbach 1947) whereby the tenses of finite forms situate the event denoted by the semantics of the verb with respect to the time of the speech time (S).

[4] Some of these tenses do not have a direct or exact correspondence in English, while others do. Thus, for example, the Spanish "pretérito perfecto" is quite similar to the present perfect tense in English. While technically a past tense, Spanish pretérito perfecto is commonly characterized as having current relevance (i.e. the event conveyed by the tensed verb is relevant at a time that extends into or overlaps the speech time). For this reason we decided to include this tense in the group of PRESENT tenses.

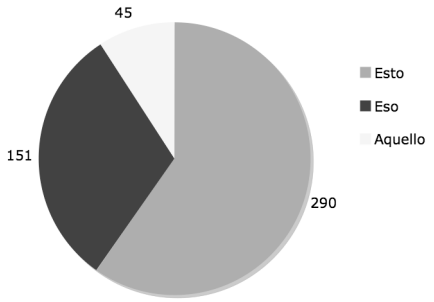


Figure 3: total number of occurrences demonstrative plus past tense found in corpus search.

The numbers shown in figures 2 and 3 indicate that the use of Spanish distal demonstrative *aquello* clearly correlates with past tense in discourse, whereas proximal *esto* and medial *eso* co-occur with past and present tenses in a similar proportion. Proximal demonstrative pronoun *esto* is the element of the tripartite system preferred by speakers for reference to events in discourse. Nevertheless, the picture is not as straightforward as it may appear from these figures. A non systematic look at the corpus reveals abundant cases of demonstrative *aquel* used in discourse along with a past tense in its immediate textual surroundings. A suitable explanation must be provided for these cases or, otherwise, the validity of our study could be questioned and the hypothesized correlation tense-demonstration in Spanish could not be sustained.

In order to test whether the semantic nature of the entity referred to or the specific demonstrative expression (pronominal vs adnominal) may have an effect on the clear correlation tense-demonstration shown by demonstrative *aquel*, we carried out a second corpus study. In this case, we have searched the corpus for occurrences of the expression *aquel hecho* 'that fact' containing the demonstrative in an adnominal use (this/that-NP type). The main goal is to test whether other explicit contextual factors besides tense may have an influence in the licensing of demonstrative *aquello*. We looked at a substantial fragment of text surrounding the demonstrative anaphor (a discourse fragment consisting of an average of 10 sentences). In all, 30 occurrences of the expression *aquel hecho* ('that fact') were scrutinized. The results are given in figure 4. The category acronyms stand for the following elements: NAPT (No Apparent Past Trigger), OTHER, EDN (Event denoting Noun), TE (Temporal Expression), PTPC (Past Tense(s) in Previous Clause(s), PTSC (Past Tense in the Same Clause as the demonstrative anaphor).

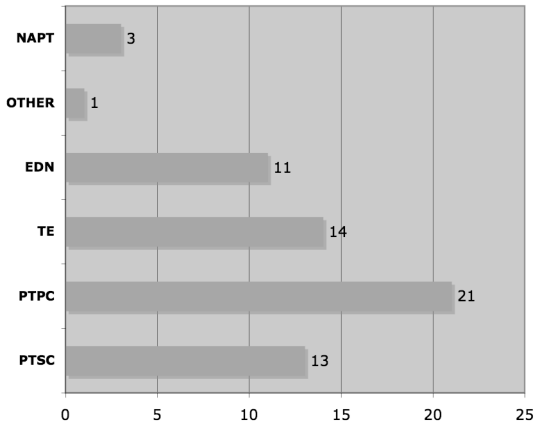


Figure 4: number of potential licensing expressions for demonstrative *aquel* in immediate context.

As figure 4 indicates, in only 3 out of the total 30 tokens analyzed no past temporal expression could be found in the immediate discourse context of the demonstrative anaphor. In all other tokens, at least one past expression 'trigger' could be found. In many cases, more than one past expression co-appeared in the anaphor's immediate context. A series of examples are given in the appendix to this paper. What all the linguistic expressions included in these categories have in common is that they directly or indirectly denote a past time or, in other words, contribute to situate the event referred to at some interval preceding the speech time. Taking discourse as a coherent and fully cohesive semantic unit, what these expressions do is contribute to the setup and maintenance of the temporal referential frame of discourse. Thus, for example, past time denoting temporal expressions (dates; time adverbs like 'tomorrow' and other past time denoting expressions like 'John visited Paris last year') are found in 14 out of the 30 cases analyzed. Some event-denoting nominals like, for example, the Beijing Olympic Games or the murder of President Kennedy may also situate the event they denote at a particular past interval, namely, the summer of year 2008 and year 1963, respectively. Of course, nothing in the morphology or the semantics of these nominals indicates obligatory reference to a past time. Rather, their interpretation as past event-denoting nominals is clearly dependent on the background knowledge of the conversation participants. Thus, if we assume that common ground and a certain amount of shared world

knowledge is essential to communication, event-denoting nominals also contribute to the setup of the referential temporal frame of discourse. Finally, tense is another key contextual factor that contributes to the setup and maintenance of the temporal frame. In 13 of the tokens examined, a past tense could be found in the same clause as the demonstrative. In 21 cases, the past tense verb form(s) was found in the preceding discourse. For example, a good number of the instances analyzed were narratives where a certain past fact, event or action was described using a series of past tenses along a variable textual span.

In principle, what all these elements (past tenses, event denoting nominals, temporal ad-verbials) have in common is their ability to situate the discourse entity referred to at an interval interval preceding the actual speech time. Whether this sort of meaning is wholly procedural or contains a mixture of conceptual and procedural elements is an issue that we will not discuss in this paper. Spanish distal demonstrative *aquel* in discourse anaphoric uses, and perhaps other demonstrative anaphors in different languages, appear to be somehow sensitive to the temporal information conveyed by these elements up to the point that demonstrative *aquel* needs a past denoting 'trigger' in its textual surroundings to be fully licensed in context. As the data presented in this paper indicate, demonstratives *este* and *ese* do not seem to be sensitive to the past/present distinction. This, along with other data like the overall frequency of use of Spanish demonstrative anaphors, can be taken as an indication that the Spanish tripartite system is in the process of evolving into a binary system, an issue we will not discuss here for reasons of space (see Gutiérrez-Rexach 2002 for a general theory along these lines).

The fundamental conclusion that can be drawn from this paper is that a variety of contextual information may contribute to the differential behavior of demonstratives as discourse anaphors. How the particular "licensing" of the tense-demonstration relation in discourse takes place and how to explain and/or characterize it is not a trivial task. Many linguistic expressions are clearly context-dependent in many languages in various ways. In some cases, such dependency can be explained on syntactic grounds (i.e. negative polarity items). In other cases, a suitable explanation has been provided on pragmatic or semantic grounds (the Spanish negative word *tampoco* 'neither/not…either' needs either an overt negation or even a presupposed negation in the textual surroundings for it to be fully licensed in discourse (Schwenter and Zulaica, 2001). Evidence presented in this paper concerning referential-distance preferences for Spanish demonstrative anaphors indicate that the abstract discourse object is commonly found in the clause immediately preceding the anaphor. It has also been shown that other contextual factors need to be taken into account when dealing with demonstrative anaphors cross-linguistically. Thus, for example, Spanish demonstrative anaphor *aquel* strongly correlates with past time denoting expressions to the point that the presence of any of these triggers in the surrounding discourse context is needed for the demonstrative to be licensed. This would seem to indicate that demonstrative *aquel* also binds a time variable thus establishing a referential relation between the anaphor and the temporal information conveyed by texts (along the lines defended by Setzer and Gaizauskas, 2000; Pustejovsky et al. 2003; Hobbs and Pustejovsky, 2006.)

4 Conclusions

It is commonly assumed that demonstratives, when used as anaphors in discourse cross-linguistically, convey additional information besides mere (co)-reference. For example, these elements (and other referential expressions) are said to mark the cognitive status of their antecedents or contribute to the joint attentional state of the participants in the conversation. The source of such supplementary information appears to be of a pragmatic nature: It arises in specific uses and discourse context plays a crucial role in its appearance. In this paper, we have shown that the tripartite system of demonstrative anaphors in Spanish seems to be sensitive to specific temporal contextual information. Thus, the Spanish distal demonstrative anaphor *aquel* requires the presence a past time denoting expression (i.e. past tensed verb forms, adverbs, event-denoting nominals and other temporal expressions) in its contextual environment for the anaphor to be properly used in discourse. In addition, we have also studied the influence that referential distance may have on Spanish demonstrative anaphors and the speaker's preferences for one anaphor over the other.

Nevertheless, our empirical study is far from being exhaustive. Future research would have to explore other contextual factors in more detail, such ase the amount of textual material between the anaphor and the antecedent, the syntactic and semantic type of specific lexical items involved, differences among discourse genres and/or syntactic prominence of the antecedent, etc. as these factors may also help us better understand the complex mechanisms underlying the semantics and pragmatics of discourse anaphors.

References

N. Asher. 1993. *Reference to Abstract Objects in Discourse*. Kluwer.

H. Diessel. 1999. *Demonstratives: Form, Function and Grammaticalization*. Typological Studies in Language 42. Amsterdam: John Benjamins.

H. Diessel. 2006. Demonstratives, joint attention and the emergence of grammar. *Cognitive Linguistics* 17(4), 463-489.

S. Fernández-Ramírez. *Gramática Española. 3.2: El pronombre*. Madrid: Arco Libros, 1986.

J. Gutiérrez-Rexach. 2002. Demonstratives in context. In J. Gutiérrez-Rexach, editor, *From Words ro Discourse*. Amsterdam: Elsevier, 195-238.

J. K. Gundel, N. Hedberg and R. Zacharski. 1993. Cognitive status and the form of referring expressions in discourse. *Language* 69, 274-307.

J. K. Gundel, M. Hegarty and K. Borthen. 2001. Information structure and pronominal reference to clausally introduced entities. In I. Kruijff and M. Steedman, editors, *Proceedings of the Workshop on Information Structure*, European Summer School of Logic, Language and Information: Helsinki, 37-51.

J. K. Gundel, M. Hegarty and K. Borthen. 2003. Cognitive status, information structure and pronominal reference to clausally introduced entities. *Journal of Logic, Language and Information* 12(3), 281-299.

M. Hegarty, J. K. Gundel and K. Borthen, 2001. Information structure and the accessibility of clausally introduced referents. *Theoretical Linguistics* 27, 163-186.

J. Hobbs and J. Pustejovsky. 2006. Annotating and reasoning about time and events. In *Proceedings of the Workshop of Annotating and Reasoning about Time and Events* (ARTE). Association for Computational Linguistics (ACL).

T. A. J. M. Janssen. 1996. Deictic and anaphoric referencing of tenses. In W. de Mulder, L. Tasmowski-De Ryck and C. Vetters, editors, *Anaphores Temporelles et (In)-coherence*. Amsterdam: Rodopi, Cahiers Chronos I, 79-107.

R. S. Kirsner, V. J. Van Heuven and J. F. M. Vermeulen. 1987. Text type, context and demonstrative choice in written Dutch: some experimental data. *Text* 7 (2), 117-144.

J. Pustejovsky, J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer and G. Katz. 2003. TimeML: Robust specification of event and temporal expressions in text. *IWCS-5*, *Fifth International Workshop on Computational Semantics*.

H. Reichenbach. 1947. *Elements of simbolic logic*. London: Macmillan.

S. Schwenter and I. Zulaica. 2001. On the contextual licensing of tampoco. In S. Montrul and F. Ordoñez, eds., *Linguistic Theory and Language Development in Hispanic Languages*, Sommerville: Cascadilla Press, 62-80.

A. Setzer and R. Gaizauskas. 2000. Annotating events and temporal information in newswire texts. In *Proceedings of the Second International Conference on Language Resources and Evaluation* (LREC2000).

Appendix 1.

In this appendix, we present some contextualized examples of the Spanish distal demonstrative *aquel* ('that yonder'). The corresponding English translation is given below each discourse sample. The demonstrative is highlighted in capital letters for an easy identification and the source is given in brackets at the end of every example. Other relevant information is given in bold characters. All the labels (PTPC, PTSC, EDN, TE and Other) correspond to the corpus sample in figure 4.

(I) This example is an instance of the temporal trigger labeled **PTPC (Past tense(s) in previous clause(s)** in our corpus study. The text tells a brief story about the movie *Suddenly*. The series of past tensed verb forms indicate the speaker is narrating a past event, which is the temporal frame that triggers the use of this particular demonstrative.

*"Suddenly" nunca **se estrenó** en España a pesar de que, además de un excelente filme policiaco, **fue** una película premonitoria. Nueve años después de su estreno, Kennedy **fue asesinado** y según dicen, el asesino oficial, Lee Oswald, **se inspiró** en ella. También se dice que cuando ya **se había producido** el magnicidio, su protagonista, Sinatra, **intentó** que la retiraran de la circulación para "evitar que otras mentes insanas la utilizaran como modelo", aunque también para disociar su imagen con AQUEL HECHO.*

*"Suddenly" was never **showed** in Spain in spite of the fact that it **was** a premonitory and excellent cop movie. Nine years after the premiere, Kennedy **was murdered** and, as it is said, the alleged murderer, Lee Oswald, **was inspired** by it. It is said that once the assassination **was committed**, the main character, Sinatra, **tried** to withdraw the film "to prevent that other insane minds could use it as a model", but also to dissociate his image from THAT FACT.*

(CREA: *La Vanguardia*, 31/08/1994)

(II) The following example is ambiguous as it may be categorized as belonging to either category **PTSC** (**Past tense in same clause as anaphor**) or as **EDN** (**Event denoting noun**). On the one hand, a past tense is accompanying the demonstrative expression in the last sentence. On the other hand, demonstrative reference is made to the past event denoting NP *the French Revolution* two clauses before the anaphor. It appears that this NP may be functioning as the element that licenses the use of distal demonstrative *aquel*. In cases like this, where a clear past event denoting NP is found, we believe the event denoting NP prevails over any other past denoting triggering expression. The reason behind this is that NPs are most commonly global or local discourse (sub)-topics. In consequence, we labeled this particular example as **EDN**.

*El editorial que publicó El País el pasado día 19 de Julio lo podría firmar cualquier diario conservador. Sólo le ha faltado una arenga anticomunista y una referencia a lo utópico de las revoluciones. En Nicaragua, y usted lo sabe bien, se ha avanzado mucho en lo que se refiere a libertades y a igualdades. El bicentenario de la **revolución francesa** parece que no ha servido ni siquiera para dejar claro cuál es la más importante de las palabras que **encabezaron** AQUEL HECHO histórico.*

*The leading article, published by El País last July 19th, could be signed by any conservative newspaper. It was only in need of an anti-comunist harangue and a reference to the revolutionary utopia. In Nicaragua, and you know that well, much progress has been made regarding liberty and equality. It appears that the bicentennial of the **French Revolution** was not even useful to make it clear whichword is the most important one among those leading to THAT historical FACT.*

(CREA: *El País*, 01/08/1989)

(III) In this example, there are, technically, no past tenses (Spanish uses subjunctive tenses in this case). There is, though, an expression that clearly contributes to locate the narrative at as past interval, namely, *el año pasado* ('last year'). Therefore, the example has been categorized as **TE** (**temporal expression**) for the purposes of our corpus study.

*Los profesores del instituto Móstoles IV están estos días en plena vorágine. Que **el año pasado** un alumno de 16 años disparara en mitad de un examen de matemáticas una escopeta de caza y el tiro pasara a metro y medio del profesor no*

ha amedrentado a nadie. De AQUEL HECHO, ahora sólo queda una gran satisfacción.

The teachers in the 'Móstoles IV' high school are going through a fuss these days. Nobody was scared by the fact that a 16 year old student fired a hunting rifle in the middle of a math exam ***last year*** *and the bullet missed the teacher by only one and a half meters. Only great relief remains from the outcame of THAT FACT.*

(CREA: *El País*, 1/10/1988)

(IV) This paragraph describes a street where a notable person from Caracas once walked. Most verb forms are in the present tense or conditional though. The explicit contextual element that helps establishing a past temporal frame for this discourse fragment is the date: March 25th 1799. The temporal expression also licenses the use of the distal demonstrative AQUEL HECHO. This is another clear instance of a **TE** (**temporal expression**) trigger.

En el tramo de la calle Mercaderes comprendido entre Obrapía y Lamparilla, donde hoy radica la Casa Simón Bolívar, una tarja recuerda el paso por la ciudad, ***el 25 de marzo de 1799****, de quien sería el más insigne de los caraqueños y el primero de los libertadores de Sudamérica. La embajada de Venezuela y la Universidad de Los Andes se han encargado de que cuantos transiten por esa acera, tengan conocimiento de AQUEL HECHO.*

In the stretch of Mercaderes street that extends from Obrapía to Lamparilla streets, the current location of Casa Simón Bolívar, a plaque honors Bolivar's march through the city ***on March 25th 1799****. He would eventually become the most notable individual born in Caracas and the first of the South American freedom fighters. The Venezuelan embassy and the Los Andes University have done the necessary work so that whoever walks on that sidewalk becomes aware of THAT FACT.*

(CREA: *Granma Internacional*, 09/1997)

(V) This discourse fragment is the only instance included in the category **OTHER** in our study. It is a case of cataphoric use of demonstrative *aquel* whereby forward reference is made to an event introduced into discourse by the clause (underlined) immediately following the demonstrative anaphor. The verb form chosen to introduce this event is the non-finite perfect form *haber dividido* ('have divided'). It appears that the element that triggers the use of the distal demonstrative anaphor is the past tense verb form *demostró* ('showed') accompanying the anaphor.

Clinton es un buen comunicador, capaz de comprender el asesoramiento que le ofrecen, lo trasmite, persuade a la gente, lucha. Ese es el papel de Clinton. Él está persuadido de todo eso que dijo. Pero los dos elementos claves, los dos cerebros del auge económico de los últimos años, los que han aconsejado cómo aprovechar bien las ventajas y privilegios que hoy disfrutan, son Rubin y Greenspan. No hay duda de eso. Creo que los gobiernos influirán de alguna manera sobre el Banco Central de la Unión Europea, lo ***demostró*** *AQUEL HECHO de haber dividido en dos períodos los ocho años que le correspondían al primer Presidente del Banco Central, un alemán.*

Clinton is a good speaker, able to understand the advice he is being offered: he passes it on, he persuades people, he fights. That is Clinton's role. He is convinced of everything he says. But the two key elements, the two brains of the economic growth over the last years, those who advised him on how to take advantage of the benefits and privileges they are enjoying today, are Rubin and Greenspan. That is beyond any doubt. I think that governments will somehow have an influence on the European Community Central Bank, as ***shown*** *by THAT FACT of having divided the eight years that corresponded to the first president of the Central Bank, a German guy, into two periods.*

(CREA: Transcription of press conference: http://www2.cuba.cu/gobierno/discursos)

Prosodic turn-yielding Cues with and without optical Feedback

Caroline Clemens
Technische Universität Berlin
Deutsche Telekom Laboratories
Ernst-Reuter-Platz 7
10587 Berlin
Caroline.Clemens@telekom.de

Christoph Diekhaus
Technische Universität Berlin
Department f. Language and Communication
Straße des 17. Juni 135
10623 Berlin
c.diekhaus@alice-dsl.net

Abstract

The authors present a study of prosodic turn-taking indicators. The aim was to investigate whether some of the prosodic cues increase in quality or quantity if the optical feedback channel in the verbal conversation is missing. For the study we built up an experimental setup in which conversational partners held a conversation once with and once without an optical feedback channel. A detailed transcription of the recorded speech material was segmented into turns. In each turn the topic units were identified and the syllables were labelled. We measured and compared prosodic feature characteristics between turn-final and turn-medial topic units.

1 Introduction

In a verbal conversation the roles of speaker and listener have to be defined. Sacks et al. (1974) stated "minimize gap and overlap" as the first rule for a working turn-taking-mechanism. According to them, the end of turn has to be marked in some way. Since linguistic cues are rarely found, it is obvious that this marking has to be realized by prosodic features. This supposition was corroborated by the findings of Lehiste (1975), that listeners got the ability to identify the position of clauses within a turn, even if the clauses were represented in isolation. In the speaker's turn several prosodic cues are presumed to indicate to the listener whether the speaker wants to keep or end the turn. At points with high speaker switch potential noticeable gestural and mimic cues can be found. It is unknown how important those non-verbal aspects are for the turn-taking indication. The main research question of the presented study is: Do prosodic cues compensate if the optical feedback channel is missing in the verbal conversation?

2 Prosodic and non-verbal turn-taking indicators

Duncan (1972) sorted turn-taking signals by their function. He classified the signals as turn-yielding, turn-demanding (listener), attempt-suppressing, and back-channel-communication (listener response).

We focused on turn-yielding as those signals are easy to locate, and because most found prosodic and non-verbal cues belong to this class. Beattie (1981) and Oreström (1983) showed that a noticeable rising or falling movement of fundamental frequency acts as a prosodic turn-yielding cue. According to Oreström (1983), the final syllable of the turn is lengthened and sometimes the syllable frequency is increased. Duncan (1972) and Oreström (1983) documented a decrease of intensity at the end of a turn.

In addition, non-verbal cues for turn-yielding have been suggested. Kendon (1967) noticed that a speaker often doesn't look at the listener during an utterance but does so at the end of the turn. An explanation is that at those points of the conversation visual contact is required. Exline (1965) discovered that participants in a conversation look at their dialogue partner more often while they are listening. Duncan (1972) found several non-verbal cues in the behaviour of a speaker as turn-yielding signals: Relaxation of a tensed hand-position, completion of a gesture, regression of the torso, and relaxation of the facial expression.

3 Data Retrieval

3.1 Experimental setup

In our experiment speakers held two conversations, both in two conditions: first with eye-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 107–110,
Queen Mary University of London, September 2009.

contact and then without. Speakers didn't know each other. The given task was to plan a party by seating guests on a map of the party location. For solving the task it was necessary that the conversational partners share their information.

3.2 Preparation of recorded speech material

There were four speakers in two bilateral conversations. During the first half of a conversation the speakers could see each other. After they had accomplished half of the task a screen foreclosed eye-contact. The recordings were transliterated into orthographic text by a phonetic expert. This detailed transliteration contains information like word fragments, hesitations, pauses, and vocal events like laughter. The transcribed text was then segmented into turns. In each turn the topic units were identified according to our definition:

- A topic unit can be considered as semantically and grammatically complete and
- there is no further division possible in grammatically and semantically complete units.

Table 1 shows the number of topic units we found in our material. The syllables were labelled and the F_0-contours were determined by manual judgment.

Table 1: Numbers of topic units for each speaker.

	Speaker No.	Condition 1	Condition 2
Number of Topic Units in turn-final Position	1	25	30
	2	17	24
	3	23	24
	4	20	25
Number of Topic Units turn-medial Position	1	92	150
	2	59	94
	3	86	129
	4	79	95

3.3 Acoustic Measurements

In the analysis of the acoustic speech signal we focused on features that have been suspected as turn-yielding signals in former studies. Each end of a topic unit has the potential to be the end of the turn and to initiate a turn taking. We assumed that a speaker marks topic units in turn-final position compared to turn-medial topic units by prosodic differences and that those differences change if the optical feedback channel is missing.

We observed the following prosodic features:

- Speech rate (syllables per second)
- Average intensity across topic units
- Difference of intensity of final last three syllables and non-final last three syllable of topic units (in Hertz)
- Mean F0 in topic units (in Hertz)
- Mean range of F0 in topic units (in Hertz)
- Difference in duration between final and non-final syllables of topic units (in ms)
- Relative distribution of five different closing F0-contours in the topic units
- Characteristic F0-values of five different closing F0-contours (manual judgment)

4 Findings

We intended to examine whether the differences between turn-final and turn-medial topic units differ in the feature characteristics between the two conditions. Feature characteristics could differ in quality or quantity. For a variation in quantity the number of potential signals would increase or decrease between the two conditions. A variation in quality could only be analyzed if the potential signal appears in both conditions and occurs as an increase or decrease of the strength of the feature.

4.1 Duration

The mean syllable rate of final topic unit and non-final topic unit was compared. Our results indicate an increased syllable frequency at the end of the turn in condition 2. But there is no significant difference between the two conditions. One speaker even decreased syllable frequency in turn-final positions compared to non-turn-final positions.

4.2 Intensity

For the intensity we analyzed differences between

- the overall intensity of the topic units in final and in non-final position, and
- the internal reduction of intensity at the end of the topic units in final and non-final position (by comparing the last three syllables to the others).

The overall intensity of topic units in turn-final position is significantly decreased for two (of four) speakers in the condition with sight and for three speakers in the condition without sight. That is, there seems to be a signal function which

is used by one more speaker in condition 2. However, this is just a quantitative difference between the two conditions. For the speakers, using this potential signal in both conditions, there's no detectable qualitative variation in condition 2 (no enhanced difference between the intensity of topic units in final and non-final position).

For all topic units a decrease of intensity at the end has been found. Due to the decrease of air pressure during an utterance this was expected. This reduction of intensity is for all four speakers only significant for topic units in final position. That is, that in topic units at the end of a turn the final reduction of intensity is much greater than in the other topic units. One could assume a signal function. Further analyses showed that this distinction is intensified by two of the speakers in condition 2, while it is weaker for the other two speakers. The modifications in condition 2 don't have a mutual direction.

4.3 Fundamental frequency

Concerning the fundamental frequency, we examined the following issues by comparing the two conditions:

- The over-all F0-mean and F0-range of the topic units in final and non-final position
- The percentage distribution of final F0-contours at the end of the topic units in final and non-final position
- The representative last F0-values of these contours in final and non-final position (last level tone for movements and F0-mean for sustained F0).

The speakers (exception is one speaker in condition 2) realized the turn-final topic units with lower fundamental frequency; which is significant only for two speakers in both conditions. These two speakers made a stronger distinction between final and non-final topic units in condition 2 by increasing the difference of mean F_0. The other two speakers diminish this distinction in condition 2.

Equivalent is the finding for the F_0-range. The same two speakers who decreased the mean frequency decreased also the F_0-range in final topic units. For these speakers there's also a noticeable intensification of the distinction in condition 2, while the other speakers behave contrarily.

Analyzing the percentage distribution of F_0-movements, we distinguished five F_0-contours at the end of the topic units: *Sustain, Fall, Rise, Rise/fall,* and *Fall/rise*.

None of these contours seemed to appear more often in turn-final position. This includes the falling and rising F_0-movements, which were assumed to be turn-yielding signals. In contrast most of the topic units where realized with a final *sustain* and there was no higher occurrence of *rise* or *fall* in turn-final position detectable.

For the final level tone (F_0-mean for *sustain*) we found only for *sustain*, *fall* and *rise/fall* differences between final and non-final topic units. These contours had lower final level tones (lower F_0-mean for *sustain*) in turn-final position for at least one speaker in condition 1 and 2 speakers in condition 2. These findings accounted for the general lowering of F_0 in final topic units. Although this distinction between final and non-final position doesn't change qualitatively between conditions, there's some evidence for a quantitative change, because more speakers seem to use these signals in condition 2.

5 Discussion and Conclusion

To examine whether turn-yielding signals are intensified in the condition without sight, we constituted the criterion that a cue has to appear in one of the conditions for at least three of the speakers to be considered. For those cues we developed a comparison chart in which qualitative and quantitative changes between condition 1 and 2 (with and without sight) were inscribed (Table 2).
Qualitative differences could only occur if a speaker shows the signal in both conditions. They are treated dichotomous (as increased and decreased). Quantitative differences are marked as added or omitted signals for each speaker.

Table 2 shows that none of the signals undergoes changes of the same direction for more than two speakers. For syllable frequency there's no change between the conditions at all. For intensity of topic units, F_0-mean of *sustain*, and last level-tone of *fall* there's only a quantitative change for one speaker. That is, one more speaker added this signal in condition 2, while one or more speaker use it in both conditions, without qualitative shades.

Only the last level-tone of *rise/fall* and the difference of intensity between final and non-final syllable groups show changes for more than one speaker. But while the difference of intensity between final and non-final syllable groups is increased in condition 2 for the conversation

partners in group 1, it is decreased for group 2. The results cancel each other.

Only the last level-tone of *rise/fall* was modified in condition 2 by more than one speaker without being modified by other speakers in the contrary way. That is, it was added. For the mean F_0 and the F_0-range one speaker omitted the signals in condition 2 and one speaker increased their distinctive function.

Taking a look at the sum of shown signals, we recognize that for none of the speakers there's a remarkable raise in the total number of shown signals in condition 2. Finally, every increased distinctive function of a signal, which could be judged as compensation, has a negative counterpart like decrease or omission. Based on the results of this study one can't assume, that the analyzed prosodic cues compensate if the optical feedback channel is missing. This leaves the question whether optical cues are necessary signals or just added as redundant indicators to intensify the effect of the prosodic cues.

Table 2: Main results of the analysis of some analyzed prosodic features.

	Comparison of feature characteristics between turn-final and turn-medial topic units											
	Condition 1: with sight				Condition 2: without sight				Change from Condition 1 to 2			
	Conversation 1		Conversation 2		Conversation 1		Conversation 2		Conversation 1		Conversation 2	
	TP	MB	SK	SI	TP	MB	SK	SI	TP	MB	SK	SI
Syllable frequency	+	-	+	+	+	-	+	+	.	.	.	.
Intensity	-	+	-	+	+	+	-	+	++	.	.	.
Difference of intensity betwen final and non-final last three syllables	+	+	+	+	+	+	+	+	i	i	d	d
Mean fundamental frequency	+	+	+	-	+	+	-	-	.	i	--	.
Mean range of fundamental frequency	+	+	+	-	+	+	-	-	.	i	--	.
Mean fundamental frequency *sus*	+	-	-	-	+	-	+	-	.	.	++	.
Last level tone *fall*	+	-	-	-	+	-	+	-	.	.	++	.
Last level tone *rise/fall*	-	-	-	+	+	+	-	+	++	++	.	.
Sum	7	5	4	5	8	5	5	5				

+/-	indicator for signal function found/not found
.	no difference found
++	signal function added
--	signal function ommited
i/d	signal function increasing/decreasing

Acknowledgement

We thank the test speakers for volunteering.

References

Adam Kendon. 1967. *Some functions of gaze-direction in social interaction.* Acta Psychologica 26: 22-63.

Bengt Oreström. 1983. *Turn-taking in English Conversation.* Lund studies in english 66, Infotryck AB, Malmö.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. *A simplest systematics for the organisation for turn-taking for conversations.* Language 50: 396-735.

Ilse Lehiste. 1975. *The phonetic structure of paragraphs.* Structure and process in speech perception. Proceedings of the symposium of dynamic aspects of speech perception. Cohen / Nooteboom (Eds.), Springer Berlin / Heidelberg. 195-206.

Ralph V. Exline, David Gray, and Dorothy Schuette. 1965. *Visual behavior in a dyad as affected by interview content and sex of respondent.* Journal of Personality and Social Psychology 1: 201-209.

Starkey Duncan, Jr.. 1972. *Some signals and rules for taking speaking turns in conversations*. Journal of Personality and Social Psychology 23: 283-292.

W. Geoffrey Beattie. 1981. The regulation of speaker turns in face-to-face conversation: Some implications for conversation in sound-only communication channels. Semiotica 34: 55-70.

Exploring Miscommunication and Collaborative Behaviour in Human-Robot Interaction

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

This paper presents the first step in designing a speech-enabled robot that is capable of natural management of miscommunication. It describes the methods and results of two WOz studies, in which dyads of naïve participants interacted in a collaborative task. The first WOz study explored human miscommunication management. The second study investigated how shared visual space and monitoring shape the processes of feedback and communication in task-oriented interactions. The results provide insights for the development of human-inspired and robust natural language interfaces in robots.

1 Introduction

Robots are now escaping laboratory and industrial environments and moving into our homes and offices. Research activities have focused on offering richer and more intuitive interfaces, leading to the development of several practical systems with Natural Language Interfaces (NLIs). However, there are numerous open challenges arising from the nature of the medium itself as well as the unique characteristics of Human-Robot Interaction (HRI).

1.1 Miscommunication in Human-Robot Interaction

HRI involves embodied interaction, in which humans and robots coordinate their actions sharing time and space. As most speech-enabled robots remain in the labs, people are generally unaware of what robots can understand and do resulting in utterances that are out of the functional and linguistic domain of the robot. Physical copresence will lead people to make strong but misplaced assumptions of mutual knowledge (Clark, 1996), increasing the use of underspecified referents and deictic expressions. Robots operate in and manipulate the same environment as humans, so failure to prevent and rectify errors has potentially severe consequences. Finally, these issues are aggravated by unresolved challenges with automatic speech recognition (ASR) technologies. In conclusion, miscommunication in HRI grows in scope, frequency and costs, impelling researchers to acknowledge the necessity to integrate miscommunication in the design process of speech-enabled robots.

1.2 Aims of study

The goal of this study is two-fold; first, to incorporate "natural" and robust miscommunication management mechanisms (namely, prevention and repair) into a mobile personal robot, which is capable of learning by means of natural language instruction (Lauria et al., 2001). Secondly, it aims to offer some insights that are relevant for the development of NLIs in HRI in general. This research is largely motivated by models of human communication. It is situated within the language-as-action tradition and its approach is to explore and build upon how humans manage miscommunication.

2 Method

We designed and performed two rounds of Wizard of Oz (WOz) simulations. Given that the general aim of the study is to determine how robots should initiate repair and provide feedback in collaborative tasks, the simulations departed from the typical WOz methodology in that the wizards were also naive participants. The domain of the task is navigation. In particular, the user

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 111–119,
Queen Mary University of London, September 2009.

guided the robot to six designated locations in a simulated town. The user had full access to the map whereas the wizard could only see the surrounding area of the robot. Thus, the wizard relied on the user's instructions on how to reach the destination. In this section we outline the aim and approach of each WOz study, the materials used and the experimental procedure. Sections 4 and 5 focus on each study individually and their results.

2.1 The first WOz study

This study is a continuation of previous work by the authors (Koulouri and Lauria, 2009). In that study, the communicative resources of the wizard were incrementally restricted, from "normal" dialogue capabilities towards the capabilities of a dialogue system, in three experimental conditions:

- The wizard simulates a super-intelligent robot capable of using unconstrained, natural language with the user (henceforth, Unconstrained Condition).
- The wizard can select from a list of default responses but can also ask for clarification or provide task-related information (henceforth, Semi-Constrained condition).
- The wizard is restricted to choose from a limited set of canned responses similar to a typical spoken dialogue system (SDS).

The current study investigates the first two conditions and presents new findings.

2.2 The second WOz study

The second round of WOz experiments explored the effects of monitoring and shared visual information on the dialogue.

2.3 Set-up

A custom Java-based system was developed and was designed to simulate the existing prototype (the mobile robot). The system consisted of two applications which sent and received coordinates and dialogue and were connected using the TCP/IP protocol over a LAN. The system kept a log of the interaction and the robot's coordinates.

The user's interface displayed the full map of the town (Figure 1). The dialogue box was below the map. Similar to an instant messaging application, the user could type his/her messages and see the robot's responses appearing on the lower part of the box. In the first WOz study, the user's interface included a small "monitor" on the upper right corner of the screen that displayed the current surrounding area of the robot, but not the robot itself. Then, for the purposes of the second study, this feature was removed (see Figure 1 in Appendix A).

Figure 1. The user's interface.

The wizard's interface was modified according to the two experimental conditions. For both conditions, the wizard could only see a fraction of the map- the area around the robot's current position. The robot was operated by the wizard using the arrow keys on the keyboard. The dialogue box of the wizard displayed the most recent messages of both participants as well as a history of the user's messages. The buttons on the right side of the screen simulated the actual robot's ability to remember previous routes: the wizard clicked on the button that corresponded to a known route and the robot automatically executed. In the interface for the Unconstrained condition, the wizard could freely type and send messages (Figure 2).

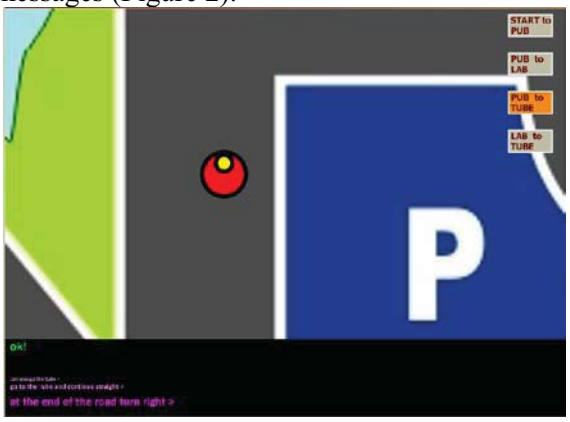


Figure 2. The wizard's interface in the Unconstrained condition.

In the version for the Semi-Constrained condition, the wizard could interact with the user in two ways: first, they could click on the buttons, situated on the upper part of the dialogue box, to automatically send the canned responses, "Hel-

lo", "Goodbye", "Yes", "No", "Ok" and the problem-signalling responses, "What?", "I don't understand" and "I cannot do that". The second way was to click on the "Robot Asks Question" and "Robot Gives Info" buttons which allowed the wizard to type his/her own responses (see Figure 2 in Appendix A).

2.4 Procedure

A total of 32 participants were recruited, 16 users and 16 wizards. The participants were randomly assigned to the studies, experimental conditions and to the roles of wizard or user. The pairs were seated in different rooms equipped with a desktop PC. The wizards were given a short demonstration and a trial period to familiarise with the operation of the system and were also informed about whether the users would be able to monitor them. The users were told that they would interact with a robot via a computer interface; this robot was very fluent in understanding spatial language and could give appropriate responses, it could learn routes but had limited vision. The users were asked to begin each task whenever they felt ready by clicking on the links on their computer screen, start the interaction with "Hello", which opened the wizard's application, and end it with "Goodbye" which closed both applications. The participants received verbal and written descriptions of the experiment. They were not given any specific guidelines on how to interact or what routes to take. However, the users were asked not to use directions such as "north", "south", "up", "down" etc. The wizards were also advised not to try to "sound like a robot", but respond normally.

3 Data analysis

The dialogues collected were annotated on the basis of dialogue acts (DAs), miscommunication, and task status. An example of an annotated dialogue is shown in Table 1.

ID	(x,y@TIME) MESSAGE	T.S.	MISC	HCRC TAG	
U1	(899,445@10:44:34) go straight on.			instruct	
R2	(880,286@10:44:59) There is a wall straight on.		Rej	explain	Imp
U3	(884,286@10:45:28) ok turn left here then take the third right			acknowledge/ instruct	
R4	(352,216@10:45:58) Before the bridge?		Non	check	Imp
U5	(351,68@10:46:17) this is the wrong place	WE	Corr	explain	
R6	(351,68@10:46:39) Wrong place. Should Robot go back?			acknowledge/ query-yn	
U7	(351,68@10:46:55) go forward and at the crossroads keep going forward and the tube is at the end of the road			instruct	
R8	(351,0@10:47:14) Out of bounds.	WE		explain	
R9	(351,608@10:47:47) Where to go?			query-w	
U10	(364,608@10:48:12) the tube is in front of you			instruct	
R11	(402,547@10:48:23) Is it this one?	BOT		query-yn	
U12	(402,547@10:49:7) yes it is.	SUC		reply-y	

Table 1. Example of an annotated dialogue. ID denotes the speaker (User or Robot), T.S. stands for task status and MISC for miscommunication.

3.1 Annotation of dialogue acts

The DAs in the corpus were annotated following the HCRC coding scheme (Carletta et al., 1996). Motivated by Skantze (2005), the last column in Table 1 contains information on the explicitness of the response. This feature was only relevant for repair initiations by the wizards. For instance, responses like "What?" and the the ones in Table 3 were considered to be explicit (EX) signals of miscommunication, whereas lines 2 and 4 in the dialogue above were labelled as implicit (IMP).

3.2 Annotation of task execution status

The coordinates (x,y) of the robot's position recorded for every exchanged message were placed on the map of the town (of dimensions 1024x600 pixels) allowing the analysts to retrace the movements of the robot. Wrong executions (WE) were determined by juxtaposing the user's instruction with the robot's execution, as indicated by the coordinates. Back-on-Track (BOT) was tagged when the first user instruction after a wrong execution was executed correctly. Finally, task success (SUC) was labelled when the robot reached the destination and it was confirmed by the user.

3.3 Annotation of miscommunication

The annotation of instances of miscommunication in the dialogues is based on the definitions given by Hirst et al. (1994). Miscommunication includes three categories of problems: misunderstandings, non-understandings and misconceptions. First, misunderstandings occur when the hearer obtains an interpretation which is not aligned to what the speaker intended him/her to obtain. In this study, without attempting to unveil the intention of the user, misunderstandings were

tagged when the user (who was monitoring the understanding) signalled a wrong execution (see line 5 in Table 1). These correction tags (Corr) did not always coincide with wrong execution tags, but were used when the user became aware of the error (after receiving visual or verbal information). Following the same definition, misunderstandings were also tagged as rejections (tag: Rej) when the wizard expressed inability to execute the instruction (for instance, given the robot's current location, as shown in line 2 in the dialogue), although he/she was able to interpret it. Secondly, non-understandings (tag: Non, line 4) occurred when the wizards obtained no interpretation at all or too many. Non-understandings also included cases in which wizards were uncertain about their interpretation (as suggested by Gabsdil, 2003). Lastly, misconceptions happen when the beliefs of the interlocutors clash, and are outside the scope of this study.

4 First WOz study

Skantze (2005) and Williams and Young (2004) performed variations of WOz studies to explore how humans handle ASR errors, using a real or simulated speech recogniser. They discovered that even after highly inaccurate recognition output, the participants rarely signalled non-understanding explicitly. Accordingly, the experimental hypothesis of the present study is that wizards in both conditions will not choose explicit responses to signal miscommunication (such as "I don't understand" or "What?") but responses that contribute with information.

ASR is a major source of errors in SDS. But as miscommunication is ubiquitous in interaction, there are many other sources of ambiguity that give rise to problematic understanding. Thus, for the current purposes of this work, it was decided that ASR would have an overwhelming effect on the interaction that might prevent the observation of other interesting dialogue phenomena.

This section describes further work on the Unconstrained and Semi-Constrained conditions (see Section 2.1). Twenty participants were recruited and randomly allocated to each condition.

4.1 Results

Analysis of the dialogues of the Unconstrained condition reinforced previous findings and confirmed the experimental hypothesis. In particular, wizards never used explicit repairs, but preferred to describe their location, request clarification and further instructions. Integrating finer classification of clarification requests (CRs) and the original dialogue act tagging, the DAs used by the wizards to signal non-understandings and rejections were categorised as shown in Table 2.

Dialogue Act	Explanation
Explain	The wizard gives description of robot's location. E.g., "I crossed the bridge.", "I am at a crossroad".
Check	This category covers CRs. The corpus contained two types of CRs: first, task-level reformulations (as in line 4 in Table 1), which reformulate the utterance on the basis of its effects on the task, showing the wizard's subjective understanding (Gabsdil, 2003). Second, alternative CRs which occur when the wizard gives two alternative interpretations, trying to resolve referential ambiguity. For instance, "back to the bridge or to the factory", to resolve "go *back* to last location".
Query-w	The wizard asks for further instructions. E.g., "Please give me further instructions."
Explain+Query-w	A combo of actions; the wizard provides information on location and asks for further instructions. E.g., "crossroads, now where?"

Table 2.Wizard DAs after miscommunication.

Figure 3 illustrates the distribution of these responses to signal non-understandings and rejections (columns labelled "Uncons-NON" and "Uncons-REJ", respectively). Evidently, there is a much greater variety of CRs than the two CR types reported here, as described in the work of Purver (2006) and Schlangen (2004). However, for a navigation task and having excluded ASR errors, problems occurred mainly in the meaning recognition level (explained below) and aimed for reference resolution.

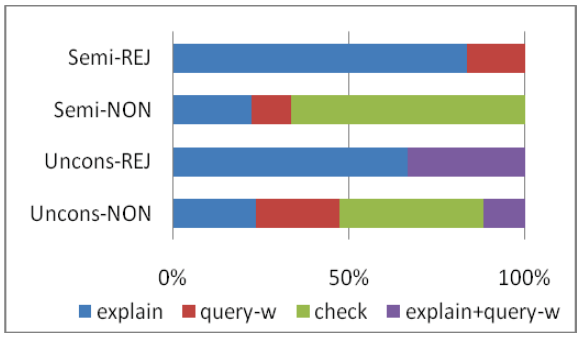


Figure 3. Use of strategies to signal non-understandings or rejections, for either condition.

In conclusion, wizards in the Unconstrained condition did not directly signal problems in understanding but, instead, they attempted to advance the dialogue by providing task-related information in either the form of CRs or simple statements. The study contributes to the findings presented in Skantze (2005) and Williams and Young (2004) in that it demonstrates the use of similar strategies to deal with different sources of problems.

In the Semi-Constrained condition, a degree of restrain and control over the error handling capacity of the wizards was introduced. In particular, the wizards could explicitly signal communication problems in the utterance, meaning and action level using three predefined responses. This is inspired by the models of Clark (1996) and Allwood (1995), according to which, miscommunication can occur in any of these levels and people select repair initiations that point to the source of the problem. The model (adapted from Mills and Healey, 2006) and the responses are schematically shown in Table 3 below.

Levels of Communication		Wizard Responses
Level 1	Securing Attention	-
Level 2	Utterance Recognition	"What?"
Level 3	Meaning Recognition	"Sorry, I don't understand."
Level 4	Action Recognition	"I cannot do that."

Table 3. Levels of communication.

Moreover, based on the classification of the wizard's error handling strategies in the Unconstrained condition (Table 2), we collapsed the observed strategies in two categories of responses which resulted in adding two more error handling buttons; namely, the button denoted as "Robot Asks Question" corresponded to the "Check" and "Query-w" strategies. The "Robot Gives Info" was associated with "Explain". This clear labelling of error handling actions presented to the wizards of the Semi-Constrained condition aimed to "coerce" them to use the strategies in a more transparent way. This could allow us a glimpse to the mechanisms and processes underlying human miscommunication management.

Analysis of the dialogues revealed that in the Semi-Constrained condition wizards employed both explicit and implicit strategies. Figure 4 shows the distribution of explicit and implicit responses to signal non-understandings and rejections. Figure 3 shows the frequency of each implicit strategy to signal non-understandings (Semi-NON) and rejections (Semi-REJ).

The initial prediction was that wizards will not use explicit signals of problems in the dialogue. This was contradicted by the results. It can be argued that the physical presence of the buttons and the less effort required account for this phenomenon. On the other hand, it is also plausible to assume that these strategies matched what the wizards wanted to say. Finally, there were no significant differences between conditions in terms of user experience, task success and time on task (as reported in Koulouri and Lauria, 2009).

Figure 4. Occurrence of implicit and explicit miscommunication signals (Semi-Constrained).

4.2 Discussion and future work

The findings of this study could be extrapolated to HRI. Classification of the responses of the wizards resulted in a limited set of error signalling strategies. Therefore, in the presence of miscommunication the robot could use the static, explicit strategies. But these strategies alone are inadequate (as shown by Koulouri and Lauria, 2009). They need to be supplemented, but not entirely replaced, with dynamic error handling strategies; namely, posing relevant questions and providing descriptions of location. Yet this entails several challenges. Gabsdil (2003) identifies the complexity of adding clarification requests to systems with deep semantic processing. With regard to alternative clarifications, systems would need to generate two alternative interpretations for one referent. Task-level reformulations would also require the system to have the capability to identify the effects of all possible executions of the instruction. As a next step, we will focus on issues concerning the implementation of such functionality.

Schlangen (2004) suggests that "general-purpose" repair initiations, such as "What?", which request repetition of the whole utterance, are more severe for the dialogue compared to reprise fragments (e.g., "Turn where?") that accept part of the utterance. Mills and Healey (2006) also found that "What's" were more disruptive to the dialogue than reprise fragments. Guided by these insights, our current work looks at how each error strategy affects the subsequent unfolding of the dialogue.

5 The second WOz study

Research in human communication has shown that in task-oriented interactions visual information has a great impact on dialogue patterns and improves performance in the task. In particular, Gergle at al. (2004), Clark and Krych (2004) and

Brennan (2005) explored different communication tasks and compared a condition, in which visual and verbal information was available, with a speech-only condition. In their experiments, a person gave instructions to another participant on how to complete a task. Their findings seem to resonate. In terms of time for task completion and number of words per turn, the interactions in the visual information condition were more efficient. The physical actions of the person following the instructions functioned as confirmations and substituted for verbal grounding. Regarding errors, no significant differences were observed between visual and speech-only conditions. Motivated by these findings in human-human interaction, the second study aims to identify the differences in the processes of communication depending on whether the user can or cannot monitor the actions of the robot.

5.1 Experimental design

The study followed a between-subjects factorial design. Experiments were performed for four different conditions, as illustrated in Table 4. The conditions "Monitor, Unconstrained" and "Monitor, Semi-Constrained" were the same as in the first study. Five pairs of participants were recruited to each of the Monitor Conditions and three pairs to each of the No Monitor Conditions.

	Unconstrained	Semi-Constrained
Monitor	Monitor, Unconstrained	Monitor, Semi-Constrained
No Monitor	No Monitor, Unconstrained	No Monitor, Semi-Constrained

Table 4. The design of the 2nd study.

5.2 Results

The data collection resulted in 96 dialogues, 93 of which were used in the analysis. The data were analysed using a two-way ANOVA. All effects that were found to be significant were verified by T-tests. The efficiency of interaction was determined using the following measures: time per task, number of turns, words, miscommunication-tagged turns, wrong executions and task success.

Time per task: The second column of Table 5 displays the average completion time per task in the four conditions. As expected, a main effect of the Monitor factor was found ($F=4.879$, $df=1,11$, $p<0.05$). Namely, when the user could monitor the robot's area the routes were completed faster. The interaction effect between factors was also marginally significant ($F=4.225$, $df=1,11$, $p<0.1$); pairs in the No Monitor, Semi-Constrained condition could not compensate for the lack of visual information and took longer for each task.

Number of turns and words: The aforementioned studies correlate task efficiency with number of turns and words. In terms of the mean number of turns per interaction, no significant differences were found across the groups. Nevertheless, we measured the number of words used per task and in accordance with previous research, we observed that pairs in the No Monitor conditions used more words ($F=4.602$, $df=1,11$, $p=0.05$). However, it was the wizards under the No Monitor conditions that had to be more "talkative" and descriptive ($F=10.324$, $df=1,11$, $p<0.01$). Figure 5 shows the "word-possession" rates attributed to wizards in the four conditions. Moreover, there seems to be a difference ($F=4.397$, $df=1,11$, $p=0.05$) in the mean number of words per turn. In particular, when the wizards' actions were visible to the users, the wizards required fewer words per turn. There is also an interaction effect showing more significant differences between the Monitor, Semi-Constrained condition and the No Monitor, Semi-Constrained condition ($F=5.970$, $df=1,11$, $p<0.05$); in the former, wizards managed with less than 2 words per utterance, taking full advantage of the luxury of the buttons and the fact that they were supervised. In the latter, wizards used more than 6 words per turn.

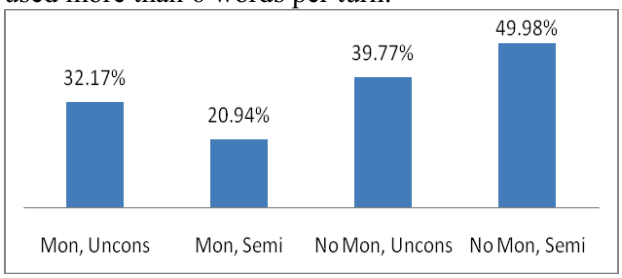


Figure 5. Words used by wizards over total.

Frequency of miscommunication: We measured the number of turns that were tagged as containing miscommunication. Surprisingly, miscommunication rates were much lower in the No Monitor conditions ($F=13.316$, $df=1,11$, $p<0.01$) and not in the conditions in which the user could check at all times the actions and understanding of the robot. The same pattern was found for user-initiated and robot-initiated miscommunication. The rates of miscommunication are included in the third column of Table 5.

Wrong executions: Analysis of number of wrong executions per task reveals a similar effect; wrong executions occurred much less frequently when the wizards were not supervised by

the users (F=6.046, df=1,11, p<0.05). They made on average 1 mistake per task, whereas the average number of wrong executions for the pairs in the Monitor conditions was 5 (fourth column in Table 5).

Task success rates: There were no differences in the number of interrupted or aborted tasks.

Condition	Time per Task (min)	Miscommunication Turns/Total Turns	#Wrong Executions per Task
Mon, Uncons	4.57	8.21%	4.2
Mon, Semi	4.63	8.82%	5.8
No Mon, Uncons	5.67	2.55%	1.0
No Mon, Semi	7.41	1.71%	0.7

Table 5. Summary of results (mean values).

5.3 Discussion and future work

These results are consistent with previous research. The conditions in which the user could see exactly what the robot saw and did resulted in faster task completion and shorter dialogues. However, a finding emerged which was not expected based on the aforementioned studies: in the conditions in which users could not monitor the robot's actions, the wizards were more accurate, leading to low occurrence of wrong executions and miscommunication (see column 3 and 4 in Table 5). The "least collaborative effort" is balanced and compromised against the need to ensure understanding. Thus, wizards provided rich and timely feedback to the users in order to compensate for the lack of visual information. This feedback acted in a proactive way and prevented miscommunication and wrong executions. In the Monitor conditions, asymmetries in perceived responsibility and knowledge between the participants could have encouraged wizards to be less cautious to act. In other words, as the user had access to the full map and the location of the wizard, the wizard felt less "obliged" to contribute to the interaction. However, due to the complex nature of the task, unless the wizard could sufficiently communicate the relevant position of the robot, the directions of the user would more likely be incorrect. It could also be assumed that since visual feedback is instant, the users were also more inclined to issue commands in a "trial and error" process. Irrespectively to the underlying motives, these findings show that despite higher costs in time and word count, linguistic resources were adequate for completing complex tasks successfully. The findings also resonate with the collaborative view of communication. The wizards adapted their behaviour in response to variations in the knowledge state of their partners and made up for the lack of visual information with rich verbal descriptions of their locations.

We are currently performing more experiments to balance the data sets of the study and validate the initial results. Moreover, a fine-grained analysis of the dialogues is under way and focuses on the linguistic content of the interactions. The aim is identical to the first WOz study, that is, to identify the strategies of the wizards in the presence and absence of visual information.

These results have important implications for HRI. As in human collaborative interaction, the robot's communicative actions have direct impact on the actions of the users. In real-world settings, there will be situations in which the users cannot monitor the robot's activities or their information and knowledge are either constrained or outdated. Robots that can dynamically determine and provide appropriate feedback could help the users avoid serious errors. Nevertheless, this is not a straightforward process; providing excessive, untimely feedback compromises the "naturalness" and efficiency of the interaction. The amount and placement of feedback should be decided upon several knowledge sources, combined in a single criterion that is adaptive within and between interactions. These issues are the object of our future work and implementation.

6 Concluding remarks

One of the most valuable but complex processes in the design of a NLI for a robot is enacting a HRI scenario to obtain naturally-occurring data which is yet generalisable and relevant for the future implementation of the system. The present study recreated a navigation scenario in which non-experienced users interacted with and taught a mobile robot. It also simulated two different setups which corresponded to the realistic situations of supervised and unsupervised interaction.
The current trend in the fields of linguistics and robotics is the unified investigation of spatial language and dialogue (Coventry et al., 2009). Exploring dialogue-based navigation of a robot, our study aimed to contribute to this body of research. It can be argued that there were limitations in the simulation as compared to the experimental testing of a real system and, thus, the study was primarily explorative. However, it yielded natural dialogues given that naive "confederates" and no dialogue script were used. The data analysis was more qualitative than quantita-

tive and followed established methods from previous research. Finally, the results of the study matched and extended these findings and provided useful information for the next version of the system as well as some insight into the processes of conversation and social psychology.

The next step in our research is to develop the dialogue manager of the robot to incorporate the feedback and miscommunication management strategies, as observed in the collected data. This holds the promise for a robust NLI that can handle uncertainties arising from language and the environment. However, miscommunication in HRI reaches beyond preventing and repairing recognition errors. Mills and Healey (2008) demonstrate that miscommunication does not inhibit but, on the contrary, it facilitates semantic coordination. Martinovsky and Traum (2003) suggest that through miscommunication, people gain awareness of the state and capabilities of each other. Miscommunication, thus, is seen as an opportunity for communication. Under this light, natural miscommunication management is not only the end, but also the means to shape and advance HRI, so that robots are not tools but partners that play a positive, practical and long-lasting role in human life.

References

Bilyana Martinovsky and David Traum. 2003.The Error Is the Clue: Breakdown in Human-Machine Interaction. *In Proceedings of the ISCA Workshop on Error Handling in Dialogue Systems.*

Dan Bohus and Alexander I. Rudnicky. 2005. Sorry, I Didn't Catch That! – An Investigation of Non-understanding Errors and Recovery Strategies. In *Proceedings of SIGdial2005*. Lisbon, Portugal.

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.

David Schlangen. 2004. Causes and Strategies for Requesting Clarification in dialogue. In *Proceedings of the 5th Workshop of the ACL SIG on Discourse and Dialogue (SIGdial04)*, Boston, USA.

Gabriel Skantze. 2005. Exploring Human Error Recovery Strategies: Implications for Spoken Dialogue Systems. *Speech Communication,* 45(3):207-359.

Graeme Hirst, Susan McRoy, Peter Heeman, Philip Edmonds, Diane Horton.1994. Repairing Conversational Misunderstandings and Nonunderstandings. Speech Communication 15:213–230.

Gregory Mills and Patrick G. T. Healey. 2008. Negotiation in Dialogue: Mechanisms of Alignment. In *Proceedings of the 8th SIGdial workshop on Discourse and Dialogue, Columbus*, OH, USA.

Gregory Mills and Patrick G. T. Healey. 2006. Clarifying Spatial Descriptions: Local and Global Effects on Semantic Co-ordination. In *Procs. of the 10th Workshop on the Semantics and Pragmatics of Dialogue.*

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge, UK.

Herbert H. Clark and Meredyth A. Krych. 2004. Speaking While Monitoring Addressees for Understanding. *Journal of Memory and Language*, 50:62-81.

Jason D. Williams and Steve Young. 2004. Characterizing Task-Oriented Dialog Using a Simulated ASR Channel. *ICSLP.* Jeju, South Korea.

Jean Carletta, Amy Isard, Stephen Isard, Jacqueline Kowtko, Gwyneth Doherty-Sneddon and Anne H. Anderson.1996. HCRC Dialogue Structure Coding Manual(HCRC/TR-82). Human Communication Research Centre, University of Edinburgh.

Jens Allwood. 1995. An Activity based Approach to Pragmatics. Gothenburg Papers in Theoretical Linguistics, 76, Göteborg University, Sweden.

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.

Malte Gabsdil. 2003. Clarification in Spoken Dialogue Systems. In: *Proceedings of 2003 AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, Stanford, USA.

Matthew Purver. 2006. CLARIE: Handling Clarification Requests in a Dialogue System. *Research on Language and Computation* . 4(2-3):259-288.

Stanislao Lauria, Guido Bugmann, Theocharis Kyriacou, Johan Bos and Ewan Klein. 2001. Training Personal Robots Using Natural Language Instruction. *IEEE Intelligent Systems.* 38–45.

Susan E. Brennan . 2005. How Conversation is Shaped by Visual and Spoken Evidence. In J. Trueswell & M. Tanenhaus (Eds.) *Approaches to Studying World-situated Language Use: Bridging the Language-as-product and Language-action Traditions*. 95-129. MIT Press, Cambridge, MA.

Theodora Koulouri and Stanislao Lauria. 2009. A WOz Framework for Exploring Miscommunication in HRI, In *Procs. of the AISB Symposium on New Frontiers in Human-Robot Interaction.* Edinburgh, UK.

Appendix A. Screenshot images of the interface

Figure 1. The interface of the user without the monitor (as used in the second WOz study).

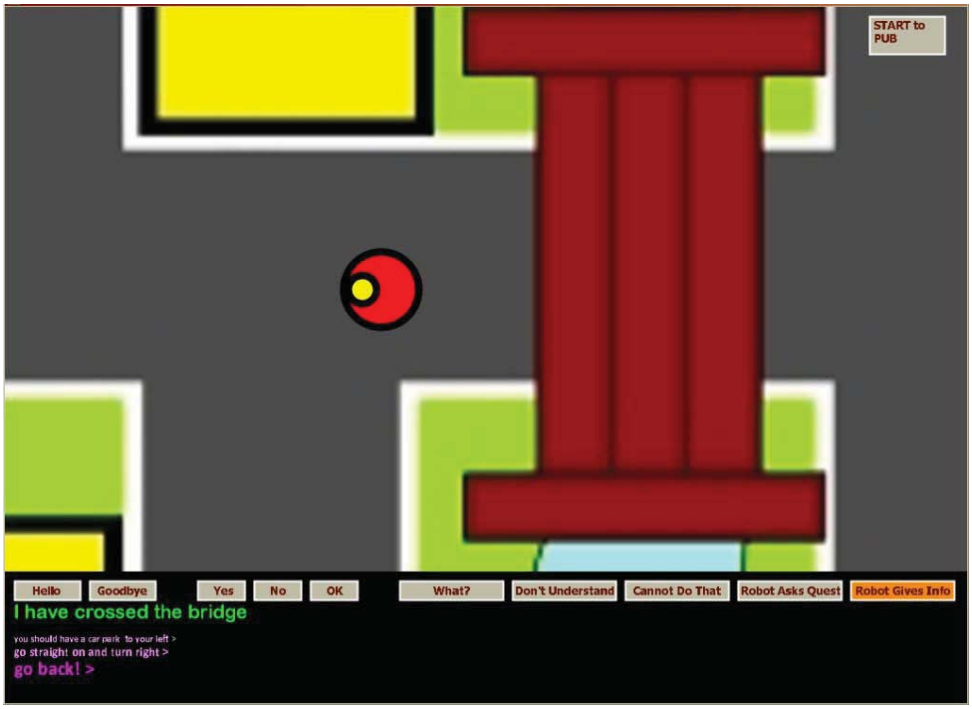


Figure 2. The interface of the wizard in the Semi-Constrained condition.

A Two-tier User Simulation Model for Reinforcement Learning of Adaptive Referring Expression Generation Policies

Srinivasan Janarthanam
School of Informatics
University of Edinburgh
s.janarthanam@ed.ac.uk

Oliver Lemon
School of Informatics
University of Edinburgh
olemon@inf.ed.ac.uk

Abstract

We present a new two-tier user simulation model for learning adaptive referring expression generation (REG) policies for spoken dialogue systems using reinforcement learning. Current user simulation models that are used for dialogue policy learning do not simulate users with different levels of domain expertise and are not responsive to referring expressions used by the system. The two-tier model displays these features, that are crucial to learning an adaptive REG policy. We also show that the two-tier model simulates real user behaviour more closely than other baseline models, using the `dialogue similarity` measure based on Kullback-Leibler divergence.

1 Introduction

We present a new user simulation model for learning adaptive referring expression generation (REG) policies for spoken dialogue systems using reinforcement learning methods. An adaptive REG policy equips a dialogue system to dynamically modify its utterances in order to adapt to user's domain knowledge level. For instance, to refer to domain objects, the system might use simple descriptive expressions with novices and technical jargon with experts. Such adaptations help grounding between the dialogue partners (Issacs and Clark, 1987). Since the user's knowledge level is unknown, the system must be able to adapt dynamically during the conversation. Hand-coding such a policy could be extremely difficult. (Janarthanam and Lemon, 2009b) have shown that such policies can be learned using simulation based reinforcement learning (RL) methods. The quality of such learned policies is directly dependent on the performance of the user simulations used to train them. So far, only hand-coded user simulations have been employed. In contrast, we now present a data driven two-tier user simulation model trained on dialogue data collected from real users. We also show that the two-tier model simulates real users more faithfully than other data driven baseline n-gram models (Eckert et al., 1997).

In section 2 we briefly discuss other work related to user simulations for dialogue policy learning using RL. In section 3 we describe the data used to build the simulation. Section 4 describes the simulation models in detail. In section 5 and 6 we present the evaluation metrics used and the results.

2 Related work

Several user simulation models have been proposed for dialogue management policy learning (Schatzmann et al., 2006; Schatzmann et al., 2007). However, these models cannot be directly used for REG policy learning because they interact with the dialogue system only using high-level dialogue acts. Also, they do not simulate different user groups like experts, novices, etc. In order to learn adaptive REG policies, user simulations need to respond to the system's choice of referring expressions and simulate user groups with different knowledge levels. We propose a two-tier simulation which simulates users with different knowledge levels and is sensitive to the system's choice of referring expressions.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 120–123,
Queen Mary University of London, September 2009.

3 Corpus

The "Wizard-of-Oz" (WOZ) methodology is a widely accepted way of collecting dialogue data for user simulation modeling (Whittaker et al., 2002). In this setup, real users interact with a human wizard disguised as a dialogue system. The wizard interprets the users responses and passes them on to the dialogue system. The dialogue system updates the dialogue state and decides the responses to user's moves. The task of the participant is to interact with the dialogue system to get instructions to setup a broadband Internet connection. The referring expression generation strategy is chosen before the dialogue starts and stays the same for the whole session. The strategies used were "jargon", "descriptive" and "tutorial". In the jargon strategy the system instructs the user using technical terms (e.g. "Plug the `broadband filter` into the `phone socket`."). In the descriptive strategy, it uses descriptive terms (e.g. "Plug the `small white box` into the `square white box on the wall`."). In the tutorial strategy, the system uses both jargon and descriptive terms together. The system provides clarifications on referring expressions when users request them. The participant's domain knowledge is also recorded during the task. Please refer to (Janarthanam and Lemon, 2009a) for a more details on our Wizard-of-Oz environment for data collection. The dialogues were collected from 17 participants (one dialogue each) with around 24 to 35 turns per dialogue depending on the strategy and user's domain knowledge.

4 User Simulation models

The dialogue data and knowledge profiles were used to build user simulation models. These models take as input the system's dialogue act $A_{s,t}$ (at turn t) and choice of referring expressions $REC_{s,t}$ and output the user's dialogue $A_{u,t}$ and environment $EA_{u,t}$ acts. User's observation and manipulation of the domain objects is represented by the environment act.

4.1 Advanced n-gram model

A simple approach to model real user behaviour is to model user responses (dialogue act and environment act) as advanced n-gram models (Georgila et al., 2006) based on many context variables - all referring expressions used in the utterance ($REC_{s,t}$), the user's knowledge of the REs (DK_u), history of clarification requests on the REs (H), and the system's dialogue act ($A_{s,t}$), as defined below:

$$P(A_{u,t}|A_{s,t}, REC_{s,t}, DK_u, H)$$
$$P(EA_{u,t}|A_{s,t}, REC_{s,t}, DK_u, H)$$

Although this is an ideal model of the real user data, it covers only a limited number of contexts owing to the limited size of the corpus. Therefore, it cannot be used for training as there may be a large number of unseen contexts which the model needs to respond to. For example, this model cannot respond when the system uses a mix of jargon and descriptive expressions in its utterance because such a context does not exist in our corpus.

4.2 A Two-tier model

Instead of using a complex context model, we divide the large context in to several sub-contexts and model the user's response based on them. We propose a two-tier model, in which the simulation of a user's response is divided into two steps. First, all the referring expressions used in the system's utterance are processed as below:

$$P(CR_{u,t}|RE_{s,t}, DK_{RE,u}, H_{RE}, A_{s,t})$$

This step is repeated for each expression $RE_{s,t}$ separately. The above model returns a clarification request based on the referring expression $RE_{s,t}$ used, the user's knowledge of the expression $DK_{RE,u}$, and previous clarification requests on the expression H_{RE} and the system dialogue act $A_{s,t}$. A clarification request is highly likely in case of the jargon strategy and less likely in other strategies. Also, if a clarification has already been issued, the user is less likely to issue another request for clarification. In such cases, the clarification request model simply returns `none`.

In the next step, the model returns a user dialogue act $A_{u,t}$ and an environment act $EA_{u,t}$ based on the system dialogue act $A_{s,t}$ and the clarification request $CR_{u,t}$, as follows:

$$P(A_{u,t}|A_{s,t}, CR_{u,t})$$
$$P(EA_{u,t}|A_{s,t}, CR_{u,t})$$

By dividing the complex context into smaller sub-contexts, the two-tier model simulates real users in contexts that are not directly observed in the dialogue data. The model will therefore respond to system utterances containing a mix of REG strategies (for example, one jargon and one descriptive expression in the same utterance).

4.3 Baseline Bigram model

A bigram model was built using the dialogue data by conditioning the user responses only on the system's dialogue act (Eckert et al., 1997).

$$P(A_{u,t}|A_{s,t})$$
$$P(EA_{u,t}|A_{s,t})$$

Since it ignores all the context variables except the system dialogue act, it can be used in contexts that are not observed in the dialogue data.

4.4 Trigram model

The trigram model is similar to the bigram model, but with the previous system dialogue act $A_{s,t-1}$ as an additional context variable.

$$P(A_{u,t}|A_{s,t}, A_{s,t-1})$$
$$P(EA_{u,t}|A_{s,t}, A_{s,t-1})$$

4.5 Equal Probability model baseline

The equal probability model is similar to the bigram model, except that it is not trained on the dialogue data. Instead, it assigns equal probability to all possible responses for the given system dialogue act.

4.6 Smoothing

We used Witten-Bell discounting to smooth all our models except the equal probability model, in order to account for unobserved but possible responses in dialogue contexts. Witten-Bell discounting extracts a small percentage of probability mass, i.e. number of distinct responses observed for the first time (T) in a context, out of the total number of instances (N), and redistributes this mass to unobserved responses in the given context ($V - T$) (where V is the number of all possible responses) . The discounted probabilities P^* of observed responses ($C(e_i) > 0$) and unobserved responses ($C(e_i) = 0$) are given below.

$$P^*(e_i) = \frac{C(e_i)}{N+T} \;\; if(C(e_i) > 0)$$
$$P^*(e_i) = \frac{t}{(N+T)(V-T)} \;\; if(C(e_i) = 0)$$

On analysis, we found that the Witten-Bell discounting assigns greater probability to unobserved responses than to observed responses, in cases where the number of responses per context is very low. For instance, in a particular context, the possible responses, their frequencies and their original probabilities were - `provide_info` (3, 0.75), `other` (1, 0.25), `request_clarification` (0, 0). After discounting, the revised probabilities P^* are 0.5, 0.167 and 0.33. `request_clarification` gets the whole share of extracted probability as it is the only unobserved response in the context and is more than the `other` responses actually observed in the data. This is counter-intuitive for our application. Therefore, we use a modified version of Witten-Bell discounting (given below) to smooth our models, where the extracted probability is equally divided amongst all possible responses. Using the modified version, the revised probabilities for the illustrated example are 0.61, 0.28 and 0.11 respectively.

$$P^*(e_i) = \frac{C(e_i)}{N+T} + \frac{T}{(N+T)V}$$

5 Metrics for evaluation of simulations

While there are many proposed measures to rank user simulation models with respect to real user data (Schatzmann et al., 2005; Georgila et al., 2006; Rieser and Lemon, 2006a; Williams, 2008), we use the `Dialogue Similarity` measure based on Kullback-Leibler (KL) (Cuayahuitl et al., 2005; Cuayahuitl, 2009) divergence to measure how similar the probability distributions of the simulation models are to the original real human data.

5.1 Dialogue Similarity

Dialogue Similarity is a measure of divergence between real and simulated dialogues and can measure how similar a model is to real data. The measure is based on Kullback-Leibler (KL) divergence and is defined as follows:

$$DS(P||Q) = \frac{1}{N}\sum_{i=1}^{N}\frac{D_{KL}(P||Q)+D_{KL}(Q||P)}{2}$$
$$D_{KL}(P||Q) = \sum_{i=1}^{M} p_i * log(\frac{p_i}{q_i})$$

The metric measures the divergence between distributions P and Q in N different contexts with M responses per context. Ideally, the dialogue similarity between two similar distributions is close to zero.

6 Evaluation results

We consider the Advanced N-gram model to be a realistic model of the real human dialogue corpus, as it takes into account all context variables and is reasonably smoothed to account for unobserved user responses. Therefore, we compare the probability distributions of all the other models to

Model	$A_{u,t}$	$EA_{u,t}$
Two-tier	0.078	0.018
Bigram	0.150	0.139
Trigram	0.145	0.158
Equal Probability	0.445	0.047

Table 1: Dialogue Similarity with Modified Witten-Bell discounting w.r.t Advanced N-gram model

the advanced n-gram model using the `dialogue similarity` measure. The results of the evaluation are given in table 1.

The results show that the two-tier model is much closer (0.078, 0.018) to the Advanced N-gram model than the other models. This is due to the fact that the bigram and trigram models don't take into account factors like the user's knowledge, the strategy used, and the dialogue history. By effectively dividing the RE processing and the environment interaction, the two-tier simulation model is not only realistic in observed contexts but also usable in unobserved contexts (unlike the Advanced N-gram model).

7 Conclusion

We have presented a data driven user simulation model called the two-tier model for learning REG policies using reinforcement learning. We have also shown that the two-tier model is much closer to real user data than the other baseline models. We will now train REG policies using the two-tier model and test them on real users in the future.

Acknowledgements

The research leading to these results has received funding from the EPSRC (project no. EP/E019501/1) and 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 British Council's UKERI programme.

References

H. Cuayahuitl, S. Renals, O. Lemon, and H. Shimodaira. 2005. Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. of ASRU 2005*.

H. Cuayahuitl. 2009. *Hierarchical Reinforcement Learning for Spoken Dialogue Systems*. Ph.D. thesis, University of Edinburgh, UK.

W. Eckert, E. Levin, and R. Pieraccini. 1997. User Modeling for Spoken Dialogue System Evaluation. In *Proc. of ASRU97*.

K. Georgila, J. Henderson, and O. Lemon. 2006. User Simulation for Spoken Dialogue System: Learning and Evaluation. In *Proc of ICSLP 2006*.

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 Wizard-of-Oz environment to study Referring Expression Generation in a Situated Spoken Dialogue Task. In *Proc. ENLG'09*.

S. Janarthanam and O. Lemon. 2009b. Learning Lexical Alignment Policies for Generating Referring Expressions for Spoken Dialogue Systems. In *Proc. ENLG'09*.

V. Rieser and O. Lemon. 2006a. Cluster-based User Simulations for Learning Dialogue Strategies. In *Proc. Interspeech/ICSLP*.

J. Schatzmann, K. Georgila, and S. J. Young. 2005. Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems. In *Proc. SIGdial workshop on Discourse and Dialogue '05*.

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*.

S. Whittaker, M. Walker, and J. Moore. 2002. Fish or Fowl: A Wizard of Oz Evaluation of Dialogue Strategies in the Restaurant Domain. In *Language Resources and Evaluation Conference*.

J. Williams. 2008. Evaluating User Simulations with the Cramer-von Mises Divergence. *Speech Communication*, 50:829–846.

Analysis of Listening-oriented Dialogue for Building Listening Agents

Toyomi Meguro, Ryuichiro Higashinaka, Kohji Dohsaka, Yasuhiro Minami, Hideki Isozaki
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237, Japan
{meguro,rh,dohsaka,minami,isozaki}@cslab.kecl.ntt.co.jp

Abstract

Our aim is to build listening agents that can attentively listen to the user and satisfy his/her desire to speak and have himself/herself heard. This paper investigates the characteristics of such listening-oriented dialogues so that such a listening process can be achieved by automated dialogue systems. We collected both listening-oriented dialogues and casual conversation, and analyzed them by comparing the frequency of dialogue acts, as well as the dialogue flows using Hidden Markov Models (HMMs). The analysis revealed that listening-oriented dialogues and casual conversation have characteristically different dialogue flows and that it is important for listening agents to self-disclose before asking questions and to utter more questions and acknowledgment than in casual conversation to be good listeners.

1 Introduction

Although task-oriented dialogue systems have been actively researched over the years (Walker et al., 2001), systems that perform more flexible (less task-oriented) dialogues such as chats are beginning to be actively investigated from their social and entertainment aspects (Bickmore and Cassell, 2001; Higuchi et al., 2008).

This paper deals with dialogues in which one conversational participant attentively listens to the other (hereafter, listening-oriented dialogue). Our aim is to build listening agents that can implement such a listening process so that a user can satisfy his/her desire to speak and have him/herself heard. Such agents would lead the user's state of mind for the better as in a therapy session, although we want our listening agents to help users mentally in everyday conversation. It should also be noted that the purpose of the listening-oriented dialogue is to simply listen to users, not to elicit information as in interviews.

L:	The topic is "travel", so did you travel during summer vacation?	(QUESTION)
S:	I like traveling.	(SELF-DISCLOSURE)
L:	Oh! I see!	(SYMPATHY)
	Why do you like to travel?	(QUESTION)
S:	This summer, I just went back to my hometown.	(SELF-DISCLOSURE)
	I was busy at work, but I'm planning to go to Kawaguchi Lake this weekend.	(SELF-DISCLOSURE)
	I like traveling because it is stimulating.	(SELF-DISCLOSURE)
L:	Going to unusual places changes one's perspective, doesn't it?	(SYMPATHY)
	You said you're going to go to Kawaguchi Lake this weekend. Is this travel?	(QUESTION)
	Will you go by car or train?	(QUESTION)

Figure 1: Excerpt of a typical listening-oriented dialogue. Dialogue acts corresponding to utterances are shown in parentheses (See Section 3.1 for their meanings). The dialogue was originally in Japanese and was translated by the authors.

There has been little research on listening agents. One exception is (Maatman et al., 2005), which showed that systems can make the user have the sense of being heard by using gestures, such as nodding and shaking of the head. Although our work is similar to theirs, the difference is that we focus more on verbal communication instead of non-verbal one.

For the purpose of gaining insight into how to build our listening agents, we collected listening-oriented dialogues as well as casual conversation, and compared them in order to reveal the characteristics of the listening-oriented dialogue. Figure 1 shows an example of a typical listening-oriented dialogue. In the figure, the conversational participants talk about travel with the listener (L), repeatedly asking the speaker (S) to make self-disclosure.

2 Approach

We analyze the characteristics of listening-oriented dialogues by comparing them with casual conversation. Here, casual conversation means a dialogue where conversational participants have no predefined roles (i.e., listeners and speakers). In this

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 124–127,
Queen Mary University of London, September 2009.

study, we collect dialogues in texts because we want to avoid the particular problems of voice, such as filled pauses and interruptions, although we plan to deal with speech input in the future.

As a procedure, we first collect listening-oriented dialogues and casual conversation using human subjects. Then, we label the collected dialogues with dialogue act tags (see Section 3.1 for details of the tags) to facilitate the analysis of the data. In the analysis, we examine the frequency of the tags in each type of dialogue. We also look into the difference of dialogue flows by modeling each type of dialogue by Hidden Markov Models (HMMs) and comparing the obtained models. We employ HMMs because they are useful for modeling sequential data especially when the number of states is unknown. We check whether the HMMs for the listening-oriented dialogue and casual conversation can be successfully distinguished from each other to see if the listening process can be successfully modeled. We also analyze the transitions between states in the created HMMs to examine the dialogue flows. We note that HMMs have been used to model task-oriented dialogues (Shirai, 1996) and casual conversation (Isomura et al., 2006). In this study, we use HMMs to model and analyze listening-oriented dialogues.

3 Data collection

We recruited 16 participants. Eight participated as listeners and the other eight as speakers. The male-to-female ratio was even. The participants were 21 to 29 years old. Each participant engaged in four dialogues: two casual conversations followed by two listening-oriented dialogues with a fixed role of listener/speaker. In listening-oriented dialogue, the listeners were instructed to make it easy for the speakers to say what they wanted to say. When collecting the casual conversation, listeners were not aware that they would be listeners afterwards. Listeners had never met nor talked to the speakers prior to the data collection. The listeners and speakers talked over Microsoft Live Messenger™ in different rooms; therefore, they could not see each other.

In each conversation, participants chatted for 30 minutes about their favorite topic that they selected from the topic list we prepared. The topics were food, travel, movies, music, entertainers, sports, health, housework and childcare, personal computers and the Internet, animals, fashion and games. Table 1 shows the number of collected dialogues, utterances and words in each utterance of listeners and speakers. Generally, utterances in listening-oriented dialogue were longer than those in casual conversation, probably because the subjects explained themselves in detail to make themselves better understood.

		Listening	Casual
# dialogues		16	16
# utterances		850	720
# words per utt.	Listener	20.60	17.92
	Speaker	26.46	21.44

Table 1: Statistics of collected dialogues.

At the end of each dialogue, the participants filled out questionnaires that asked for their satisfaction levels of dialogue, as well as how well they could talk about themselves to their conversational partners on the 10-point Likert scale. The analysis of the questionnaire results showed that, in listening-oriented dialogue, speakers were having a better sense of making themselves heard than in casual conversation (Welch's pairwise t-test; p=0.016) without any degradation in the satisfaction level of dialogue. This indicates that the subjects were successfully performing attentive listening and that it is meaningful to investigate the characteristics of the collected listening-oriented dialogues.

3.1 Dialogue act

We labeled the collected dialogues using the dialogue act tag set: (1) SELF-DISCLOSURE (disclosure of one's preferences and feelings), (2) INFORMATION (delivery of objective information), (3) ACKNOWLEDGMENT (encourages the conversational partner to speak), (4) QUESTION (utterances that expect answers), (5) SYMPATHY (sympathetic utterances and praises) and, (6) GREETING (social cues to begin/end a dialogue).

We selected these tags from the DAMSL tag set (Jurafsky et al., 1997) that deals with general conversation and also from those used to label therapy conversation (Ivey and Ivey, 2002). Since our work is still preliminary, we selected only a small number of labels that we thought were important for modeling utterances in our collected dialogues, although we plan to incorporate other tags in the future. We expected that self-disclosure would occur quite often in our data because the subjects were to talk about their favorite topics and the participants would be willing to communicate about their experiences and feelings. We also expected that the listeners would sympathize often to make others talk with ease. Note that sympathy has been found useful to increase closeness between conversational partic-

	Listener		Speaker	
	Casual	Listening	Casual	Listening
DISC	66.6%	44.5%	53.3%	57.3%
INFO	6.5%	1.4%	5.6%	5.2%
ACK	8.0%	12.3%	6.6%	6.9%
QUES	4.1%	25.8%	21.3%	14.0%
SYM	2.6%	3.7%	3.2%	3.3%
GR	10.9%	9.8%	7.2%	9.6%
OTHER	1.3%	2.5%	2.9%	3.7%

Table 2: Rates of dialogue act tags.

	DISC	INFO	ACK	QUES	SYM	GR
Increase	0	0	8	8	5	4
Decrease	8	8	0	0	3	4

Table 3: Number of listeners whose tags increased/decreased in listening-oriented dialogue.

ipants (Reis and Shaver, 1998).

A single annotator, who is not one of the authors, labeled each utterance using the seven tags (six dialogue act tags plus OTHER). As a result, 1,177 tags were labeled to the utterances in the listening-oriented dialogues and 1,312 tags to those in casual conversation. The numbers of tags and utterances do not match because, in text dialogue, an utterance can be long and may be annotated with several tags.

4 Analysis

4.1 Comparing the frequency of dialogue acts

We compared the frequency of the dialogue act tags in listening-oriented dialogues and casual conversation. Table 2 shows the rates of the tags in each type of dialogue. In the table, OTHER means the expressions that did not fall into any of our six dialogue acts, such as facial expressions and mistypes. Table 3 shows the number of listeners whose rates of tags increased or decreased from casual conversation to listening-oriented dialogue.

Compared to casual conversation, the rates of SELF-DISCLOSURE and INFORMATION decreased in the listening-oriented dialogue. On the other hand, the rates of ACKNOWLEDGMENT and QUESTION increased. This means that the listeners tended to hold the transmission of information and focused on letting speakers self-disclose or deliver information. It can also be seen that the speakers decreased QUESTION to increase self-disclosure.

4.2 Modeling dialogue act sequences by HMM

We analyzed the flow of listening-oriented dialogue and casual conversation by modeling their dialogue act sequences using HMMs. We defined 14 observation symbols, corresponding to the seven tags for a listener and the same number of tags for a speaker.

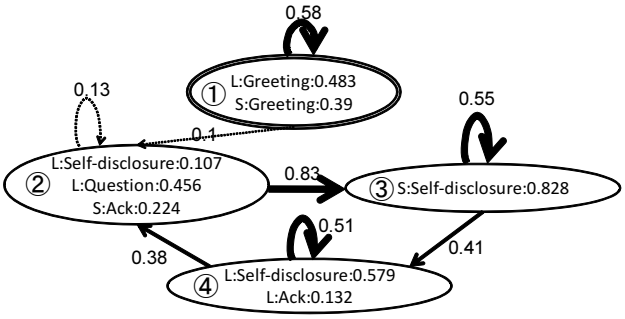


Figure 2: Ergodic HMM for listening-oriented dialogue. Circled numbers represent state IDs.

We trained the following two types of HMMs for each type of dialogue.

Ergodic HMM: Each state emits all 14 observation symbols. All states are connected to each other.

Speaker HMM: Half the states in this HMM only emit one speaker's dialogue acts and the other half emit other speaker's dialogue acts. All states are connected to each other.

The EM algorithm was used to train the HMMs. To find the best fitting HMM with minimal states, we trained 1,000 HMMs for each type of HMM by increasing the number of states from one to ten and training 100 HMMs for each number of states. This was necessary because the HMMs severely depend on the initial probabilities. From the 1,000 HMMs, we chose the most fitting model using the MDL (Minimum Description Length) criterion.

4.2.1 Distinguishing Dialogue Types

We performed an experiment to examine whether the trained HMMs can distinguish listening-oriented dialogues and casual conversation. For this experiment, we used eight listening-oriented dialogues and eight casual conversations to train HMMs and made them classify the remaining 16 dialogues. We found that Ergodic HMM can distinguish the dialogues with an accuracy of 87.5%, and the Speaker HMM achieved 100% accuracy. This indicates that we can successfully train HMMs for each type of dialogue and that investigating the trained HMMs would show the characteristics of each type of dialogue. In the following sections, we analyze the HMMs trained using all 16 dialogues of each type.

4.2.2 Analysis of Ergodic HMM

Figure 2 shows the Ergodic HMM for listening-oriented dialogue. It can be seen that the major flow

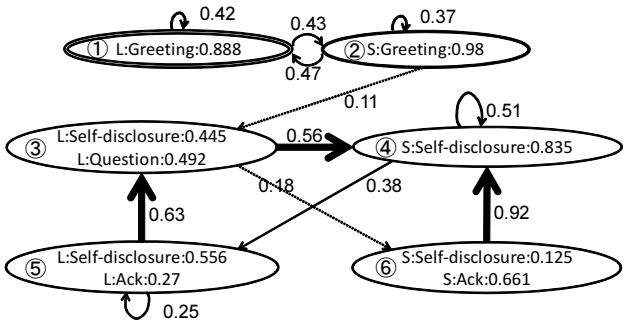


Figure 3: Speaker HMM for listening-oriented dialogue.

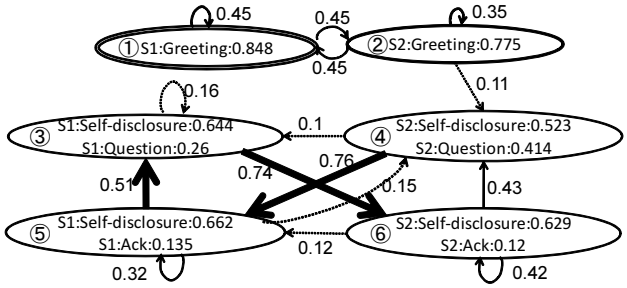


Figure 4: Speaker HMM for casual conversation.

of dialogue acts are: ② L's question → ③ S's self-disclosure → ④ L's self-disclosure → ② L's question. This flow indicates that listeners tend to self-disclose before the next question, showing the cycle of reciprocal self-disclosure. This indicates that listening agents would need to have the capability of self-disclosure in order to become human-like listeners.

4.2.3 Analysis of Speaker HMM

Figures 3 and 4 show the Speaker HMMs for listening-oriented dialogue and casual conversation, respectively. Here, L and S correspond to S1 and S2. It can be clearly seen that the two HMMs have very similar structures. From the probabilities, states with the same IDs seem to correspond to each other. When we compare state IDs 3 and 5, it can be seen that, when speakers take the role of listeners, they reduce self-disclosure while increasing questions and acknowledgment. Questions seem to have more importance in listening-oriented dialogue than in casual conversation, indicating that listening agents need to have a good capability of generating questions. The agents would also need to explicitly increase acknowledgment in their utterances. Note that, compared to spoken dialogue, acknowledgment has to be performed consciously in text-based dialogue. When we compare state ID 4, we see that the speaker starts questioning in casual conversation, whereas the speaker only self-discloses in listening-oriented dialogue. This shows that, in our data, the speakers are successfully concentrating on making self-disclosure in listening-oriented dialogue.

5 Conclusion and Future work

We collected listening-oriented dialogue and casual conversation, and compared them to find the characteristics of listening-oriented dialogues that are useful for building automated listening agents. Our analysis found that it is important for listening agents to self-disclose before asking questions and that it is necessary to utter more questions and acknowledgment than in casual conversation to be good listeners. As future work, we plan to use a more elaborate tag set to further analyze the dialogue flows. We also plan to extend the HMMs to Partially Observable Markov Decision Processes (POMDPs) (Williams and Young, 2007) to achieve dialogue management of listening agents from data.

References

Timothy Bickmore and Justine Cassell. 2001. Relational agents: A model and implementation of building user trust. In *Proc. ACM CHI*, pages 396–403.

Shinsuke Higuchi, Rafal Rzepka, and Kenji Araki. 2008. A casual conversation system using modality and word associations retrieved from the web". In *EMNLP*, pages 382–390.

Naoki Isomura, Fujio Toriumi, and Kenichiro Ishii. 2006. Evaluation method of non-task-oriented dialogue system by HMM. In *Proc. the 4th Symposium on Intelligent Media Integration for Social Information Infrastructure*, pages 149–152.

Allen E. Ivey and Mary Bradford Ivey. 2002. *Intentional Interviewing and Counseling: Facilitating Client Development in a Multicultural Society*. Brooks/Cole Publishing Company.

Dan Jurafsky, Liz Shriberg, and Debra Biasca, 1997. *Switchboard SWBD-DAMSL Shallow-Discourse-Function Annotation Coders Manual*.

Martijn Maatman, Jonathan Gratch, and Stacy Marsella. 2005. Natural behavior of a listening agent. *Lecture Notes in Computer Science*, 3661:25–36.

Harry T. Reis and Phillip Shaver. 1998. Intimacy as an interpersonal process. In S. Duck, editor, *Handbook of personal relationships*, pages 367–398. John Wiley & Sons Ltd.

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

Marilyn A. Walker, Rebecca Passonneau, and Julie E. Boland. 2001. Quantitative and qualitative evaluation of darpa communicator spoken dialogue systems. In *Proc. ACL*, pages 515–522.

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

On NoMatchs, NoInputs and BargeIns: Do Non-Acoustic Features Support Anger Detection?

Alexander Schmitt, Tobias Heinroth
Dialogue Systems Research Group
Institute for Information Technology
Ulm University, Germany
alexander.schmitt@uni-ulm.de
tobias.heinroth@uni-ulm.de

Jackson Liscombe
SpeechCycle, Inc.
Broadway 26
New York City, USA
jackson@speechcycle.com

Abstract

Most studies on speech-based emotion recognition are based on prosodic and acoustic features, only employing artificial acted corpora where the results cannot be generalized to telephone-based speech applications. In contrast, we present an approach based on utterances from 1,911 calls from a deployed telephone-based speech application, taking advantage of additional dialogue features, NLU features and ASR features that are incorporated into the emotion recognition process. Depending on the task, non-acoustic features add 2.3% in classification accuracy compared to using only acoustic features.

1 Introduction

Certainly, the most relevant employment of speech-based emotion recognition is that of a telephone-based Interactive Voice Response System (IVR).

Emotion recognition for IVR differs insofar to "traditional" emotion recognition, that it can be reduced to a binary classification problem, namely the distinction between angry and non-angry whereas studies on speech-based emotion recognition analyze complete and relatively long sentences covering the full bandwidth of human emotions. In a way, emotion recognition in the telephone domain is less challenging since a distinction between two different emotion classes, angry and non-angry, is sufficient. We don't have to expect callers talking to IVRs in a sad, anxious, happy, disgusted or bored manner. I.e., even if a caller is happy, the effect on the dialogue will be the same as if he is neutral. However, there still remain challenges for the system developer such as varying speech quality caused by, e.g., varying distance to the receiver during the call leading to loudness variations (which emotion recognizers might mistakenly interpret as anger). But also bandwidth limitation introduced by the telephone channel and a strongly unbalanced distribution of non-angry and angry utterances with more than 80% non-angry utterances make a reliable distinction of the caller emotion difficult. While hot anger with studio quality conditions can be determined with over 90% (Pittermann et al., 2009) studies on IVR anger recognition report lower accuracies due to these limitations. However, there is one advantage of anger recognition in IVR systems that can be exploited: additional information is available from the dialogue context, the speech recognizer and the natural language parser.

This contribution is organized as follows: first, we introduce related work and describe our corpus. In Section 4 we outline our employed features with emphasis on the non-acoustic ones. Experiments are shown in Section 5 where we analyze the impact of the newly developed features before we summarize our work in Section 6.

2 Related Work

Speech-based emotion research regarding telephone applications has been increasingly discussed in the speech community. While in early studies acted corpora were used, such as in (Yacoub et al., 2003), training and testing data in later studies has been more and more based on real-life data, see (Burkhardt et al., 2008),(Burkhardt et al., 2009). Most studies are limited to acoustic/prosodic features that have been extracted out of the audio data. Linguistic information was additionaly exploited in (Lee et al., 2002) resulting in

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 128–131,
Queen Mary University of London, September 2009.

a 45.7% accuracy improvement compared to using only acoustic features. In (Liscombe et al., 2005) the lexical and prosodic features were additionaly enriched with dialogue act features leading to an increase in accuracy of 2.3%.

3 Corpus Description

For our studies we employed a corpus of 1,911 calls from an automated agent helping to resolve internet-related problems comprising 22,724 utterances. Three labelers divided the corpus into *angry*, *annoyed* and *non-angry* utterances (Cohen's $\kappa = 0.70$ on whole corpus; L1 vs. L2 $\kappa = 0.8$, L1 vs. L3 $\kappa = 0.71$, L2 vs. L3 $\kappa = 0.59$). The reason for choosing three emotion classes instead of a binary classification lies in the hope to find clearer patterns for strong anger. A distinction between non-angry and somewhat annoyed callers is rather difficult even for humans. The final label was defined based on majority voting resulting in 90.2% non-angry, 5.1% garbage, 3.4% annoyed and 0.7% angry utterances. 0.6% of the samples in the corpus were sorted out since all three raters had different opinions. The raters were asked to label "garbage" when the utterance is incomprehensible or consists of non-speech events. While the number of angry and annoyed utterances seems very low, 429 calls (i.e. 22.4%) contained annoyed or angry utterances.

4 Features

We created two different feature sets: one based on typical acoustic/prosodic features and another one to which we will refer as 'non-acoustic' features consisting of features from the Automatich Speech Recognition (ASR), Natural Language Understanding (NLU), Dialogue Manager (DM) and Context features.

4.1 Acoustic Features

The acoustic/prosodic features were extracted with the aid of Praat (Boersma, 2001) and consist of power, mean, rms, mean harmonicity, pitch (mean, deviation, voiced frames, time step, mean slope, minimum, maximum, range), voiced pitch (mean, minimum mean, maximum mean, range), intensity (mean, maximum, minimum, deviation, range), jitter points, formants 1-5, MFCC 1-12. The extraction was performed on the complete short utterance.

4.2 Non-Acoustic Features

The second, i.e. non-acoustic, feature set is based on features logged with the aid of the speech platform hosting the IVR application and is presented here in more detail. They include:

ASR features: raw ASR transcription of caller's utterance (*Utterance*) (unigram bag-of-words); ASR confidence of returned utterance transcription, as floating point number between 0 (least confident) and 1 (most confident) (*Confidence*); names of all grammars active (*GrammarName*); name of the grammar that returned the parse (*TriggeredGrammarName*); did the caller begin speaking before the prompt completed? ('yes', 'no') (*BargedIn*); did the caller communicate with speech ('voice') or keypad ('dtmf') (*InputModeName*); was the speech recognizer successful ('Complete') or not and if it was not successful, an error message is recorded such as 'NoInput' or 'NoMatch' (*RecognitionStatus*)

NLU-Features: the semantic parse of the caller utterance as returned by the activated grammar in the current dialog module (*Interpretation*); given caller speech input, we need to try and recognize the semantic meaning. The first time we try to do this, this is indicated with a value of 'Initial'. If we were not returned a parse then we have to reprompt ('Retry1' or 'Timeout1'). Similar for if the caller asks for help or a repetition of the prompt. Etc. (*LoopName*)

DM-Features: the text of what the automated agent said prior to recording the user input (*PromptName*); the number of tries to elicit a desired response. Integer values range from 0 (first try) to 7 (6th try) (*RoleIndex*); an activity may request substantive user input ('Collection') or confirm previous substantive input ('Confirmation') (*RoleName*); within a call each event is sequentially organized by these numbers (*SequenceID*); the name of the activity (aka dialog module) that is active (*ActivityName*); type of activity. Possible values are: Question, PlatformValue, Announcement, Wait, Escalate (*ActivityType*)

Context-Features: We further developed additional cumulative features based on the previous ones in order to keep track of the NoMatch, NoInputs and similar parameters serving as an indicator for the call quality: number of non-empty NLU parses (*CumUserTurns*); number of statements and questions by the system (*CumSysTurns*); number of questions (*CumSysQuestions*); number of

help requests by the user (*CumHelpReq*); number of operator requests (*CumOperatorReq*); number of NoInput events (*CumNoInputs*); number of NoMatch events (*CumNoMatchs*) number of BargeIns (*CumBargeIns*).

5 Experiments

In order to prevent an adaption of the anger model to specific callers we seperated the corpus randomly into 75% training and 25% testing material and ensured that no speaker contained in training was used for testing. To exclude that we receive a good classification result by chance, we performed 50 iterations in each test and calculated the performance's mean and standard deviation over all iterations.

Note, that our aim in this study is less finding an optimum classifer, than finding additional features that support the distinction between angry and non-angry callers. Support Vector Machines and Artificial Neural Networks are thus not considered, although the best performances are reported with those learning algorithms. A similar performance, i.e. only slightly poorer, can be reached with Rule Learners. They enable a thorough study of the features, leading to the decision for one or the other class, since they produce a human readable set of if-then-else rules. Our hypotheses on a perfect feature set can thus easily be confirmed or rejected.

We performed experiments with two different classes: 'angry' vs. 'non-angry' and 'angry+annoyed' vs. 'non-angry'. Merging angry and annoyed utterances aims on finding all callers, where the customer satisfaction is endangered. In both tasks, we employ a) only acoustic features b) only ASR/NLU/DM/Context features and c) a combination of both feature sets. The number of utterances used for training and testing is shown in Table 1.

As result we expect acoustic features to perform better than non-acoustic features. Among the relevant non-acoustic features we assume as an indicator for angry utterances low ASR confidences and high barge-in rates, which we consider as signal for the caller's impatience. All tests have been performed with the machine learning framework RapidMiner (Mierswa et al., 2006) featuring all common supervised and unsupervised learning schemes.

Results are listed in Table 2, including precision and recall values. As expected, Test B (angry vs. non-angry) has the highest accuracy with 87.23% since the patterns are more clearly separable compared to Test A (annoyed vs. non-angry, 72.57%). Obviously, adding non-acoustic features increases classification accuracy significantly, but only where the acoustic features are not expressive enough. While the additional information increases the accuracy of the combined angry+annoyed task by 2.3 % (Test A), it does not advance the distinction between only angry vs. non-angry (Test B).

	Test A		**Test B**	
	angry+ annoyed	non-a.	angry	non-a.
Training	~ 320	~ 320	~ 80	~ 80
Testing	~ 140	~ 140	~ 40	~ 40

Table 1: Number of utterances employed for both tests per iteration. Since the samples are selected randomly and the corpus was separated by speakers before training and testing, the numbers may vary in each iteration.

5.1 Emotional History

One could expect, that the probability of an angry/annoyed turn following another angry/annoyed turn is rather high and that this information could be exploited. Thus, we further included two features *PrevEmotion* and *PrevPrevEmotion*, taking into account the two previous hand-labeled emotions in the dialogue discourse. If they would contribute to the recognition process, we would replace them by automatically labelled ones. All test results, however, did not improve.

5.2 Ruleset Analysis

For a determination of the relevant features in the non-acoustic feature set, we analyzed the ruleset generated by the RuleLearner in Test A. Interestingly, a dominant feature in the resulting ruleset is 'AudioDuration'. While shorter utterances were assigned to non-angry (about $<$2s), longer utterances tended to be assigned to angry/annoyed. A following analysis of the utterance length confirms this rule: utterances labeled as angry averaged 2.07 (+/-0.73) seconds, annoyed utterances lasted 1.82 (+/-0.57) s and non-angry samples were 1.57 (+/- 0.66) s in average. The number of NoMatch

Test A: Angry/Annoyed vs. Non-angry	only Acoustic	only Non-Acoustic	both
Accuracy	70.29 (+-2.94) %	61.43 (+-2.75) %	72.57 (+-2.37) %
Precision/Recall Class 'Ang./Ann.'	71.51% / 61.57%	68.35% / 42.57%	73.67% / 70.14%
Precision/Recall Class 'Non-angry'	69.19% / 73.00%	58.30% / 80.29%	71.57% / 75.00%
Test B: Angry vs. Non-angry	only Acoustic	only Non-Acoustic	both
Accuracy	87.06 (+-3.76) %	64.29 (+-1.32) %	87.23 (+-3.72) %
Precision/Recall Class 'Angry'	87.13% / 86.55%	66.0% / 58.9%	86.88% / 87.11%
Precision/Recall Class 'Non-angry'	86.97% / 87.53%	62.9% 69.9%	87.55% / 87.33%

Table 2: Classification results for angry+annoyed vs. non-angry and angry vs. non-angry utterances.

events (CumNoMatch) up to the angry turn played a less dominant role than expected: only 8 samples were assigned to angry/annoyed due to reoccurring NoMatch events (>5 NoMatchs). Utterances that contained 'Operator', 'Agent' or 'Help' were, as expected, assigned to angry/annoyed, however, in combination with high AudioDuration values (>2s). Non-angry utterances were typically better recognized: average ASR confidence values are 0.82 (+/-0.288) (non-angry), 0.71 (+/- 0.36) (annoyed) and 0.56 (+/- 0.41) (angry).

6 Conclusion and Discussion

In IVR systems, we can take advantage of non-acoustic information, that comes from the dialogue context. As demonstrated in this work, ASR, NLU, DM and contextual features support the distinction between angry and non-angry callers. However, where the samples can be separated into clear patterns, such as in Test B, no benefit from the additional feature set can be expected. In what sense a late fusion of linguistic, dialogue and context features would improve the classifier, i.e. by building various subsystems whose opinions are subject to a voting mechanism, will be evaluated in future work. We will also analyze why the linguistic features did not have any visible impact on the classifier. Presumably a combination of n-grams, bag-of-words and bag of emotional salience will improve classification.

7 Acknowledgements

We would like to take the opportunity to thank the following colleagues for contributing to the development of our emotion recognizer: Ulrich Tschaffon, Shu Ding and Alexey Indiryakov.

References

Paul Boersma. 2001. Praat, a System for Doing Phonetics by Computer. *Glot International*, 5(9/10):341–345.

Felix Burkhardt, Richard Huber, and Joachim Stegmann. 2008. Advances in anger detection with real life data.

Felix Burkhardt, Tim Polzehl, Joachim Stegmann, Florian Metze, and Richard Huber. 2009. Detecting real life anger. In *Proc. of ICASSP*, April.

Chul Min Lee, Shrikanth Narayanan, and Roberto Pieraccini. 2002. Combining Acoustic and Language Information for Emotion Recognition. In *International Conference on Speech and Language Processing (ICSLP)*, Denver, USA, October.

Jackson Liscombe, Guiseppe Riccardi, and Dilek Hakkani-Tür. 2005. Using Context to Improve Emotion Detection in Spoken Dialog Systems. In *International Conference on Speech and Language Processing (ICSLP)*, Lisbon, Portugal, September.

Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. 2006. Yale: Rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, August.

Johannes Pittermann, A. Pittermann, and Wolfgang Minker. 2009. *Handling Emotions in Human-Computer Dialogues*. Text, Speech and Language Technology. Springer, Dordrecht (The Netherlands).

Sherif Yacoub, Steven Simske, Xiaofan Lin, and John Burns. 2003. Recognition of emotions in interactive voice response systems. In *Proc. Eurospeech, Geneva*, pages 1–4.

Estimating probability of correctness for ASR N-Best lists

Jason D. Williams and Suhrid Balakrishnan
AT&T Labs - Research, Shannon Laboratory, 180 Park Ave., Florham Park, NJ 07932, USA
{jdw,suhrid}@research.att.com

Abstract

For a spoken dialog system to make good use of a speech recognition N-Best list, it is essential to know how much trust to place in each entry. This paper presents a method for assigning a probability of correctness to each of the items on the N-Best list, and to the hypothesis that the correct answer is not on the list. We find that both multinomial logistic regression and support vector machine models yields meaningful, useful probabilities across different tasks and operating conditions.

1 Introduction

For spoken dialog systems, speech recognition errors are common, and so identifying and reducing dialog understanding errors is an important problem. One source of potentially useful information is the *N-Best list* output by the automatic speech recognition (ASR) engine. The N-Best list contains N ranked hypotheses for the user's speech, where the top entry is the engine's best hypothesis. When the top entry is incorrect, the correct entry is often contained lower down in the N-Best list. For a dialog system to make use of the N-Best list, it is useful to estimate the probability of correctness for each entry, and the probability that the correct entry is not on the list. This paper describes a way of assigning these probabilities.

2 Background and related work

To begin, we formalize the problem. The user takes a communicative action u, saying a phrase such as "Coffee shops in Madison New Jersey". Using a language model g, the speech recognition engine processes this audio and outputs an ordered list of N hypotheses for u, $\tilde{\mathbf{u}} = \{\tilde{u}_1, \ldots \tilde{u}_N\}$, $N \geq 2$. To the N-Best list we add the entry $\tilde{u}_*$, where $u = \tilde{u}_*$ indicates that u does not appear on the N-Best list.

The ASR engine also generates a set of K recognition features $\mathbf{f} = [f_1, \ldots, f_K]$. These features might include properties of the lattice, word confusion network, garbage model, etc. The aim of this paper is to estimate a *model* which accurately assigns the $N + 1$ probabilities $P(u = \tilde{u}_n | \tilde{\mathbf{u}}, \mathbf{f})$ for $n \in \{*, 1, \ldots, N\}$ given $\tilde{\mathbf{u}}$ and $\mathbf{f}$. The model also depends on the language model g, but we don't include this conditioning in our notation for clarity.

In estimating these probabilities, we are most concerned with the estimates being *well-calibrated*. This means that the probability estimates we obtain for events should accurately represent the empirically observed proportions of those events. For example, if 100 1-best recognitions are assigned a probability of 60%, then approximately 60 of those 100 should in fact be the correct result.

Recent work proposed a generative model of the N-Best list, $P(\tilde{\mathbf{u}}, \mathbf{f} | u)$ (Williams, 2008). The main motivation for computing a generative model is that it is a component of the update equation used by several statistical approaches to spoken dialog (Williams and Young, 2007). However, the difficulty with a generative model is that it must estimate a joint probability over all the features, $\mathbf{f}$; thus, making use of many features becomes problematic. As a result, discriminative approaches often yield better results. In our work, we propose a discriminative approach and focus on estimating the probabilities *conditioned on* the features. Additionally, under some further fairly mild assumptions, by applying Bayes Rule our model can be shown equivalent to the generative model required in the dialog state update. This is a desirable property because dialog systems using this re-statement have been shown to work in practice (Young et al., 2009).

Much past work has assigned meaningful proba-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 132–135,
Queen Mary University of London, September 2009.

bilities to the top ASR hypothesis; the novelty here is assigning probabilities to *all* the entries on the list. Also, our task is different to *N-Best list re-ranking*, which seeks to move more promising entries toward the top of the list. Here we trust the ordering provided by the ASR engine, and only seek to assign meaningful probabilities to the elements.

3 Model

Our task is to estimate $P(u = \tilde{u}_n|\tilde{\mathbf{u}}, \mathbf{f})$ for $n \in \{*, 1, \ldots, N\}$. Ideally we could view each element on the N-Best list as its own class and train an $(N+1)$-class regression model. However this is difficult for two reasons. First, the number of classes is variable: ASR results can have different N-Best list lengths for different utterances. Second, we found that the distribution of items on the N-Best list has a very long tail, so it would be difficult to obtain enough data to accurately estimate late position class probabilities.

As a result, we model the probability P in two stages: first, we train a (discriminative) model P_a to assign probabilities to just three classes: $u = \tilde{u}_*$, $u = \tilde{u}_1$, and $u \in \tilde{\mathbf{u}}_{\mathbf{2}+}$, where $\tilde{\mathbf{u}}_{\mathbf{2}+} = \{\tilde{u}_2, \ldots, \tilde{u}_N\}$. In the second stage, we use a separate probability model P_b to distribute mass over the items in $\tilde{\mathbf{u}}_{\mathbf{2}+}$:

$$P(\tilde{u}_n = u|\tilde{\mathbf{u}}, \mathbf{f}) = \quad (1)$$
$$\begin{cases} P_a(u = \tilde{u}_1|\mathbf{f}) & \text{if } n = 1, \\ P_a(u \in \tilde{\mathbf{u}}_{\mathbf{2}+}|\mathbf{f}) P_b(u = \tilde{u}_n|\mathbf{f}) & \text{if } n > 1, \\ P_a(u = \tilde{u}_*|\mathbf{f}) & \text{if } n = * \end{cases}$$

To model P_a, multinomial logistic regression (MLR) is a natural choice as it yields a well-calibrated estimator for multi-class problems. Standard MLR can over-fit when there are many features in comparison to the number of training examples; to address this we use ridge *regularized* MLR in our experiments below (Genkin et al., 2005).

An alternative to MLR is support vector machines (SVMs). SVMs are typically formulated including regularization; however, their output scores are generally not interpretable as probabilities. Thus for P_a, we use an extension which re-scales SVM scores to yield well-calibrated probabilities (Platt, 1999).

Our second stage model P_b, distributes mass over the items in the tail of the N-best list ($n \in \{2, \ldots, N\}$). In our exploratory analysis of N-Best lists, we noticed a trend that facilitates modeling this distribution. We observed that the distribution of the *fraction* of the correction position n/N was relatively invariant to N. For example, for both short ($N < 100$) and long ($N \geq 100$) lists, the probability that the answer was in the top half of the list was very similar (see Figure 1). Thus, we chose a continuous distribution in terms of the fractional position n/N as the underlying distribution in our second stage model. Given the domain of the fractional position $[0, 1]$, we chose a Beta distribution. Our final second stage model is then an appropriately discretized version of the underlying Beta, namely, P_b:

$$P_b(u = \tilde{u}_n|\mathbf{f}) = P_b(u = \tilde{u}_n|N) = P_{\text{beta}}(\frac{n-1}{N-1}; \alpha, \beta) - P_{\text{beta}}(\frac{n-2}{N-1}; \alpha, \beta)$$

where $P_{\text{beta}}(x; \alpha, \beta)$ is the standard Beta cumulative distribution function parametrized by α and β. Figure 1 shows an illustration. In summary, our method requires training the three-class regression model P_a, and estimating the Beta distribution parameters α and β.

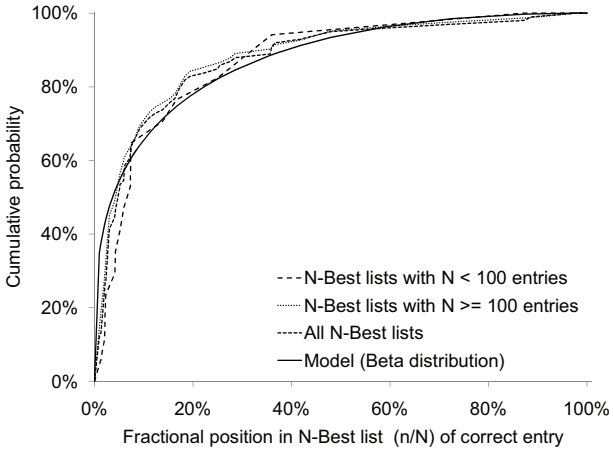


Figure 1: Empirical cumulative distribution of correct recognitions for N-Best lists, and the Beta distribution model for P_b on $1,000$ business search utterances (Corpus 1 training set, from Section 4.)

4 Data and experiments

We tested the method by applying it to three corpora of utterances from dialog systems in the business search domain. All utterances were from

Corpus	WCN	SVM	MLR
1	-0.714	**-0.697**	-0.703
2	-0.251	-0.264	**-0.222**
3	-0.636	-0.605	**-0.581**

Table 1: Mean log-likelihoods on the portion of the test set with the correct answer on the N-Best list. None of the MLR nor SVM results differ significantly from the WCN baseline at $p < 0.02$.[2]

Corpus	WCN	SVM	MLR
1	-1.12	**-0.882**	-0.890
2	-0.821	-0.753	**-0.734**
3	-1.00	**-0.820**	-0.824

Table 2: Mean log-likelihoods on the complete test set. All MLR and SVM results are significantly better than the WCN baseline ($p < 0.0054$).[2]

users with real information needs. Corpus 1 contained $2,000$ high-quality-audio utterances spoken by customers using the Speak4It application, a business search application which operates on mobile devices, supporting queries containing a listing name and optionally a location.[1] Corpus 2 and 3 contained telephone-quality-audio utterances from $14,000$ calls to AT&T's "411" business directory listing service. Corpus 2 contained locations (responses to "Say a city and state"); corpus 3 contained listing names (responses to "OK what listing?"). Corpus 1 was split in half for training and testing; corpora 2 and 3 were split into $10,000$ training and $4,000$ testing utterances.

We performed recognition using the Watson speech recognition engine (Goffin et al., 2005), in two configurations. Configuration A uses a statistical language model trained to recognize business listings and optionally locations, and acoustic models for high-quality audio. Configuration B uses a rule-based language model consisting of all city/state pairs in the USA, and acoustic models for telephone-quality audio. Configuration A was applied to corpora 1 and 3, and Configuration B was applied to corpus 2. This experimental design is intended to test our method on both rule-based and statistical language models, as well as matched and mis-matched acoustic and language model conditions.

We used the following recognition features in $\mathbf{f}$: f_1 is the posterior probability from the best path through the word confusion network, f_2 is the number of segments in the word confusion network, f_3 is the length of the N-Best list, f_4 is the average per-frame difference in likelihood between the highest-likelihood lattice path and a garbage model, and f_5 is the average per-frame difference in likelihood between the highest-likelihood lattice path and the maximum likelihood of that frame on *any* path through the lattice. Features are standardized to the range $[-1, 1]$ and MLR and SVM hyperparameters were fit by cross-validation on the training set. The α and β parameters were fit by maximum likelihood on the training set.

We used the BMR toolkit for regularized multinomial logistic regression (Genkin et al., 2005), and the LIB-SVM toolkit for calibrated SVMs (Chang and Lin, 2001).

We first measure average log-likelihood the models assign to the test sets. As a baseline, we use the posterior probability estimated by the word confusion network (WCN), which has been used in past work for estimating likelihood of N-Best list entries (Young et al., 2009). However, the WCN does not assign probability to the $u = \tilde{u}_*$ case – indeed, this is a limitation of using WCN posteriors. So we reported two sets of results. In Table 1, we report the average log-likelihood given that the correct result is on the N-Best list (higher values, i.e., closer to zero are better). This table includes only the items in the test set for which the correct result appeared on the N-Best list (that is, excluding the $u = \tilde{u}_*$ cases). This table compares our models to WCNs on the task for which the WCN is designed. On this task, the MLR and SVM methods are competitive with WCNs, but not significantly better.

In Table 2, we report average log-likelihood for the entire test set. Here the WCNs use a fixed prior for the $u = \tilde{u}_*$ case, estimated on the training sets ($u = \tilde{u}_*$ class is always assigned 0.284; other classes are assigned $1 - 0.284 = 0.716$ times the WCN posterior). This table compares our models to WCNs on the task for which our model is designed. Here, the MLR and SVM models yielded

[1] `http://speak4it.com`

[2] 2-tailed Wilcoxon Signed-Rank Test; 10-way partitioning.

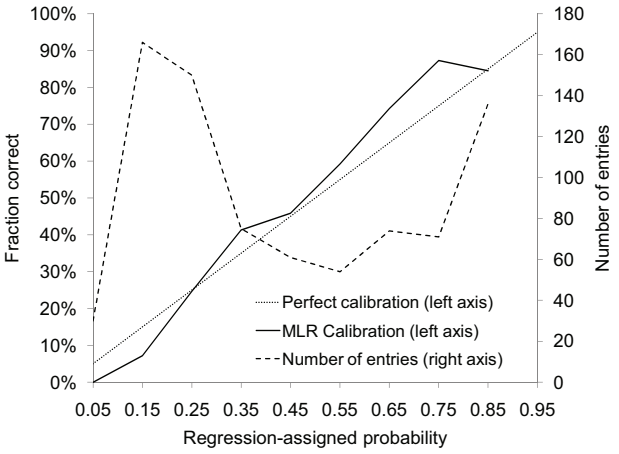


Figure 2: Calibration and histogram of probabilities assigned by MLR on corpus 1 (test set).

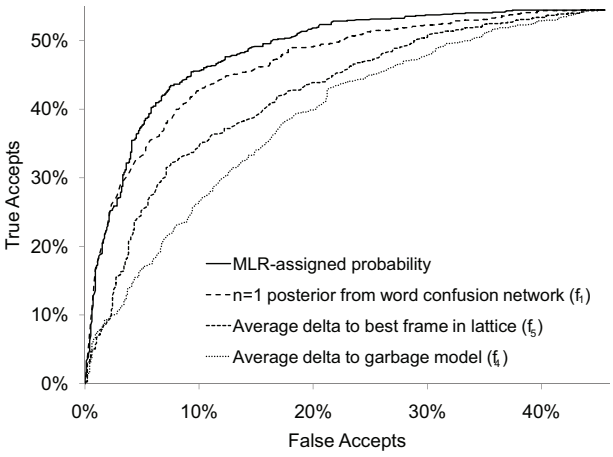


Figure 3: ROC curve for MLR and the 3 most informative input features on corpus 1 (test set).

significantly better results than the WCN baseline.

We next investigated the calibration properties of the models. Results for the MLR model on the $u = \tilde{u}_1$ class from corpus 1 test set are shown in Figure 2. This illustrates that the MLR model is relatively well-calibrated and yields broadly distributed probabilities. Results for the SVM were similar, and are omitted for space.

Finally we investigated whether the models yielded better accept/reject decisions than their individual features. Figure 3 shows the MLR model a receiver operating characteristic (ROC) curve for corpus 1 test set for the $u = \tilde{u}_1$ class. This confirms that the MLR model produces more accurate accept/reject decisions than the individual features alone. Results for the SVM were similar.

5 Conclusions

This paper has presented a method for assigning useful, meaningful probabilities to elements on an ASR N-Best list. Multinomial logistic regression (MLR) and support vector machines (SVMs) have been tested, and both produce significantly better models than a word confusion network baseline, as measured by average log likelihood. Further, the models appear to be well-calibrated and yield a better indication of correctness than any of its input features individually.

In dialog systems, we are often more interested in the concepts than specific words, so in future work, we hope to assign probabilities to concepts. In the meantime, we are applying the method to our dialog systems, to verify their usefulness in practice.

References

CC Chang and CJ Lin, 2001. *LIBSVM: a library for support vector machines*. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

A Genkin, DD Lewis, and D Madigan, 2005. *BMR: Bayesian Multinomial Regression Software*. `http://www.stat.rutgers.edu/~madigan/BMR/`.

V Goffin, C Allauzen, E Bocchieri, D Hakkani-Tur, A Ljolje, S Parthasarathy, M Rahim, G Riccardi, and M Saraclar. 2005. The AT&T Watson speech recognizer. In *Proc ICASSP, Philadelphia*.

JC Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.

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

JD Williams. 2008. Exploiting the ASR N-best by tracking multiple dialog state hypotheses. In *Proc ICSLP, Brisbane*.

SJ Young, M Gašić, S Keizer, F Mairesse, J Schatzmann, B Thomson, and K Yu. 2009. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*. To appear.

Not a simple yes or no: Uncertainty in indirect answers

Marie-Catherine de Marneffe, Scott Grimm and **Christopher Potts**
Linguistics Department
Stanford University
Stanford, CA 94305
{mcdm,sgrimm,cgpotts}@stanford.edu

Abstract

There is a long history of using logic to model the interpretation of indirect speech acts. Classical logical inference, however, is unable to deal with the combinations of disparate, conflicting, uncertain evidence that shape such speech acts in discourse. We propose to address this by combining logical inference with probabilistic methods. We focus on responses to polar questions with the following property: they are neither *yes* nor *no*, but they convey information that can be used to infer such an answer with some degree of confidence, though often not with enough confidence to count as resolving. We present a novel corpus study and associated typology that aims to situate these responses in the broader class of indirect question–answer pairs (IQAPs). We then model the different types of IQAPs using Markov logic networks, which combine first-order logic with probabilities, emphasizing the ways in which this approach allows us to model inferential uncertainty about both the context of utterance and intended meanings.

1 Introduction

Clark (1979), Perrault and Allen (1980), and Allen and Perrault (1980) study indirect speech acts, identifying a wide range of factors that govern how speakers convey their intended messages and how hearers seek to uncover those messages. Prior discourse conditions, the relationship between the literal meaning and the common ground, and specific lexical, constructional, and intonational cues all play a role. Green and Carberry (1992, 1994) provide an extensive computational model that interprets and generates indirect answers to polar questions. Their model focuses on inferring categorical answers, making use of discourse plans and coherence relations.

This paper extends such work by recasting the problem in terms of probabilistic modeling. We focus on the interpretation of indirect answers where the respondent does not answer with *yes* or *no*, but rather gives information that can be used by the hearer to infer such an answer only with some degree of certainty, as in (1).

(1) A: Is Sue at work?
B: She is sick with the flu.

In this case, whether one can move from the response to a *yes* or *no* is uncertain. Based on typical assumptions about work and illness, A might take B's response as indicating that Sue is at home, but B's response could be taken differently depending on Sue's character — B could be reproaching Sue for her workaholic tendencies, which risk infecting the office, or B could be admiring Sue's steadfast character. What A actually concludes about B's indirect reply will be based on some combination of this disparate, partially conflicting, uncertain evidence. The plan and logical inference model of Green and Carberry falters in the face of such collections of uncertain evidence. However, natural dialogues are often interpreted in the midst of uncertain and conflicting signals. We therefore propose to enrich a logical inference model with probabilistic methods to deal with such cases.

This study addresses the phenomenon of indirect question–answer pairs (IQAP), such as in (1), from both empirical and engineering perspectives.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 136–143,
Queen Mary University of London, September 2009.

First, we undertake a corpus study of polar questions in dialogue to gather naturally occurring instances and to determine how pervasive indirect answers that indicate uncertainty are in a natural setting (section 2). From this empirical base, we provide a classification of IQAPs which makes a new distinction between fully- and partially-resolving answers (section 3). We then show how inference in Markov logic networks can successfully model the reasoning involved in both types of IQAPs (section 4).

2 Corpus study

Previous corpus studies looked at how pervasive indirect answers to yes/no questions are in dialogue. Stenström (1984) analyzed 25 face-to-face and telephone conversations and found that 13% of answers to polar questions do not contain an explicit *yes* or *no* term. In a task dialogue, Hockey et al. (1997) found 38% of the responses were IQAPs. (This higher percentage might reflect the genre difference in the corpora used: task dialogue vs. casual conversations.) These studies, however, were not concerned with how confidently one could infer a *yes* or *no* from the response given.

We therefore conducted a corpus study to analyze the types of indirect answers. We used the Switchboard Dialog Act Corpus (Jurafsky et al., 1997) which has been annotated for approximately 60 basic dialog acts, clustered into 42 tags. We are concerned only with direct yes/no questions, and not with indirect ones such as "May I remind you to take out the garbage?" (Clark, 1979; Perrault and Allen, 1980). From 200 5-minute conversations, we extracted yes/no questions (tagged "qy") and their answers, but discarded tag questions as well as disjunctive questions, such as in (2), since these do not necessarily call for a *yes* or *no* response. We also did not take into account questions that were lost in the dialogue, nor questions that did not really require an answer (3). This yielded a total of 623 yes/no questions.

(2) [sw_0018_4082]

A: Do you, by mistakes, do you mean just like honest mistakes

A: or do you think they are deliberate sorts of things?

B: Uh, I think both.

(3) [sw_0070_3435]

A: How do you feel about your game?

A: I guess that's a good question?

B: Uh, well, I mean I'm not a serious golfer at all.

To identify indirect answers, we looked at the answer tags. The distribution of answers is given in Table 1. We collapsed the tags into 6 categories. Category I contains direct yes/no answers as well as "agree" answers (e.g., *That's exactly it.*). Category II includes statement–opinion and statement–non-opinion: e.g., *I think it's great*, *Me I'm in the legal department*, respectively. Affirmative non-*yes* answers and negative non-*no* answers form category III. Other answers such as *I don't know* are in category IV. In category V, we put utterances that avoid answering the question: by holding (*I'm drawing a blank*), by returning the question — wh-question or rhetorical question (*Who would steal a newspaper?*) — or by using a backchannel in question form (*Is that right?*). Finally, category VI contains dispreferred answers (Schegloff et al., 1977; Pomerantz, 1984).

We hypothesized that the phenomenon we are studying would appear in categories II, III and VI. However, some of the "na/ng" answers are disguised yes/no answers, such as *Right*, *I think so*, or *Not really*, and as such do not interest us. In the case of "sv/sd" and "nd" answers, many answers include reformulation, question avoidance (see 4), or a change of framing (5). All these cases are not really at issue for the question we are addressing.

(4) [sw_0177_2759]

A: Have you ever been drug tested?

B: Um, that's a good question.

(5) [sw_0046_4316]

A: Is he the guy wants to, like, deregulate heroin, or something?

B: Well, what he wants to do is take all the money that, uh, he gets for drug enforcement and use it for, uh, drug education.

A: Uh-huh.

B: And basically, just, just attack the problem at the demand side.

	Definition	Tag	Total
I	yes/no answers	ny/nn/aa	341
II	statements	sv/sd	143
III	affirmative/negative non-yes/no answers	na/ng	91
IV	other answers	no	21
V	avoid answering	^h/qw/qh/bh	18
VI	dispreferred answers	nd	9
Total			623

Table 1: Distribution of answer tags to yes/no questions.

(6) [sw_0046_4316]

A: That was also civil?

B: The other case was just traffic, and you know, it was seat belt law.

We examined by hand all yes/no questions for IQAPs and found 88 examples (such as (6), and (7)–(11)), which constitutes thus 14% of the total answers to direct yes/no questions, a figure similar to those of Stenström (1984). The next section introduces our classification of answers.

3 Typology of indirect answers

We can adduce the general space of IQAPs from the data assembled in section 2 (see also Bolinger, 1978; Clark, 1979). One point of departure is that, in cooperative dialogues, a response to a question counts as an answer only when some relation holds between the content of the response and the semantic desiderata of the question. This is succinctly formulated in the relation *IQAP* proposed by Asher and Lascarides (2003), p. 403:

> IQAP(α,β) holds only if there is a true direct answer p to the question $[\![\alpha]\!]$, and the questioner can infer p from $[\![\beta]\!]$ in the utterance context.

The apparent emphasis on truth can be set aside for present purposes; Asher and Lascarides's notions of truth are heavily relativized to the current discourse conditions. This principle hints at two dimensions of IQAPs which must be considered, and upon which we can establish a classification: (i) the type of answer which the proffered response provides, and (ii) the basis on which the inferences are performed. The typology established here adheres to this, distinguishing between fully- and partially-resolving answers as well as between the types of knowledge used in the inference (logical, linguistic, common ground/world).

3.1 Fully-resolving responses

An indirect answer can fully resolve a question by conveying information that stands in an inclusion relation to the direct answer: if $q \subseteq p$ (or $\neg p$), then updating with the response q also resolves the question with p (or $\neg p$), assuming the questioner knows that the inclusion relation holds between q and p. The inclusion relation can be based on logical relations, as in (7), where the response is an "over-answer", i.e., a response where more information is given than is strictly necessary to resolve the question. Hearers supply more information than strictly asked for when they recognize that the speaker's intentions are more general than the question posed might suggest. In (7), the most plausible intention behind the query is to know more about B's family. The hearer can also identify the speaker's plan and any necessary information for its completion, which he then provides (Allen and Perrault, 1980).

(7) [sw_0001_4325]

A: Do you have kids?

B: I have three.

While logical relations between the content of the question and the response suffice to treat examples such as (7), other over-answers often require substantial amounts of linguistic and/or world-knowledge to allow the inference to go through, as in (8) and (9).

(8) [sw_0069_3144]

A: Was that good?

B: Hysterical. We laughed so hard.

(9) [sw_0057_3506]

A: Is it in Dallas?

B: Uh, it's in Lewisville.

In the case of (8), a system must recognize that *hysterical* is semantically stronger than *good*. Similarly, to recognize the implicit *no* of (9), a system must recognize that Lewisville is a distinct location from Dallas, rather than, say, contained in Dallas, and it must include more general constraints as well (e.g., an entity cannot be in two physical locations at once). Once the necessary knowledge is in place, however, the inferences are properly licensed.

3.2 Partially-resolving responses

A second class of IQAPs, where the content of the answer itself does not fully resolve the question, known as partially-resolved questions (Groenendijk and Stokhof, 1984; Zeevat, 1994; Roberts, 1996; van Rooy, 2003), is less straightforward. One instance is shown in (10), where the gradable adjective *little* is the source of difficulty.

(10) [sw_0160_3467]

A: Are they [your kids] little?

B: I have a seven-year-old and a ten-year-old.

A: Yeah, they're pretty young.

The response, while an answer, does not, in and of itself, resolve whether the children should be considered *little*. The predicate *little* is a gradable adjective, which inherently possesses a degree of vagueness: such adjectives contextually vary in truth conditions and admit borderline cases (Kennedy, 2007). In the case of *little*, while some children are clearly little, e.g., ages 2–3, and some clearly are not, e.g., ages 14–15, there is another class in between for which it is difficult to assess whether *little* can be truthfully ascribed to them. Due to the slippery nature of these predicates, there is no hard-and-fast way to resolve such questions in all cases. In (10), it is the questioner who resolves the question by accepting the information proffered in the response as sufficient to count as *little*.

The dialogue in (11) shows a second example of an answer which is not fully-resolving, and intentionally so.

(11) [sw_0103_4074]

A: Did he raise him [the cat] or something[1]?

B: We bought the cat for him and so he's been the one that you know spent the most time with him.

Speaker B quibbles with whether the relation his son has to the cat is one of *raising*, instead citing two attributes that go along with, but do not determine, *raising*. *Raising an animal* is a composite relation, which typically includes the relations *owning* and *spending time with*. However, satisfying these two sub-relations does not strictly entail satisfying the *raising* relation as well. It is not obvious whether a system would be mistaken in attributing a fully positive response to the question, although it is certainly a *partially* positive response. Similarly, it seems that attributing a negative response would be misguided, though the answer is partly negative. The rest of the dialogue does not determine whether A considers this equivalent to *raising*, and the dialogue proceeds happily without this resolution.

The preceding examples have primarily hinged upon conventionalized linguistic knowledge, viz. what it means to *raise X* or for *X* to be *little*. A further class of partially-resolving answers relies on knowledge present in the common ground. Our initial example (1) illustrates a situation where different resolutions of the question were possible depending on the respondent's intentions: *no* if sympathetic, *yes* if reproachful or admiring.

The relationship between the response and question is not secured by any objective world facts or conventionalized meaning, but rather is variable — contingent on specialized world knowledge concerning the dialogue participants and their beliefs. Resolving such IQAPs positively or negatively is achieved only at the cost of a degree of uncertainty: for resolution occurs against the backdrop of a set of defeasible assumptions.

3.3 IQAP classification

Table 2 is a cross-classification of the examples discussed by whether the responses are fully- or partially-resolving answers and by the types of knowledge used in the inference (logical, linguistic, world). It gives, for each category, the counts of examples we found in the corpus. The partially-resolved class contains more than a third of the answers.

[1]The disjunct *or something* may indicate that A is open to hearing about alternatives to *raise*. We abstract away from this issue for present purposes and treat the more general case by assuming A's contribution is simply equivalent to "Did he raise him?"

	Logic	Linguistic	World	Total
Fully-Resolved	27 (Ex. 7)	18 (Ex. 8)	11 (Ex. 9)	56
Partially-Resolved	–	20 (Ex. 10;11)	12 (Ex. 1)	32

Table 2: Classification of IQAPs by knowledge type and resolvedness: counts and examples.

The examples given in (7)–(9) are fully resolvable via inferences grounded in logical relations, linguistic convention or objective facts: the answer provides enough information to fully resolve the question, and the modeling challenge is securing and making available the correct information. The partially-resolved pairs are, however, qualitatively different. They involve a degree of uncertainty that classical inference models do not accommodate in a natural way.

4 Towards modeling IQAP resolution

To model the reasoning involved in all types of IQAPs, we can use a relational representation, but we need to be able to deal with uncertainty, as highlighted in section 3. Markov logic networks (MLNs; Richardson and Domingos, 2006) exactly suit these needs: they allow rich inferential reasoning on relations by combining the power of first-order logic and probabilities to cope with uncertainty. A logical knowledge-base is a set of hard constraints on the set of possible worlds (set of constants and grounded predicates). In Markov logic, the constraints are "soft": when a world violates a relation, it becomes less probable, but not impossible. A Markov logic network encodes a set of weighted first-order logic constraints, such that a higher weight implies a stronger constraint. Given constants in the world, the MLN creates a network of grounded predicates which applies the constraints to these constants. The network contains one feature f_j for each possible grounding of each constraint, with a value of 1 if the grounded constraint is true, and 0 otherwise. The probability of a world x is thus defined in terms of the constraints j satisfied by that world and the weights w associated with each constraint (Z being the partition function):

$$P(X = x) = \frac{1}{Z} \sum_j w_j f_j(x)$$

In practice, we use the Alchemy implementation of Markov logic networks (Kok et al., 2009). Weights on the relations can be hand-set or learned. Currently, we use weights set by hand, which suffices to demonstrate that an MLN handles the pragmatic reasoning we want to model, but ultimately we would like to learn the weights.

In this section, we show by means of a few examples how MLNs give a simple and elegant way of modeling the reasoning involved in both partially- and fully-resolved IQAPs.

4.1 Fully-resolved IQAPs

While the use of MLNs is motivated by partially-resolved IQAPs, to develop the intuitions behind MLNs, we show how they model fully-resolved cases, such as in (9). We define two distinct places, Dallas and Lewisville, a relation linking a person to a place, and the fact that person K is in Lewisville. We also add the general constraint that an individual can be in only one place at a time, to which we assign a very high weight. Markov logic allows for infinite weights, which Alchemy denotes by a closing period. We also assume that there is another person L, whose location is unknown.

Constants and facts:
Place = {Dallas, Lewisville}
Person = {K,L}
BeIn(Person,Place)
BeIn(K,Lewisville)

Constraints:
// "If you are in one place, you are not in another."
(BeIn(x,y) ∧ (y != z)) ⇒ !BeIn(x,z).

Figure 1 represents the grounded Markov network obtained by applying the constraint to the constants K, L, Dallas and Lewisville. The graph contains a node for each predicate grounding, and an arc between each pair of nodes that appear together in some grounding of the constraint. Given that input, the MLN samples over possible worlds, and infers probabilities for the predicate BeIn, based on the constraints satisfied by each world and their weights. The MLN returns a very low probability for K being in Dallas, meaning that the answer to the question *Is it in Dallas?* is *no*:

BeIn(K,Dallas): 4.9995e-05

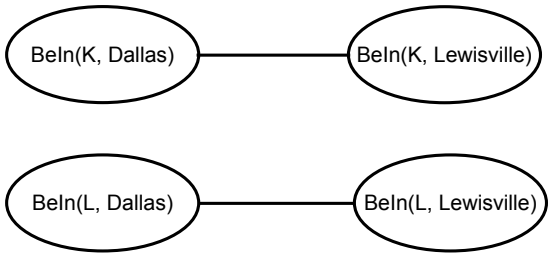


Figure 1: Grounded Markov network obtained by applying the constraints to the constants K, L, Dallas and Lewisville.

Since no information about L's location has been given, the probabilities of L being in Dallas or Lewisville will be equal and low (0.3), which is exactly what one would hope for. The probabilities returned for each location will depend on the number of locations specified in the input.

4.2 Partially-resolved IQAPs

To model partially-resolved IQAPs appropriately, we need probabilities, since such IQAPs feature reasoning patterns that involve uncertainty. We now show how we can handle three examples of partially-resolved IQAPs.

Gradable adjectives. Example (10) is a borderline case of gradable adjectives: the question bears on the predicate *be little* for two children of ages 7 and 10. We first define the constants and facts about the world, which take into account the relations under consideration, "BeLittle(X)" and "Age(X, i)", and specify which individuals we are talking about, K and L, as well as their ages.

Constants and facts:
age = $\{0 \ldots 120\}$
Person = $\{$K, L$\}$
Age(Person,age)
BeLittle(Person)
Age(K,7)
Age(L,10)

The relation between age and being little involves some uncertainty, which we can model using a logistic curve. We assume that a 12-year-old child lies in the vague region for determining "littleness" and therefore 12 will be used as the center of the logistic curve.

Constraints:
// "If you are under 12, you are little."
1.0 (Age(x,y) $\wedge$ y $<$ 12) $\Rightarrow$ BeLittle(x)
// "If you are above 12, you are not little."
1.0 (Age(x,y) $\wedge$ y $>$ 12) $\Rightarrow$!BeLittle(x)
// The constraint below links two instances of BeLittle.
(Age(x,u) $\wedge$ Age(y,v) $\wedge$ v$>$u $\wedge$ BeLittle(y)) $\Rightarrow$ BeLittle(x).

Asking the network about K being little and L being little, we obtain the following results, which lead us to conclude that K and L are indeed little with a reasonably high degree of confidence, and that the indirect answer to the question is heavily biased towards *yes*.

BeLittle(K): 0.92
BeLittle(L): 0.68

If we now change the facts, and say that K and L are respectively 12 and 16 years old (instead of 7 and 10), we see an appropriate change in the probabilities:

BeLittle(K): 0.58
BeLittle(L): 0.16

L, the 16-year-old, is certainly not to be considered "little" anymore, whereas the situation is less clear-cut for K, the 12-year-old (who lies in the vague region of "littleness" that we assumed).

Ideally, we would have information about the speaker's beliefs, which we could use to update the constraints' weights. Absent such information, we could use general knowledge from the Web to learn appropriate weights. In this specific case, we could find age ranges appearing with "little kids" in data, and fit the logistic curve to these.

This probabilistic model adapts well to cases where categorical beliefs fit uneasily: for borderline cases of vague predicates (whose interpretation varies by participant), there is no deterministic *yes* or *no* answer.

Composite relations. In example (11), we want to know whether the speaker's son raised the cat inasmuch as he owned and spent time with him. We noted that *raise* is a composite relation, which entails simpler relations, in this case *spend time with* and *own*, although satisfying any one of the simpler relations does not suffice to guarantee the truth of *raise* itself. We model the constants, facts, and constraints as follows:

Constants and Facts:
Person = {K}
Animal = {Cat}
Raise(Person,Animal)
SpendTime(Person,Animal)
Own(Person,Animal)
SpendTime(K,Cat)
Own(K,Cat)

Constraints:
// "If you spend time with an animal, you help raise it."
1.0 SpendTime(x,y) $\Rightarrow$ Raise(x,y)
// "If you own an animal, you help raise it."
1.0 Own(x,y) $\Rightarrow$ Raise(x,y)

The weights on the relations reflect how central we judge them to be in defining *raise*. For simplicity, here we let the weights be identical. Clearly, the greater number of relevant relations a pair of entities fulfills, the greater the probability that the composite relation holds of them. Considering two scenarios helps illustrate this. First, suppose, as in the example, that both relations hold. We will then have a good indication that by owning and spending time with the cat, the son helped raise him:

Raise(K,Cat): 0.88

Second, suppose that the example is different in that only one of the relations holds, for instance, that the son only spent time with the cat, but did not own it, and accordingly the facts in the network do not contain Own(K,Cat). The probability that the son raised the cat decreases:

Raise(K,Cat): 0.78

Again this can easily be adapted depending on the centrality of the simpler relations to the composite relation, as well as on the world-knowledge concerning the (un)certainty of the constraints.

Speaker beliefs and common ground knowledge. The constructed question–answer pair given in (1), concerning whether Sue is at work, demonstrated that how an indirect answer is modeled depends on different and uncertain evidence. The following constraints are intended to capture some background assumptions about how we regard working, being sick, and the connections between those properties:

// "If you are sick, you are not coming to work."
Sick(x) $\Rightarrow$!AtWork(x)
// "If you are hardworking, you are at work."
HardWorking(x) $\Rightarrow$ AtWork(x)
// "If you are malicious and sick, you come to work."
(Malicious(x) $\wedge$ Sick(x)) $\Rightarrow$ AtWork(x)
// "If you are at work and sick, you are malicious or thoughtless."
(AtWork(x) $\wedge$ Sick(x)) $\Rightarrow$ (Malicious(x) $\vee$ Thoughtless(x))

These constraints provide different answers about Sue being at work depending on how they are weighted, even while the facts remain the same in each instance. If the first constraint is heavily weighted, we get a high probability for Sue not being at work, whereas if we evenly weight all the constraints, Sue's quality of being a hard-worker dramatically raises the probability that she is at work. Thus, MLNs permit modeling inferences that hinge upon highly variable common ground and speaker beliefs.

Besides offering an accurate treatment of fully-resolved inferences, MLNs have the ability to deal with degrees of certitude. This power is required if one wants an adequate model of the reasoning involved in partially-resolved inferences. Indeed, for the successful modeling of such inferences, it is essential to have a mechanism for adding facts about the world that are accepted to various degrees, rather than categorically, as well as for updating these facts with speakers' beliefs if such information is available.

5 Conclusions

We have provided an empirical analysis and initial treatment of indirect answers to polar questions. The empirical analysis led to a categorization of IQAPs according to whether their answers are fully- or partially-resolving and according to the types of knowledge used in resolving

the question by inference (logical, linguistic, common ground/world). The partially-resolving indirect answers injected a degree of uncertainty into the resolution of the predicate at issue in the question. Such examples highlight the limits of traditional logical inference and call for probabilistic methods. We therefore modeled these exchanges with Markov logic networks, which combine the power of first-order logic and probabilities. As a result, we were able to provide a robust model of question–answer resolution in dialogue, one which can assimilate information which is not categorical, but rather known only to a degree of certitude.

Acknowledgements

We thank Christopher Davis, Dan Jurafsky, and Christopher D. Manning for their insightful comments on earlier drafts of this paper. We also thank Karen Shiells for her help with the data collection and Markov logic.

References

James F. Allen and C. Raymond Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press, Cambridge.

Dwight Bolinger. 1978. Yes–no questions are not alternative questions. In Henry Hiz, editor, *Questions*, pages 87–105. D. Reidel Publishing Company, Dordrecht, Holland.

Herbert H. Clark. 1979. Responding to indirect speech acts. *Cognitive Psychology*, 11:430–477.

Nancy Green and Sandra Carberry. 1992. Conversational implicatures in indirect replies. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 64–71, Newark, Delaware, USA, June. Association for Computational Linguistics.

Nancy Green and Sandra Carberry. 1994. A hybrid reasoning model for indirect answers. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 58–65, Las Cruces, New Mexico, USA, June. Association for Computational Linguistics.

Jeroen Groenendijk and Martin Stokhof. 1984. *Studies in the Semantics of Questions and the Pragmatics of Answers*. Ph.D. thesis, University of Amsterdam.

Beth Ann Hockey, Deborah Rossen-Knill, Beverly Spejewski, Matthew Stone, and Stephen Isard. 1997. Can you predict answers to Y/N questions? Yes, No and Stuff. In *Proceedings of Eurospeech 1997*.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder Institute of Cognitive Science.

Christopher Kennedy. 2007. Vagueness and grammar: The semantics of relative and absolute gradable adjectives. *Linguistics and Philosophy*, 30(1):1–45.

Stanley Kok, Marc Sumner, Matthew Richardson, Parag Singla, Hoifung Poon, Daniel Lowd, Jue Wang, and Pedro Domingos. 2009. The Alchemy system for statistical relational AI. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, WA.

C. Raymond Perrault and James F. Allen. 1980. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3-4):167–182.

Anita M. Pomerantz. 1984. Agreeing and disagreeing with assessment: Some features of preferred/dispreferred turn shapes. In J. M. Atkinson and J. Heritage, editors, *Structure of Social Action: Studies in Conversation Analysis*. Cambridge University Press.

Matt Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Craige Roberts. 1996. Information structure: Towards an integrated formal theory of pragmatics. In Jae Hak Yoon and Andreas Kathol, editors, *OSU Working Papers in Linguistics*, volume 49: Papers in Semantics, pages 91–136. The Ohio State University Department of Linguistics, Columbus, OH. Revised 1998.

Robert van Rooy. 2003. Questioning to resolve decision problems. *Linguistics and Philosophy*, 26(6):727–763.

Emanuel A. Schegloff, Gail Jefferson, and Harvey Sacks. 1977. The preference for self-correction in the organization of repair in conversation. *Language*, 53:361–382.

Anna-Brita Stenström. 1984. Questions and responses in English conversation. In Claes Schaar and Jan Svartvik, editors, *Lund Studies in English 68*, Malmö Sweden. CWK Gleerup.

Henk Zeevat. 1994. Questions and exhaustivity in update semantics. In Harry Bunt, Reinhard Muskens, and Gerrit Rentier, editors, *Proceedings of the International Workshop on Computational Semantics*, pages 211–221. ITK, Tilburg.

Concept Form Adaptation in Human-Computer Dialog

Svetlana Stoyanchev and **Amanda Stent**
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794-4400, USA
svetastenchikova@gmail.com, amanda.stent@gmail.com

Abstract

In this work we examine user adaptation to a dialog system's choice of realization of task-related concepts. We analyze forms of the time concept in the *Let's Go!* spoken dialog system. We find that users adapt to the system's choice of time form. We also find that user adaptation is affected by perceived system adaptation. This means that dialog systems can guide users' word choice and can adapt their own recognition models to gain improved ASR accuracy.

1 Introduction

Considerable research has now demonstrated that human dialog partners exhibit lexical and syntactic *convergence*; that is, that in a human-human conversation the participants become more similar in their use of language over time (Brennan and Clark, 1996; Lockridge and Brennan, 2002; Pickering and others, 2000; Reitter et al., 2006). Several Wizard-of-Oz studies have also shown evidence of convergence in human-computer dialog (Branigan and others, 2003; Brennan, 1996; Gustafson and others, 1997).

In recent work, we examined user adaptation[1] to the system's choice of verb and preposition using the deployed *Let's Go!* spoken dialog system (Stoyanchev and Stent, 2009a). This was the first study to look at convergence with real users of a real dialog system and examined user adaptation to verbs and prepositions. The study described in this paper is a follow-on to our previous study. Here we look at user adaptation to the system's choice of realization of task-related concepts. In this paper, we: (1) Confirm our previous results showing that users adapt to the system's choice of words, using transcribed data rather than ASR output; (2) Show that adaptation can persist over time; and (3) Show that user adaptation is strengthened by perceived system adaptation.

In addition to providing further evidence of convergence in human-computer dialog, our results have implications for dialog system design. Currently, much dialog systems research is devoted to improving ASR accuracy, because this is a significant contributor to task success rates and to dialog length. One way to improve ASR accuracy is to use targeted language models. Since users adapt to the system's choices of realization for task-related concepts, we can predict the user's choice of realization and use this to adjust the ASR's language model, improving ASR accuracy specifically on concept words. Another way to improve ASR accuracy is to guide the user into using words that are likely to be recognized correctly (Hockey and others, 2003; Sheeder and Balogh, 2003; Tomko and Rosenfeld, 2006). Our results imply that if the designer of a dialog system wants to improve ASR accuracy, system prompts should be designed to use word choices that are more recognizable; and, when, possible, to be adaptive to the user's choice of form for task-related concepts.

2 System

We conducted our experiment using the *Let's Go!* deployed telephone-based spoken dialog system which provides information about bus routes in Pittsburgh (Raux and others, 2005). *Let's Go!* answers the phones at the transit authority's website outside of normal business hours. Its users are

[1] In this paper, we use the term *adaptation* to indicate directional convergence, e.g. user adaptation to a system. We make no claims about the psycholinguistic models underlying this adaptation.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 144–147,
Queen Mary University of London, September 2009.

SYS:	Welcome to the CMU Let's Go! Bus Information System. How may I help you?
USR:	I'd like to go to Forbes and Murray.
SYS:	To Forbes and Murray. Where are you leaving from?
USR:	Carnegie Mellon.
SYS:	From Carnegie Mellon. What time would you like to leave?
USR:	Ten thirty p.m.
SYS:	Leaving at ten thirty p. m.?
USR:	Yes
SYS:	There are five buses running between Carnegie Mellon and Forbes and Murray...

Figure 1: Sample dialog with *Let's Go!*

Time form	Example realizations	Frequency
TIME	four, five, six thirty...	31.1%
APM	four a.m., ten p.m., one fifteen p. m.	43.5%
POD	four in the morning/evening/afternoon/	4.6%
OCLOCK	five o'clock	16%
OTHER	four o'clock p. m., sixteen hundred hours	4.8%

Table 1: Time forms in users' utterances and their relative frequencies in one month of *Let's Go!* 2006 data.

naive callers from the general population with a real task to perform. In order to provide bus route information, *Let's Go!* elicits values for several task-related concepts: an optional bus route number, a departure place, a destination and a desired travel time. Each concept is explicitly confirmed. Figure 1 shows a sample dialog with the system.

In this work we investigate adaptation to the time concept because it has multiple different realizations, as shown in Table 1. This variability is not unique to time; however, it is the only task-related concept in *Let's Go!* that is not usually realized using named entities (which exhibit less variability).

3 Method

In order to study adaptation, we need to identify a *prime*, a point in the conversation where one partner introduces a realization. In *Let's Go!* the system always asks the user to specify a departure time. The user then typically says a time, which the system confirms (see Figure 1). We simulate an ASR error on the user's response to the system's time request, so that when the system confirms the departure time it confirms a time other than that recognized in the user's response. To make the system's error more realistic, the time in the simulated error is a time that is phonetically close to the time (hour and minute) recognized in the user's response. The system's confirmation prompt is our *prime*.

The system runs in one of the three conditions: SYS_TIME, SYS_APM, or SYS_POD. In each condition it uses the corresponding time format (TIME, APM, or POD as shown in Table 1). TIME is the most frequent form in the 2006 *Let's Go!* corpus, but it is potentially ambiguous as it can mean either night or day. APM is the shortest unambiguous form. POD is longer and has a very low frequency in the 2006 *Let's Go!* corpus.[2]

We collected approximately 2000 dialogs with *Let's Go!* using this setup. We used the ASR output to identify dialogs where a time appears in the ASR output at least twice[3]. We manually transcribed 50 dialogs for each experimental condition. Some of these turned out not to contain mentions of time either before or after the system's time confirmation prompt, so we excluded them.

We examine whether the user adapts to the system's choice of form for realizing the time concept, both in the first time-containing post-confirmation utterance, and in the rest of the dialog (until the user hangs up or says "New query").

4 Results

In this section we first examine user adaptation to system's choice of time expression, and then look at how perceived system adaptation affects user adaptation.

4.1 User adaptation to system time form

If the user adapts to the system's time form, then we would expect to see a greater proportion of the system's time form in user utterances following the prime. We compare the proportion of three time forms (APM, TIME, and POD) in each system condition for 1) *Unprimed*, 2) *First After*, and 3) *All After* user's utterances, as shown in Table 2. *Unprimed* utterances are the user's time specification immediately prior to the prime (the system's confirmation prompt). *First After* utterances are user utterances immediately following the prime. *All After* utterances are all user utterances from the prime until the user either hangs up or says "New

[2]We would have liked to also include OCLOCK in the experiment. However, due to resource limitations we had to choose only three conditions.

[3]The most frequent user response to the system's request to specify a departure time is "Now"; we exclude these from our experiment.

Unprimed			
system/user	Usr:APM	Usr:TIME	Usr:POD
SYS_APM	25%	42%	**8%**
SYS_TIME	**30%**	**52%**	2%
SYS_POD	24%	49%	4%
First After			
system/user	Usr:APM	Usr:TIME	Usr:POD
SYS_APM	**49%**	29% ♠	2%
SYS_TIME	21% ♣	**58%**	0%
SYS_POD	29%	45%	5%
All After			
system/user	Usr:APM	Usr:TIME	Usr:POD
SYS_APM	**63%**	19% ♣	3%
SYS_TIME	21% ♣	**50%**	2%
SYS_POD	37% ♣	38%	**4%**

Table 2: Proportions of time forms in different system prompt conditions. The highest proportion among system conditions for each time form is highlighted. Occurrences of time forms other than the three examined time forms are excluded from this table. ♠ indicates a statistically significant difference from the highlighted value in the column ($p < .05$ with Bonferroni adjustment). ♣ indicates a statistically significant difference from the highlighted value in the column ($p < .01$ with Bonferroni adjustment).

query". To test the statistical significance of our results we perform inference on proportions for a large sample.

APM There are no statistically significant differences in the proportions of Usr:APM[4] forms in *Unprimed* utterances for the different system conditions. The proportion of Usr:APM forms in *First After* utterances is significantly higher in the SYS_APM condition than in the SYS_TIME condition ($p < .01$), although not significantly different than in the SYS_POD condition. The proportion of Usr:APM forms in the *All After* utterances is significantly higher in the SYS_APM condition than in both the SYS_TIME and the SYS_POD conditions ($p < .01$). We conclude that there is user adaptation to system time form in the SYS_APM condition.

TIME There are no statistically significant differences in the proportions of Usr:TIME forms in *Unprimed* utterances for the different system conditions. The proportions of Usr:TIME forms in the *First After* utterances in the SYS_TIME condition is significantly higher than that in the SYS_APM condition ($p < .01$), but not significantly higher than that in the SYS_POD condition. The same is true of Usr:TIME forms in the *All After* utterances. We conclude that there is user adaptation to system time form in the SYS_TIME condition.

POD We did not find statistically significant differences in Usr:POD forms for the different system conditions in either the *Unprimed*, *First After* or *All After* data. Because this is the *long unambiguous* form, users may have felt that it would not be recognized or that it would be inefficient to produce it.

Figures 2 illustrates the effect of user adaptation on time form for the SYS_APM and SYS_TIME conditions.

[4]Usr:*time-form* refers to the occurrence of the *time-form* in a user's utterance.

condition	keep	adapt	switch	total
adaptive	81.8%	-	18.2%	33
non-adaptive	37.5%	29.1%	35.4%	48

Table 3: Proportions of user actions in *First After* confirmation utterances

4.2 The effect of system adaptation on user adaptation

Sometimes the user happens to use the same form in their initial specification of time that the system uses in its confirmation prompt. This gives the illusion that the system is adapting its choice of time form to the user. We examined whether users' perception of system adaptation affected user adaptation in *First After* confirmation utterances.

For this analysis we used only the dialogs in the SYS_APM and SYS_TIME conditions since the POD form is rare in the *Unprimed* utterances. We distinguish between three possible user actions following the system's confirmation prompt: 1) *keep* - use the same form as in the unprimed utterance; 2) *adapt* – switch to the same form as in the system's confirmation prompt; and 3) *switch* - switch to a different form than the one used in the system's confirmation prompt or in the unprimed utterance.

Table 3 shows the proportions for each possible user action. In the *adaptive* condition users are twice as likely to keep the time form than in the *non-adaptive* condition (81.8% vs. 37.5%). This difference is statistically significant ($p < .001$). In the *non-adaptive* system condition users who change time form are slightly more likely to switch (35.4%) than to adapt (29.1%).

These results suggest that when the system does not adapt to the user, the user's choice is unpredictable. However, if the system adapts to the user, the user is likely to keep the same form. This

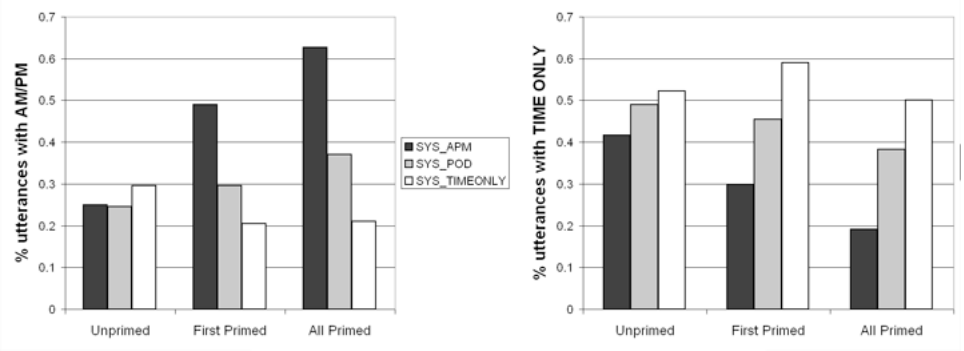


Figure 2: User Utterances with TIME_APM and TIME_ONLY.

means that if the system can adapt to the user when the user chooses a form that is more likely to be recognized correctly, that provides positive reinforcement, making the user more likely to use that felicitous form in the future. Furthermore, if the system does adapt to the user then it may be possible with high accuracy to predict the user's form for subsequent utterances, and to use this information to improve ASR accuracy for subsequent utterances (Stoyanchev and Stent, 2009b).

5 Conclusions and Future Work

In this paper, we analyzed user adaptation to a dialog system's choice of task-related concept forms. We showed that users do adapt to the system's word choices, and that users are more likely to adapt when the system appears to adapt to them. This information may help us guide users into more felicitous word choices, and/or modify the system to better recognize anticipated user word choices. In future work we plan to analyze the effect of ASR adaptation to user word choice on speech recognition performance in spoken dialog.

References

H. Branigan et al. 2003. Syntactic alignment between computers and people: The role of belief about mental states. In *Proceedings of the 25th Annual Conference of the Cognitive Science Society*.

S. Brennan and H. Clark. 1996. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology*, 22(6):1482–1493.

S. Brennan. 1996. Lexical entrainment in spontaneous dialog. In *Proceedings of ISSD*, pages 41–44.

J. Gustafson et al. 1997. How do system questions influence lexical choices in user answers? In *Proceedings of Eurospeech*.

B. Hockey et al. 2003. Targeted help for spoken dialogue systems: intelligent feedback improves naive users performance. In *Proceedings of EACL*.

C. Lockridge and S. Brennan. 2002. Addressees' needs influence speakers' early syntactic choices. *Psychonomics Bulletin and Review*, 9:550–557.

M. Pickering et al. 2000. Activation of syntactic priming during language production. *Journal of Psycholinguistic Research*, 29(2):205–216.

A. Raux et al. 2005. Let's go public! taking a spoken dialog system to the real world. In *Proceedings of Eurospeech*.

E. Reitter, J. Moore, and F. Keller. 2006. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation. In *Proceedings of CogSci*.

T. Sheeder and J. Balogh. 2003. Say it like you mean it: Priming for structure in caller responses to a spoken dialog system. *International Journal of Speech and Technology*, 6:103–111.

S. Stoyanchev and A. Stent. 2009a. Lexical and syntactic priming and their impact in deployed spoken dialog systems. In *Proceedings of NAACL*.

S. Stoyanchev and A. Stent. 2009b. Predicting concept types in user corrections in dialog. In *Proceedings of the EACL Workshop on the Semantic Representation of Spoken Language*.

S. Tomko and R. Rosenfeld. 2006. Shaping user input in speech graffiti: a first pass. In *Proceedings of CHI*.

Automatic Generation of Information State Update Dialogue Systems that Dynamically Create Voice XML, as Demonstrated on the iPhone

Helen Hastie, Xingkun Liu and **Oliver Lemon**
School of Informatics
University of Edinburgh
{hhastie,xliu4,olemon}@inf.ed.ac.uk

Abstract

We demonstrate DUDE[1] (Dialogue and Understanding Development Environment), a prototype development environment that automatically generates dialogue systems from business-user resources and databases. These generated spoken dialogue systems (SDS) are then deployed on an industry standard Voice XML platform. Specifically, the deployed system works by dynamically generating context-sensitive Voice XML pages. The dialogue move of each page is determined in real time by the dialogue manager, which is an Information State Update engine. Firstly, we will demonstrate the development environment which includes automatic generation of speech recognition grammars for robust interpretation of spontaneous speech, and uses the application database to generate lexical entries and grammar rules. A simple graphical interface allows users (i.e. developers) to easily and quickly create and the modify SDS without the need for expensive service providers. Secondly, we will demonstrate the deployed system which enables participants to call up and speak to the SDS recently created. We will also show a pre-built application running on the iPhone and Google Android phone for searching for places such as restaurants, hotels and museums.

[1]Patent Pending

1 Introduction

With the advent of new mobile platforms such as the iPhone and Google Android, there is a need for a new way to interact with applications and search for information on the web. Google Voice Search is one such example. However, we believe that this simple "one-shot" search using speech recognition is not optimal for the user. A service that allows the user to have a dialogue via their phone opens up a wider set of possibilities. For example, the user may be visiting a foreign city and would like to have a discussion about the types of restaurants, their cuisine, their price-range and even ask for recommendations from the system or their friends on social networking sites. The Dialogue Understanding Development Environment or DUDE makes this possible by providing a flexible, natural, mixed initiative dialogue using an information state update dialogue engine (Bos et al., 2003).

Currently, if a company wishes to deploy such a spoken dialogue system, they have to employ a costly service provider with a long turn around time for any changes to the system, even minor ones such as a special promotion offer. In addition, there is steep competition on application sites such as Google Market Place and Apple App Store which are populated with very cheap applications. DUDE's Development Environment takes existing business-user resources and databases and automatically generates the dialogue system. This reduces development time and, therefore, costs and opens up the technology to a wider user-base. In addition, the DUDE environment is so easy to use that it gives the control back into the business-user and away from independent services providers.

In this paper, we describe the architecture and

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 148–151,
Queen Mary University of London, September 2009.

technology of the DUDE Development Environment and then discuss how the deployed system works on a mobile platform.

2 The DUDE Development Environment

Figure 1 shows the DUDE Development Environment architecture whereby the main algorithm takes the business-user resources and databases as input and uses these to automatically generate the spoken dialogue system which includes a Voice XML generator. Advantages of using business-user resources such as Business Process Models (BPM) (Williams, 1967) include the fact that graphical interfaces and authoring environments are widely available (e.g. Eclipse). In addition, business-user resources can contain a lot of additional information as well as call flow including context, multi-media and multiple customer interactions.

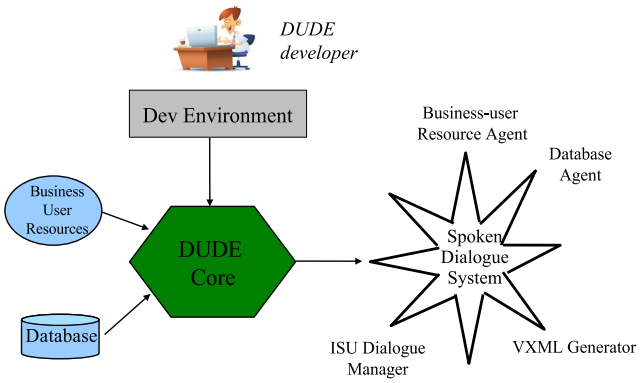


Figure 1: The DUDE Architecture

2.1 Spoken Dialogue System Generation

Many sophisticated research systems are developed for specific applications and cannot be easily transferred to another, even very similar task or domain. The problem of components being domain specific is especially prevalent in the core area of dialogue management. For example MIT's Pegasus and Mercury systems (Seneff, 2002) have dialogue managers (DM) that use approximately 350 domain-specific hand-coded rules each. The sheer amount of labour required to construct systems prevents them from being more widely and rapidly deployed. We present a solution whereby BPMs and related authoring tools are used to specify *domain-specific* dialogue interactions which are combined with *domain-general* dialogue managers. Specifically, the DM consults the BPM to determine what task-based steps to take next, such as asking for price range after establishing preferred cuisine type. General aspects of dialogue, such as confirmation and clarification strategies, are handled by the domain-general DM. Values for constraints on transitions and branching in the BPM, for example "present insurance offer if the user is business-class", are compiled into domain-specific parts of the Information State. XML format is used for BPMs, and they are compiled into finite state machines consulted by the spoken dialogue system. The domain-general dialogue manager was mostly abstracted from the TALK system (Lemon et al., 2006).

Using DUDE, developers do not have to write a single line of grammar code. There are three types of grammars: (1) a core grammar, (2) a grammar generated from the database and BPM, and (3) dynamically generated grammars created during the dialogue. The core grammar (1) was developed to cover basic information-seeking interactions. In addition (2), the system compiles relevant database entries and their properties into the appropriate "slot-filling" parts of a SRGS GRXML (Speech Recognition Grammar Specification) grammar for each specific BPM node. Task level grammars are used to allow a level of mixed initiative, for example, if the system asks "what type of cuisine?" the user can reply with cuisine and also any other slot type, such as, "cheap Italian". The dynamically generated grammars (3), such as for restaurants currently being recommended, minimizes grammar size and makes the system more efficient. In addition to the above-mentioned grammars, developers are able to provide task spotter phrases and synonyms reflecting how users might respond by using the DUDE Development Environment. If these are not already covered by the existing grammar, DUDE automatically generates rules to cover them.

The generated SRGS GRXML grammars are used to populate the Voice XML pages and consequently used by the Voice XML Platform Speech recogniser. In this case, we deploy our system to the Voxeo Platform (http://www.voxeo.com). As well as the W3C standard SRGS GRXML, DUDE is able to generate alternative grammar specifications such as SRGS ABNF (Augmented Backus-Naur Form), JSGF ABNF (Java Speech Grammar Format) and Nuance's GSL (Grammar Specifica-

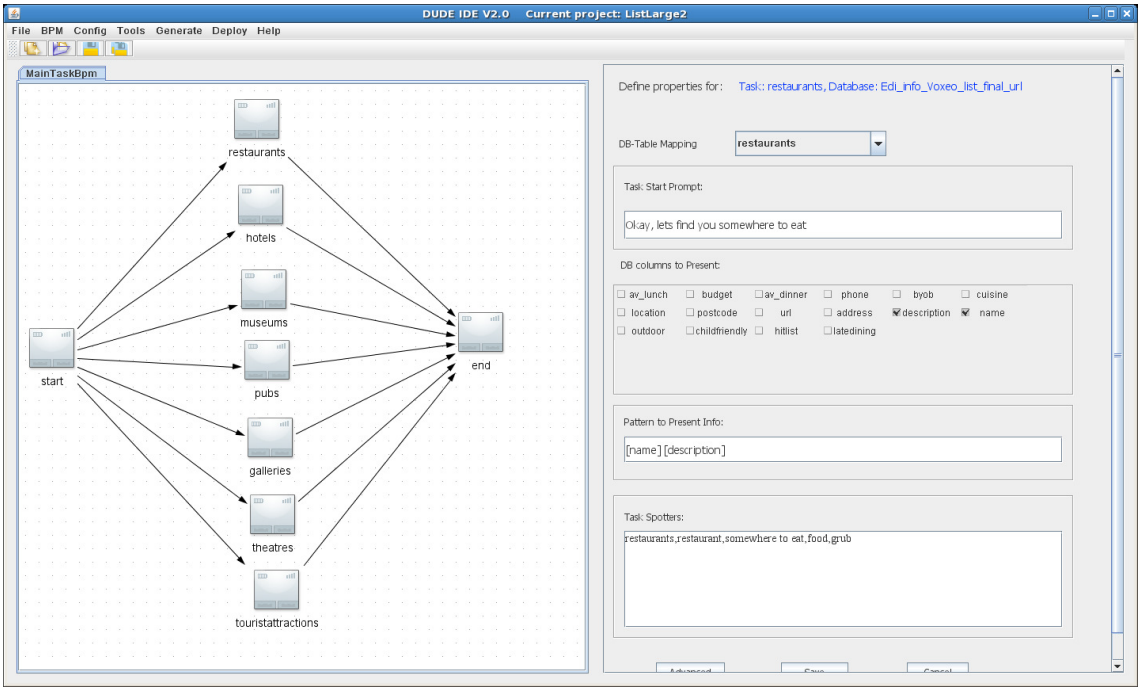


Figure 2: Example: using the DUDE Development Environment to define spotter phrases and other information for the different BPM tasks

tion Language).

2.2 The Development Environment

As mentioned above, the DUDE Development Environment can be used to define system prompts and add task spotter phrases and synonyms to the grammars. Figure 2 shows the GUI with the BPM on the left hand side and the properties pane for the restaurants task on the right hand side. In this pane the developer can define the system prompt, the information to be presented to the user and the spotter phrases. Here the developer is associating the phrases "restaurants, restaurant, somewhere to eat...." with the restaurant task. This means that if the user says "I want somewhere to eat", the restaurant part of the BPM will be triggered. Note that multi-word phrases may also be defined. The defined spotters are automatically compiled into the grammar for parsing and speech recognition. By default all the lexical entries for answer-types for the subtasks will already be present as spotter phrases. DUDE checks for possible ambiguities, for example if "pizza' is a spotter for both cuisine_type for a restaurant task and food_type for a shopping task, the system uses a clarification subdialogue to resolve them at runtime.

Figure 3 shows the developer specifying the required linguistic information to automate the cuisine subtask of the restaurants task. Here the developer specifies the system prompt "What type of cuisine do you want?" and a phrase for implicit confirmation of provided values, e.g. "a [X] restaurant", where [X] is a variable that will be replaced with the semantics of the speech recognition hypothesis for the user input. The developer also specifies here the answer type that will resolve the system prompt. There are predefined answer-types extracted from the databases, and the developer can select and/or edit these, adding phrases and synonyms. In addition, they have the ability to define their own answer-types.

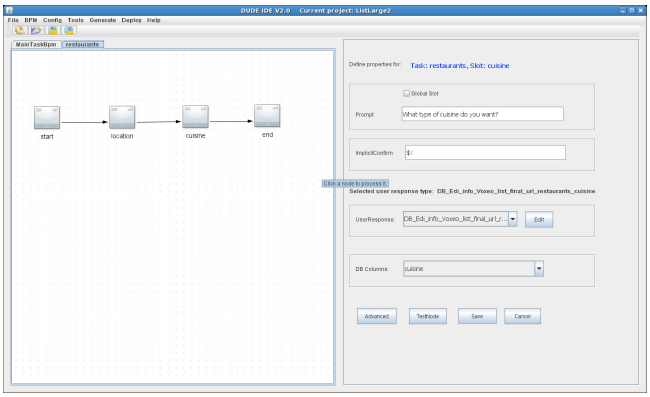


Figure 3: Example: using the DUDE Development Environment to define prompts, answer sets, and database mappings for the cuisine subtask

3 Deployment of the Generated Spoken Dialogue System

The second part of the demonstration shows a pre-built multimodal application running on the iPhone (http://www.apple.com) and Google Android phone (http://code.google.com//android). This application allows the user to have a dialogue about places of interest using The List website (http://www.list.co.uk). Figure 4 shows screenshots of the iPhone, firstly with The List homepage and then a page with content on Bar Roma, an "italian restaurant in Edinburgh" as requested by the user through spoken dialogue.

Figure 4: DUDE-generated iPhone List Application pushing relevant web content

Figure 5 shows the architecture of this system whereby the DUDE server runs the spoken dialogue system (as outputted from the DUDE Development Environment). This system dynamically generates Voice XML pages whose dialogue move and grammar is determined by the Information State Update Dialogue Model. These Voice XML pages are sent in real time to the Voice XML platform (in our case Voxeo) which the user talks to by placing a regular phone call. In addition, DUDE communicates the relevant URL via a server connection.

4 Summary

This paper describes a demonstration of the DUDE Development Environment and its resulting spoken dialogue systems as deployed on a mobile phone, specifically the iPhone and Google Android. With the emergence of web-enabled smart-phones, a new and innovative interactive method is needed that combines web-surfing and dialogue in order to get the user exactly the information required in real time.

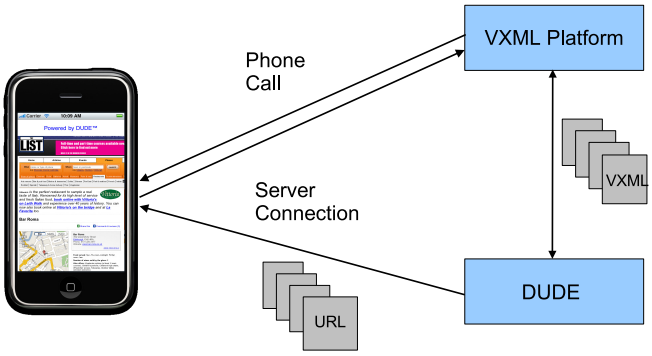


Figure 5: Architecture of deployed DUDE Application on a mobile phone (e.g. the iPhone)

5 Acknowledgement

This project is funded by a Scottish Enterprise Proof of Concept Grant (project number 8-ELM-004). We gratefully acknowledge The List for giving us data for our prototype application.

References

Johan Bos, Ewan Klein, Oliver Lemon, and Tetsushi Oka. 2003. DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture. In *4th SIGdial Workshop on Discourse and Dialogue*, pages 115–124, Sapporo.

Adam Cheyer and David Martin. 2001. The Open Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1/2):143–148.

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 *Proceedings of EACL*, pages 119–122.

Stephanie Seneff. 2002. Response Planning and Generation in the Mercury Flight Reservation System. *Computer Speech and Language*, 16.

S Williams. 1967. Business process modeling improves administrative control. *Automation*, pages 44–50.

Dialog System for Mixed Initiative One-Turn Address Entry and Error Recovery

Rajesh Balchandran, Leonid Rachevsky, Larry Sansone, Roberto Sicconi
IBM T J Watson Research Center, Yorktown Heights, NY 10598, USA
rajeshb,lrachevs,lsansone,rsicconi@us.ibm.com

Abstract

In this demonstration we present a mixed-initiative dialog system for address recognition that lets users to specify a complete addresses in a single sentence with address components spoken in their natural sequence. Users can also specify fewer address components in several ways, based on their convenience. The system extracts specified address components, prompts for missing information, disambiguates items independently or collectively all the while guiding the user so as to obtain the desired valid address. The language modeling and dialog management techniques developed for this purpose are also briefly described. Finally, several use cases with screen shots are presented. The combined system yields very high task completion accuracy on user tests.

1 Introduction

In recent years, speech recognition has been employed for address input by voice for GPS navigation and similar applications. Users are typically directed to speak address components one at a time - first a state name, then city, street and finally the house number - typically taking four or more turns. In this demonstration we present a mixed-initiative dialog system that makes address input by voice more natural, so users can speak the complete address (in normal order) (for e.g. "Fifteen State Street Boston Massachusetts"), in a single turn. They could also specify fewer address components as per their convenience, and the system would be expected to guide them to obtain a complete and valid address.

2 System Description

Figure 1 shows the high-level architecture and key components of the system. A programmable framework consisting of a *system bus* that connects various components (called *plugins*) forms the core of the speech-dialog system. Key components include plugins to connect to the ASR (Automatic Speech Recognition) and TTS (Text-To-Speech) engines, the GUI (Graphical User Interface), the *Natural Language Processor* and the *Dialog Manager*.

2.1 Speech Recognition and component Extraction

Speech recognition is carried out using a statistical Language Model (LM) with Embedded Grammars (Gillett and Ward, 1998) to represent *Named Entities* such as city names, numbers etc. This provides flexibility for the user, while allowing for dynamic content to be updated when required, simply by swapping associated embedded grammars. For e.g., the grammar of street names could be updated based on the selected city. The IBM Embedded Via Voice (EVV) (Sicconi et al., 2009) (Beran et al., 2004) ASR engine provides this functionality and is used in this system.

In this system, a two-pass speech recognition technique (Balchandran et al., 2009) is employed, based on multiple LMs where, the first pass is used to accurately recognize some components, and the values of these components are used to dynamically update another LM which is used for the second pass to recognize remaining components. Specifically, the first LM is used to recognize the city and state while the second is used to recognize the street name and number. The street names and optionally the house number embedded grammars

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 152–155,
Queen Mary University of London, September 2009.

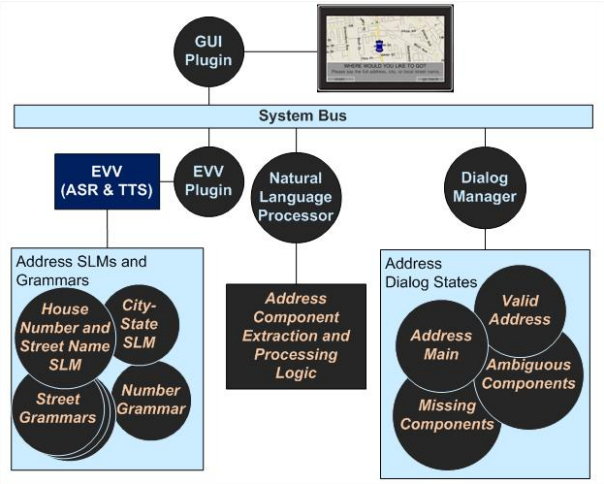


Figure 1: System Architecture

in the second LM are updated based on the city and state recognized using the first LM. This is carried out transparent to the user - so the user perceives full address recognition in one step.

2.2 Dialog management

A key part of this system is the dialog management component that handles incompletely specified input, various types of ambiguities and error conditions, all the while having an intelligent dialog with the user so as to correct these errors and obtain a valid address at the end. A goal oriented approach for managing the dialog that does not require manual identification of all possible scenarios was employed and is described in (Balchandran et al., 2009). The algorithm iteratively tries to achieve the goal (getting valid values for all address components), validating available input components, and prompting for missing input components, as defined by a priority order among components. This algorithm was implemented on a *state* based, programmable dialog manager as shown in Figure 1.

3 Scenarios

The following scenarios illustrate different situations that need to be handled by the dialog system when processing addresses.

3.1 Successful one-turn address recognition

Figure 2 shows the scenario where the user speaks a complete address in one sentence and the system recognizes it correctly.

3.2 One-turn address with error correction

The user speaks a complete address, but the system mis-recognizes the street name and number (Figure 3 (b)). The user requests to "go back" and the system re-prompts the user for the street name and number. User repeats the number in a different way (Figure 3 (c)) and the system gets it correctly.

3.3 Street and number around current location

In addition to complete addresses, the language models are built to include streets and numbers around the "current location" of the car. This data could be periodically updated based on changing car positions. In this example, (Figure 4) the user just specifies, "15 Lake View Drive" and the system defaults to the current city – Shelter Island, NY.

3.4 Ambiguous city

In this example, the user specifies an ambiguous city name (Figure 5 (a)). The system prompts the user to disambiguate by selecting a state. Once the user has done this, the system re-processes the street name and number to obtain the full address without needing the user to specify it again. The same concept is applied to other address components.

References

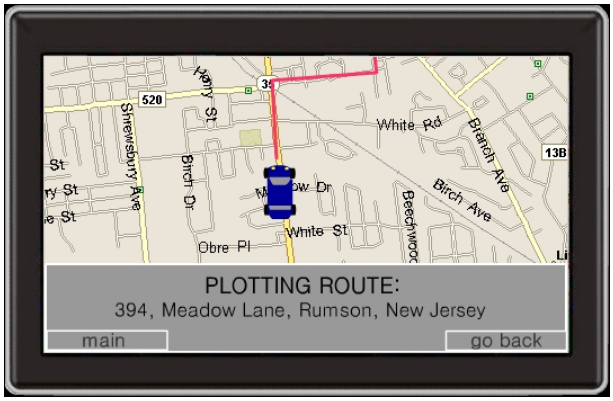
Rajesh Balchandran, Leonid Rachevsky, and Larry Sansone. 2009. Language modeling and dialog management for address recognition. In *Interspeech*.

Tomás Beran, Vladimír Bergl, Radek Hampl, Pavel Krbec, Jan Sedivý, Borivoj Tydlitát, and Josef Vopicka. 2004. Embedded viavoice. In *TSD*, pages 269–274.

John Gillett and Wayne Ward. 1998. A language model combining trigrams and stochastic context-free grammars. In *in International Conference on Spoken Language Processing*, volume 6, pages 2319–2322.

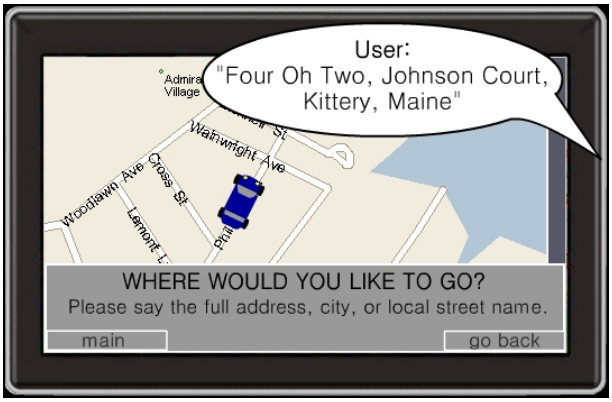
Roberto Sicconi, Kenneth White, and Harvey Ruback. 2009. Honda next generation speech user interface. In *SAE World Congress*, pages 2009–01–0518.

(a) User specifies address

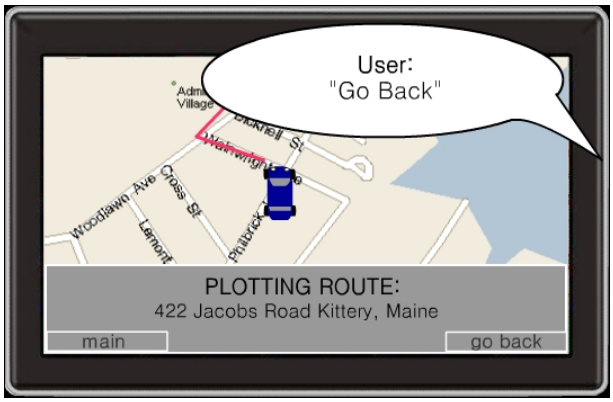


(b) System gets correct address

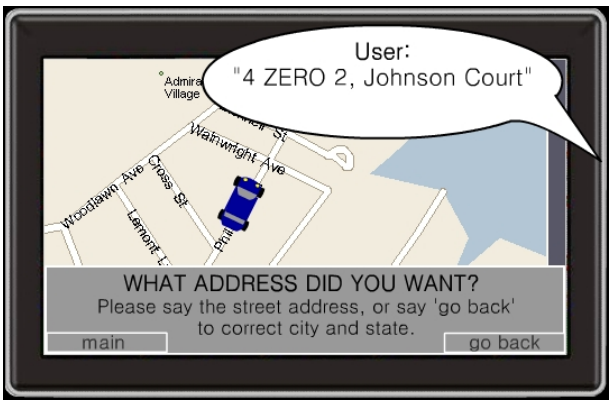
Figure 2: Successful one-turn address recognition



(a) User specifies address



(b) System makes mistake



(c) User corrects erroneous components

(d) System gets correct address

Figure 3: One-turn address recognition with error recovery

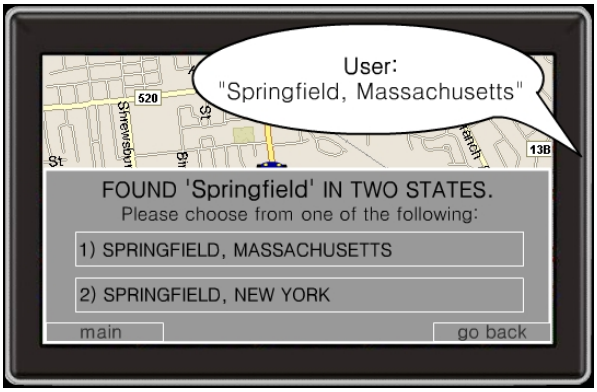
(a) User specifies street and number

(b) System locates street and number around current location

(c) System gets correct address

Figure 4: Street and number around current location (Shelter Island)

(a) User specifies address with city which is ambiguous



(b) User selects state and system combines previously specified information to get complete address

(c) System gets correct address

Figure 5: Ambiguous city example

Leveraging POMDPs trained with User Simulations and Rule-based Dialogue Management in a Spoken Dialogue System

Sebastian Varges, Silvia Quarteroni, Giuseppe Riccardi, Alexei V. Ivanov, Pierluigi Roberti
Department of Information Engineering and Computer Science
University of Trento
38050 Povo di Trento, Italy
{varges|silviaq|riccardi|ivanov|roberti}@disi.unitn.it

Abstract

We have developed a complete spoken dialogue framework that includes rule-based and trainable dialogue managers, speech recognition, spoken language understanding and generation modules, and a comprehensive web visualization interface.

We present a spoken dialogue system based on Reinforcement Learning that goes beyond standard rule based models and computes on-line decisions of the best dialogue moves. Bridging the gap between handcrafted (e.g. rule-based) and adaptive (e.g. based on Partially Observable Markov Decision Processes - POMDP) dialogue models, this prototype is able to learn high rewarding policies in a number of dialogue situations.

1 Reinforcement Learning in Dialogue

Machine Learning techniques, and particularly Reinforcement Learning (RL), have recently received great interest in research on dialogue management (DM) (Levin et al., 2000; Williams and Young, 2006). A major motivation for this choice is to improve robustness in the face of uncertainty due for example to speech recognition errors. A second important motivation is to improve adaptivity w.r.t. different user behaviour and application/recognition environments.

The RL approach is attractive because it offers a statistical model representing the *dynamics* of the interaction between system and user. This contrasts with the supervised learning approach where system behaviour is learnt based on a fixed corpus. However, exploration of the range of dialogue management strategies requires a simulation environment that includes a simulated user (Schatzmann et al., 2006) in order to avoid the prohibitive cost of using human subjects.

We demonstrate various parameters that influence the learnt dialogue management policy by using pre-trained policies (section 5). The application domain is a tourist information system for accommodation and events in the local area. The domain of the trained DMs is identical to that of a rule-based DM that was used by human users (section 4), allowing us to compare the two directly.

2 POMDP demonstration system

The POMDP DM implemented in this work is shown in figure 1: at each turn at time t, the incoming N user act hypotheses $a_{n,u}$ split the state space S_t to represent the complete set of interpretations from the start state (N=2). A belief update is performed resulting in a probability assigned to each state. The resulting ranked state space is used as a basis for action selection. In our current implementation, belief update is based on probabilistic user responses that include SLU confidences. Action selection to determine system action $a_{m,s}$ is based on the best state (m is a counter for actions in action set A). In each turn, the system uses an ϵ-greedy action selection strategy to decide probabilistically if to exploit the policy or explore any other action at random. (An alternative would be softmax, for example.) At the end of each dialogue/session a reward is assigned and policy entries are added or updated for each state-action pair involved. These pairs are stored in tabular form. We perform Monte Carlo updating similar to (Levin et al., 2000):

$$Q_t(s,a) = R(s,a)/n + Q_{t-1} \cdot (n-1)/n \quad (1)$$

where n is the number of sessions, R the reward and Q the estimate of the state-action value.

At the beginning of each dialogue, a user goal U_G (a set of concept-value pairs) is generated randomly and passed to a user simulator. The user simulator takes U_G and the current dialogue context to produce plausible SLU hypotheses. These

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 156–159,
Queen Mary University of London, September 2009.

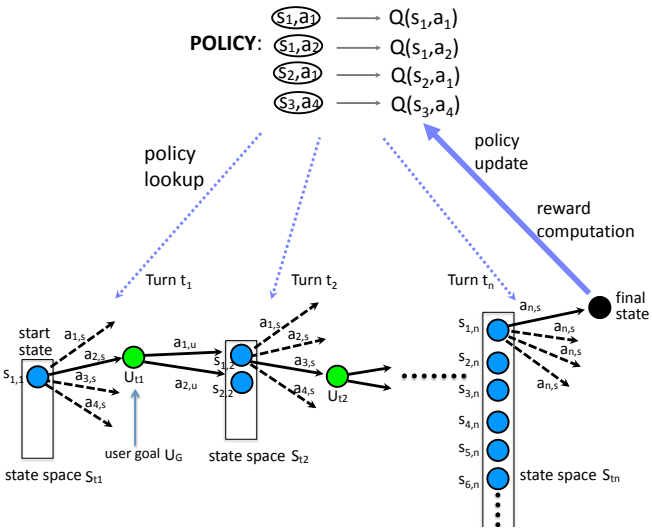


Figure 1: POMDP Dialogue Manager

are a subset of the concept-value pairs in U_G along with a confidence estimate bootstrapped from a small corpus of 74 in-domain dialogs. We assume that the user 'runs out of patience' after 15 turns and ends the call.

The system visualizes POMDP-related information live for the ongoing dialogue (figure 2). The visualization tool shows the internal representation of the dialogue manager including the the N best dialogue states after each user utterance and the reranking of the action set. At the end of each dialogue session, the reward and the policy updates are shown, i.e. new or updated state entries and action values. Moreover, the system generates a plot that relates the current dialogue's reward to the reward of previous dialogues.

3 User Simulation

To conduct thousands of simulated dialogues, the DM needs to deal with heterogeneous but plausible user input. We designed a User Simulator (US) which bootstraps likely user behaviors starting from a small corpus of 74 in-domain dialogs, acquired using a rule-based version of the system (section 4). The role of the US is to simulate the output of the SLU module to the DM during the whole interaction, fully replacing the ASR and SLU modules. This differs from other user simulation approaches where n-gram models of user dialog acts are represented.

For each simulated dialogue, one or more user goals are randomly selected from a list of possible user goals stored in a database table. A goal is represented as the set of concept-value pairs defining a task. Simulation of the user's behaviour happens in two stages. First, a *user model*, i.e. a model of the user's intentions at the current stage of the dialogue, is created. This is done by mining the previous system move to obtain the concepts required by the DM and their corresponding values (if any) from the current user goal. Then, the output of the user model is passed to an *error model* that simulates the "noisy channel" recognition errors based on statistics from the dialogue corpus. Errors produce perturbations of concept values as well as phenomena such as *noInput*, *noMatch* and *hangUp*. If the latter phenomena occur, they are directly propagated to the DM; otherwise, plausible confidences (based on the dialogue corpus) are attached to concept-value pairs. The probability of a given concept-value observation at time $t+1$ given the system move at time t, $a_{s,t}$, and the session user goal g_u, called $P(o_{t+1}|a_{s,t}, g_u)$, is obtained by combining the outputs of the error model and the user model:

$$P(o_{t+1}|a_{u,t+1}) \cdot P(a_{u,t+1}|a_{s,t}, g_u)$$

where $a_{u,t+1}$ is the true user action. Finally, concept-value pairs are combined in an SLU hypothesis and, as in the regular SLU module, a cumulative utterance-level confidence is computed, determining the rank of each of the N hypotheses output to the DM.

4 Rule-based Dialogue Management

A rule-based DM was developed as a meaningful comparison to the trained DM, to obtain training data from human-system interaction for the US, and to understand the properties of the domain. Rule-based dialog management works in two stages: retrieving and preprocessing facts (tuples) taken from a dialogue state database, and inferencing over those facts to generate a system response. We distinguish between the 'context model' of the first phase – essentially allowing more recent values for a concept to override less recent ones – and the 'dialog move engine' of the second phase. In the second stage, acceptor rules match SLU results to dialogue context, for example perceived user concepts to open questions. This may result in the decision to verify the application parameter in question, and the action is verbalized by language generation rules. If the parameter is accepted, application dependent task

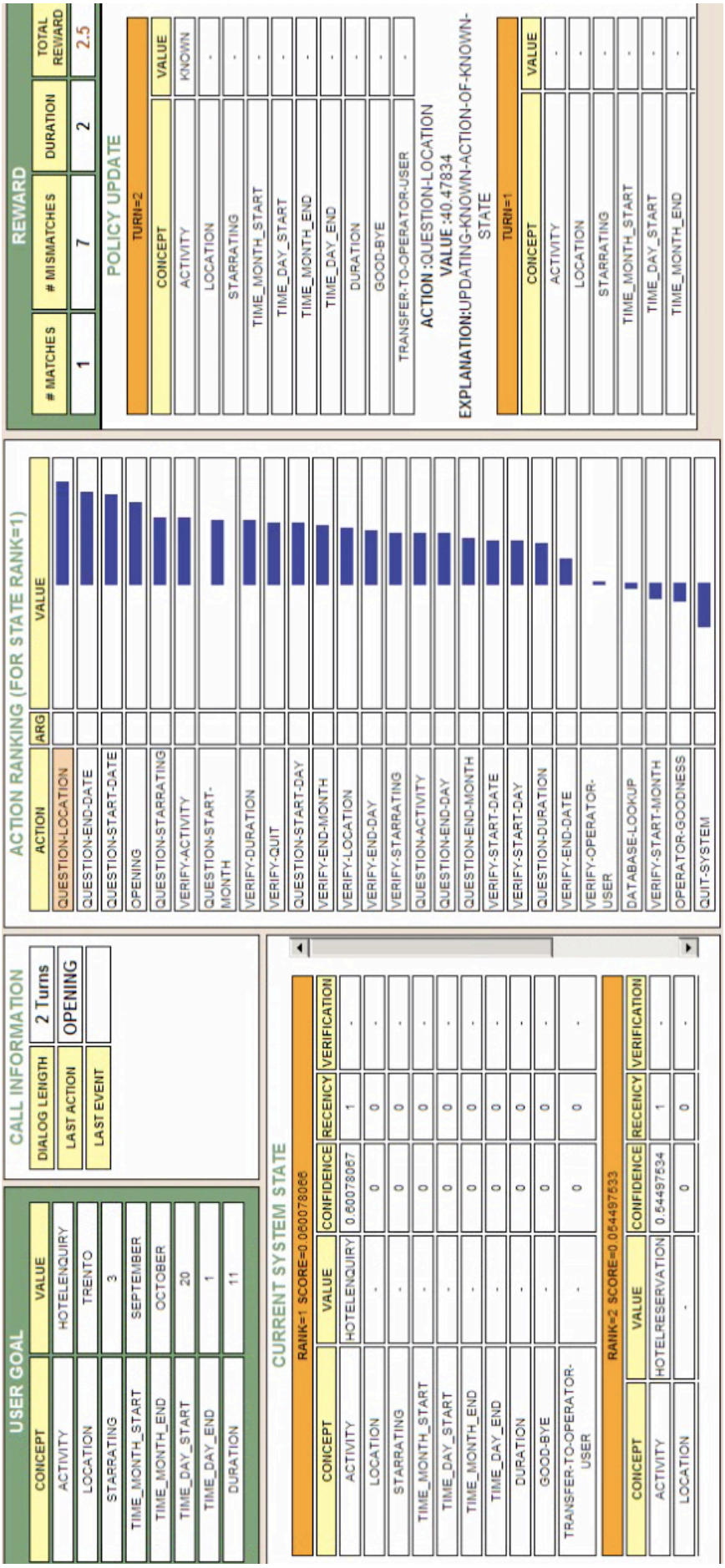


Figure 2: A screenshot of the online visualization tool. Left: user goal (top), evolving ranked state space (bottom). Center: per state action distribution at turn t_i. Right: consequent reward computation (top) and policy updates (bottom). See video at `http://www.youtube.com/watch?v=69QR0tKKhCw`.

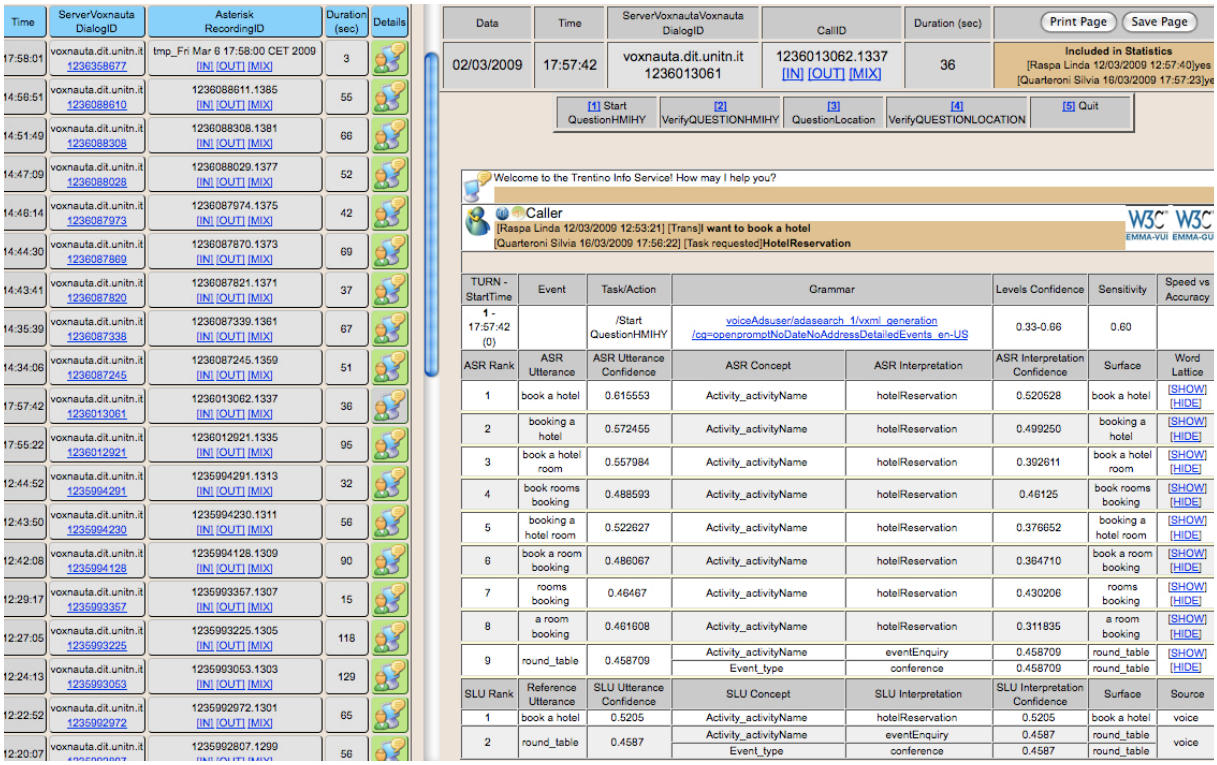


Figure 3: Left Pane: overview of a selection of dialogues in our visualization tool. Right Pane: visualization of a system opening prompt followed by the user's activity request. All *distinct* SLU hypotheses (concept-value combinations) deriving from ASR are ranked based on concept-level confidence (2 in this turn).

rules determine the next parameter to be acquired, resulting in the generation of an appropriate request. See (Varges et al., 2008) for more details.

5 Visualization Tool

In addition to the POMDP-related visualization tool (figure 2), we developed another web-based dialogue tool for both rule-based and POMDP system that displays ongoing and past dialogue utterances, semantic interpretation confidences and distributions of confidences for incoming user acts (see dialogue logs in figure 3).

Users are able to talk with several systems (via SIP phone connection to the dialogue system server) and see their dialogues in the visualization tool. They are able to compare the rule-based system, a randomly exploring learner that has not been trained yet, and several systems that use various pre-trained policies. The web tool is available at `http://cicerone.dit.unitn.it/DialogStatistics/`.

Acknowledgments

This work was partially supported by the European Commission Marie Curie Excellence Grant for the ADAMACH project (contract No. 022593) and by LUNA STREP project (contract No. 33549).

References

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).

J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review*, 21(2):97–126.

Sebastian Varges, Giuseppe Riccardi, and Silvia Quarteroni. 2008. Persistent information state in a data-centric architecture. In *Proc. 9th SIGdial Workhop on Discourse and Dialogue*, Columbus, Ohio.

J. D. Williams and S. Young. 2006. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech and Language*, 21(2):393–422.

Speeding Up the Design of Dialogue Applications by Using Database Contents and Structure Information

L. F. D'Haro, R. Cordoba, J. M. Lucas, R. Barra-Chicote, R. San-Segundo
Speech Technology Group
Dept. of Electronic Engineering
Universidad Politécnica de Madrid, Spain
{lfdharo, cordoba, juanmak, barra, lapiz}@die.upm.es

Abstract

Nowadays, most commercial and research dialogue applications for call centers are created using sophisticated and fully-feature development platforms. Surprisingly, most of them lack of some kind of acceleration strategy based on an automatic analysis of the contents or structure of the backend database. This paper describes our efforts to incorporate this kind of information which continues the work done in (D'Haro et al, 2006). Our main proposed strategies are: the generation of automatic state proposals for defining the dialogue flow network, the automatic selection of slots to be requested using mixed-initiative, the semi-automatic generation of SQL statements, and the quick generation of the data model of the application and the connection with the database fields. Subjective and objective evaluations demonstrate the advantages of using the accelerations and their high acceptance, both in our current proposals and in previous work.

1 Introduction

Currently, the growing demand of automatic dialogue services for different domains, user profiles, and languages has led to the development of a large number of sophisticated commercial and research platforms that provide all the necessary components for designing, executing, deploying and maintaining such services with minimum effort and with innovative functions that make them interesting for developers and final users.

In their effort for accelerating the design, most commercial platforms provide several high-level tools to build multimodal and multilingual dialogue applications using widespread standards such as VoiceXML, CCXML, J2EE, RCP, SRGS, etc. In addition, they include state-of-the-art modules such as speech recognizers, high quality speech synthesizers, language identification capabilities, etc., that guarantees user satisfaction and interaction. In addition, they present a very user-friendly graphical interface that makes easy the development of very complex dialogues, besides the incorporation of predefined libraries for typical dialogues states such as requesting card or social security numbers, etc., and additional assistants for debugging, logging and simulate the service.

In contrast to commercial platforms, research or academic platforms (e.g. CSLU-RAD [1], DialogDesigner[2], Olympus[3], Trindi-kit[4], etc.) do not necessary incorporate all the above-mentioned features; especially because they are limited to the number of standards that they are able to handle and to the integration level with other platforms, as well as the number of capabilities that they can offer to the users and programmers. However, they allow more complex dialogue interactions, most of them are freely available as open source, and using third party modules it is possible to extend their functionalities.

Surprisingly, these platforms do not include any kind of acceleration strategies based on the contents or in the structure of the backend database that, as we will show, can provide important information for the design. Next, we will describe some examples of applications or dialogue systems that use data mining techniques or heuristic infor-

[1] http://cslu.cse.ogi.edu/toolkit/
[2] http://spokendialogue.dk/
[3] http://www.ravenclaw-olympus.org/
[4] http://www.ling.gu.se/projekt/trindi/trindikit/

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 160–169,
Queen Mary University of London, September 2009.

mation extracted from the database contents in order to create automatic dialogue services.

In (Polifroni and Walker, 2006), different data mining techniques are used to automate the selection of content data to be used in system initiative queries and to provide summarized answers. At runtime, the system automatically selects the attributes to constrain the prompt queries that narrow down best the interaction flow with the final users.

In (Chung, 2004), the database is used together with a simulation system in order to generate thousands of unique dialogues that can be used to train the speech recognizer and the understanding module, as well as to diagnose the system behaviour against problematic user's interactions or answers.

In (Pargellis et al, 2004), a complete platform to build voice services where the database contents change constantly is described. At runtime, the system retrieves information that the user is interested in according to his personal profile. In addition, the system is able to create automatically dynamic speech grammars and prompts, as well as the dialogue flow for presenting information to the user, or for solving some interaction errors through predefined dialogue templates.

Finally, (Feng et al, 2003) proposes a very different approach, not using a database but mining the content of corporate websites for automatically creating spoken and text-based dialogue applications for custom care. Although the dialogue flow is predefined, it is interesting to see that important knowledge, for the different modules of the dialogue system, can also be extracted and used from a well-designed content.

In this work, we have solved some of the limitations of current platforms by incorporating successfully heuristic information into the different assistants of the platform and allowing them to collaborate between each other in several ways, as they collect the information already provided in the first stages of the design to improve and accelerate the design in the last stages. This way, the platform assistants classify which fields of the database could be relevant for the design, generate different kinds of automatic proposals according to the design step, reduce the information displayed to the designer, and accelerate different typical procedures required to define the application.

The paper is organized as follows: section 2 provides an overview of the overall architecture of the platform, including a brief description of the main assistants and layers that makes it up. Section 3 describes previous accelerations in the platform related with the current work; Sections 4, 5, and 6 describe in detail the new strategies and the assistants that include them. Section 7 describes the subjective and objective evaluations, and section 8 outlines some conclusions and future work.

2 Platform Architecture

The Application Generation Platform (AGP), created during the European project GEMINI, is an open and modular architecture made up of different assistants and tools that simplifies the generation of multimodal and multilingual dialogue applications with a high adaptability to different kinds of services (see Figure 4 in Appendix A). The platform consists of three main layers integrated into a common graphical user interface (GUI) that guides the designer step-by-step and lets him go back and forth.

In the first layer, called Framework Layer, the designer specifies global aspects related to the application and the data. This layer includes the Data Model Assistant (DMA), where the database structure is created, and the Data Connector Model Assistant (DCMA), where the application specific database access functions are created.

The next layer, called Retrieval layer, includes the State Flow Model Assistant (SFMA) and the Retrieval Model Assistant (RMA). The former is used to create the dialogue flow at an abstract level, by specifying the high-level states of the dialogue, plus the slots to ask to the user and the transitions among states. Then, the later is used to include all the actions (e.g., variables, loops, if-conditions, math or string operations, conditions for making transitions between states, calls to dialogs to provide/obtain information to/from the user) to be done in each state defined previously.

Finally, the third layer, called Dialogs Layer, contains the assistants that complete the general flow specifying for each dialogue the details that are modality and language dependent. For instance, the prompts and grammars for each language and modality, the definition of user profiles, the appearance and contents of the Web pages, the error treatment for speech recognition errors or Internet access, the presentation of information on screen or using speech, etc., are defined. Furthermore, the

VoiceXML and xHTML scripts used by the real-time system are automatically generated.

3 Previous Acceleration Strategies

In (D'Haro et al, 2006) and (D'Haro et al, 2004), we described several acceleration strategies based on using the data model structure and applied them successfully to different assistants of the platform, with a special emphasis in the assistant for defining the actions to be done in each dialogue (i.e. RMA). The data model information was used to:

a.) Create configurable and generic dialogue proposals for obtaining (called DGet) and for showing (called DSay) information from/to the user. In this case, the assistant creates a DGet or DSay dialogue for each class and attribute defined.

b.) Automatically propose the actions required for completing the information for each state of the dialogue flow; basically, the assistant proposes the dialogues to ask information to the user, the database access functions, and the dialogues to show information to the user. Figure 1 shows an example of the proposals for a banking application. In this example, the designer is editing a dialogue where given a currency name the system provides its specific information (buy and sell price, general information, etc.). Using the proposal window, all the designer would need to do is to select the corresponding DGet in the window (*DGet_CurrencyName_IN_CLASS_Currency*), then the database access function *GetCurrencyByName*, and finally the DSays that provide the desired attributes from the currency. In order to provide these proposals, we use the information of the relationships between slots and arguments of the database functions and the attributes and classes in the data model (section 5 and 6). When there is no relationship specified, we apply relaxed filters such as matching in types, similarity in names, or same number of arguments and slots in the state.

c.) Automate the process of passing information among actions/dialogues by proposing the variables that best match the connections or allowing the creation of new variables when no match exists. This is a critical aspect of dialogue applications design. Several actions and states have to be 'connected' as they use the information from the preceding dialogues. In general, most current design platforms allow the same kind of functionality, offering the user a selectable list of all the available variables in the dialogue. In other cases, especially considering the connections with database access functions, some platforms only allow the designer to define the matching by modifying by hand the script code. In this acceleration, we have tried to provide a better solution by automating the connection through automatic proposals. The assistant detects the input/output variables required in each action and offers the most suitable already defined variable of a compatible type; if there are more than one variable to show, the assistant sorts them according to the name similarity between variable and dialogue. If there is no compatible variable already defined in the system or the name proposed is not desired, the assistant allows the creation of a new local/global variable. Additionally, the assistant includes a window where all this matching can be edited.

Other accelerations included in this assistant were the quick creation of mixed-initiative dialogues, dialogues with over-answering (that do not exist in any current dialogue platform) and the quick definition of dialogue variables.

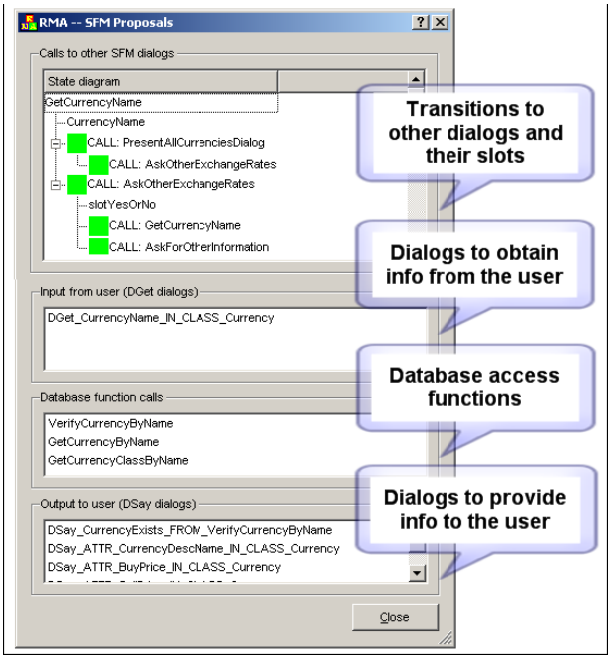


Figure 1. Example of automatic dialogues and database access function proposals

In the present work, the new accelerations additionally exploit the database contents and have been incorporated into the assistant to define the data model structure (section 4), into the assistant for defining the database access functions (section

5), and into the assistant to define the states of the dialogue flow (section 6). The next sections describe in detail these assistants and accelerations.

4 Strategies Applied to the Data Model Assistant (DMA)

This assistant helps in the creation of the data model structure of the service through a visual representation of all possible fields to be requested and presented to the user, which consists of object oriented classes and attributes. The goal with these classes and attributes is to provide information to the next assistants in the platform about which fields in the database are relevant for the service and the relationships between tables and fields.

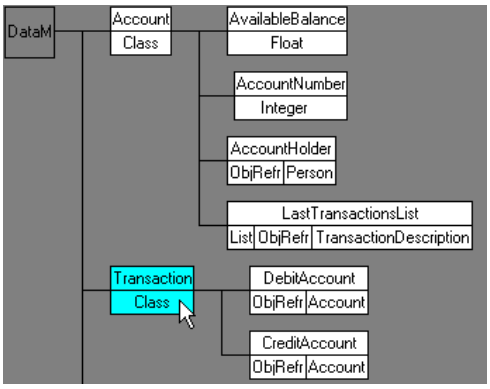


Figure 2. Example of class and attributes

Each class, see Figure 2, can be characterized by a list of attributes, a description, and optionally a list of base classes (inheriting their attributes). The attributes may be: a) of atomic types (e.g., string, Boolean, float, date, etc., e.g., *AvailableBalance*), b) complex objects, obtained by embedding or referring to an existing class (e.g., *AccountHolder*), or c) lists of either atomic type items or complex objects (e.g., *LastTransactionList*).

The main acceleration strategies, previously included in this assistant, are: a) re-utilization of libraries with models created beforehand, which can be copied totally or partially, or used to create a new class by mixing them, b) automatic creation of a class when it is referenced as an attribute inside another one, and c) definition of classes inheriting the attributes of a base class. Since this is one of the first assistants of the platform, a significant effort was done to accelerate the creation of the database structure and to include information about the relationships between the class attributes and the fields and tables in the database. To start with, the system generates and analyzes automatically heuristic information from the database contents. Then, with this information, the system proposes full custom classes and attributes that the designer can use when creating the data structure.

4.1 Extraction of heuristic information

The process is done using an open SQL query to retrieve information of every table, field and record in the database. This information includes the name and number of the tables and fields, and the number of records for every table. In addition, the following features for each field are also generated: a) field type, b) average length, c) number of empty records, d) language dependent fields, and e) the proportion of records that are different. This information is shared among the assistants in order to simplify the design or to improve the presentation of information in the posterior assistants. For instance, they are used for: (a) to accelerate the creation of the data model structure (section 4.2), (b) and (e) to unify slots as mixed initiative or not (see section 6.1), (c) to sort by relevance the attributes displayed by the wizard when creating the database structure (section 4.2), and (d) to not generate states for these fields in the SFMA since the dialogue flow in this assistant is language independent (section 6.1).

An important issue we found when retrieving the field type was that sometimes the metadata information provided by the SQL function was incorrect due to: a) the driver for accessing the database was only able to return a limited number of types, e.g. Boolean or dates were mapped as integer or string types respectively, b) the designer of the database defined the field using a generic type such as string or float when the visual inspection of the records showed that they actually corresponded to dates or integers, c) there were problems to map special types such as hyperlinks, binary, etc. into the types supported by our platform.

In order to solve these problems, we implemented a post-processing step based on using regular expressions to detect the following types: integers, floats, dates, strings, Boolean, mixed or empty fields. In general, the process is to analyze all non-empty records in a given field and to select as field type the one with more than the 90% of occurrences. Exceptions to this rule are: a) a numeric field is considered integer if all its records are classified as such, if not it is classified as float,

b) the empty type is assigned to fields with more than 95% of empty records.

In order to analyse the performance of the post-processing step, an objective evaluation was carried out. In this evaluation, twenty-one databases, most of them available online, were retrieved and visually inspected field by field. In total, there were 109 tables (an average of 5 tables per database), 767 fields and 610.506 records, which were classified by a human evaluator.

In our results, the average recognition was 89.6%, obtaining the best rates for dates, strings, and numeric quantities, which are the most common types in most databases. Analyzing in detail the misrecognitions, 0.9% of floats were incorrectly detected as integers due to values such as 2.0, 30.0, etc. which were automatically returned by the database driver without the decimal part. Another source of errors was detecting some numeric quantities due to special symbols such as dashes, percentages, or the euro symbol, which were incorrectly interpreted as a string type (3.3% and 1.6%). The major problems occurred for distinguishing between the String type and what we called Mixed type (i.e. fields containing: URLs, emails, long strings, etc.) since they are, in practice, the same. However, we wanted to separate them since for a speech recognizer they may be handled using different strategies (e.g. spelling, general grammars, etc.). The importance of these results is that they mean a reduction in the number of times the designer will need to change the proposed type for a given attribute when creating the classes (section 4.2).

4.2 Semi-automatic classes proposals

After collecting all the heuristics, the assistant provides a wizard window that allows the designer to create the attributes for a new class from the tables and fields of the database or from already existing classes in the model. The information of the selected field and table is saved in the definition of the class attribute allowing future assistants in the platform to access this information easily (section 5.1 and 6.1). The heuristic information is used to set automatically the field types in the wizard, although it can be edited by the designer. Besides, the wizard also proposes automatic alternative names for the new class and attributes when it detects duplicated names with already defined ones.

Finally, if the number of tables in the database is too high the designer can select those that will be really needed during the design, this way reducing the information displayed on the screen. In addition, it is also possible to customize the name of the tables in the database in order to make them more intuitive to the dialogue designer.

5 Strategies Applied to the Data Connector Model Assistant (DCMA)

This assistant allows the definition of the prototypes (i.e. only the input and output parameters) of the database access functions used in the runtime system. The advantage of using prototypes is that their actual implementation is not required during the design of the dialogue flow.

The main acceleration strategy, previously included in this assistant, was the possibility of relating the input/output arguments to the attributes and classes of the data model. This information is used by the retrieval model assistant to create dialogue proposals and to automatically propose database access functions for a given dialogue in the design (section 3). In this work, we have introduced a new acceleration by incorporating a wizard window that allows the creation and debugging of the SQL statements used at run-time.

5.1 Semi-automatic generation of SQL queries

The main motivation behind this wizard window was to simplify the process of creating the function prototypes (API), reducing the necessity of learning a new programming language (SQL), and to simplify the process of adding the query into the real-time modules and scripts. The new wizard semi-automatically creates the SQL statements for the given prototype and provides a pre-view of the results that the system would retrieve in the real-time system (see Figure 5 in Appendix A). This new acceleration is interesting since currently few development platforms include such kind of wizard forcing the designer to use third party software. Besides, current wizards only provide debugging tools, nice GUI features or support for many DB standards, but no automatic query proposals.

In order to automatically create the SQL statement, the assistant uses the input arguments (defined in the function prototype) as constraints for the WHERE clause, and the information of the

output arguments as returned fields for the SELECT clause. The assistant allows the inclusion of new input or output arguments if the function prototype is not complete or if the designer wants to test new combinations of arguments.

Finally, the wizard allows the designer to preview the records that the proposed SQL statement will retrieve at real-time. In order to debug the query, the designer specifies, using a pop-up window, the values for the input arguments of the function to test the query (as acceleration, the wizard automatically proposes real values retrieved from the database).

6 Strategies Applied to the State Flow Model Assistant (SFMA)

This assistant is used to define the dialogue flow at an abstract level, i.e. specifying only the high-level states of the dialogue, the slots to be asked to the user, and the transitions between states, but not the specific details of each state. The flow is specified using a state transition network representation that is common in this kind of platforms and dialogue modelling. The GUI allows the definition of new states using wizard-driven steps and a drag-and-drop interface. An important acceleration strategy from the previous version is the possibility of specifying the slots through attributes offered automatically from the data model. The new accelerations are the automatic proposal of the slots to be requested using system or mixed initiative dialogues (section 6.1) and the automatic generation of proposals of states for defining the dialogue flow (section 6.2).

6.1 Automatic unification of slots for mixed initiative

The idea of this acceleration is to allow the system to propose automatically when two or more slots must be requested one by one (using directed forms) or at the same time (using mixed initiative forms) according to the VoiceXML standard.

This functionality is only available when the slots to be analyzed have been defined from a table and field in the backend database. In this case, the assistant uses the heuristics obtained for the given fields and applies a set of customizable rules used to decide which slots can be unified and which ones cannot. Some examples of the rules applied to not propose the unification are: a) one of the slots is defined as a string with an average length greater than 20 characters, an average number of words per record greater than 3, and the other slot is an integer/float number greater than 5 characters. In this case, the rule avoids the recognition of long strings, e.g. an address or name, plus the recognition of long numeric quantities, e.g. phone or account numbers, b) when two slots are defined as strings and the sum of the average length of both is greater than 20 characters; in this case, the system tries to avoid the recognition of very long sentences. c) there are two numeric slots with a proportion of different values close to one, and the total number of records of both fields is high (configurable value), then the system determines that these slots have a large vocabulary and a high probability of misrecognition. So, in all these cases, the system decides that it is better to ask one slot at a time (system initiative). In case there are more than two slots, the system checks different combinations of the slots in order to find those that can be requested at the same time and leaving the other one to be requested alone.

6.2 Automatic states

In this strategy, the assistant creates automatically dialogue states that include the slots to be requested to the user. Using the information of the database structure and the database access functions, the wizard allows the designer to access to the following state proposals:

Empty states and already created states: The first one allows the creation of a new empty state, with no defined slots inside, that the designer can define completely afterwards. This way, we allow a top-down design. The second one allows the designer to re-use already defined states.

From attributes with database dependency: This kind of states is created from any attribute defined in the database model (DMA) that refers to a database field only if the attribute has been used as an input argument of any database access function. The proposed states contain only one slot and its name corresponds to the name of the attribute in the data model. However, the designer can select several states to create states with multiple slots.

From the database access functions: In this case, the system analyzes all the defined database functions containing input arguments defined as atomic types. Then, the system uses the name of the function as proposal for the name of the state,

and the input arguments as slots for that state. The assistant allows the designer to select several of these proposals in order to create more complex states. For instance, in case there is a database access function called *convertCurrencies*, which receives three input arguments (i.e. *fromCurrency*, *toCurrency*, and *Amount*), the system automatically creates a new state proposal called *convertCurrencies* that includes these three slots. Applying similar rules to the ones described in section 6.1 the system would propose to request the first two at the same time (mixed-initiative) and the *Amount* separately (directed forms).

From classes defined in the data model structure: In this case, the assistant creates a template that the designer can drag and drop into the workspace (see Figure 6 in Appendix A). Then, a pop-up window allows the designer to select the attributes to be used as slots. The assistant expands complex attributes (with inheritance and objects) allowing only the selection of atomic attributes.

7 Evaluation

With the objective of evaluating the performance of each of the acceleration strategies and assistants described above, we carried out a subjective and objective evaluation with 9 developers with different experience levels and profiles (4 novices, 3 intermediates, and 2 experts) on designing dialogue services. They were requested to fulfil different typical tasks covering each of the proposed accelerations and assistants to evaluate. Further details can be obtained in (D'Haro, 2009).

For the subjective evaluation, the participants were asked to answer a questionnaire that consists of four questions per assistant and seventeen for the overall platform, with a range between 1 and 10. This subjective evaluation confirms the designer-friendliness of the assistants, as well as their usability, since all the assistants obtained a global score higher than 8.0, which is a nice result. In detail, the DMA and DCMA obtained an 8.3, the SFMA a 9.0, the RMA an 8.6, and Diagen a 4.5. Regarding the acceleration strategies, see Figure 3a, the evaluators scored the automatic states with 9.3, the SQL generation and the unification of slots for mixed initiative with 9.0, and the class proposals with 8.9. Regarding the RMA and the accelerations related with the information extracted from the database (see section 3), the passing of arguments between actions and the proposal of dialogue actions obtained a 9.8 and 8.6 respectively.

For the objective evaluation, we collected the metrics proposed in (Jung et al, 2008): elapsed time, number of clicks, number of keystrokes, and number of corrections using the keyboard (keystroke errors). We compared our assistants with a built-in editor called Diagen, created during the GEMINI project and improved later on by (Hamerich, 2008), which features fewer accelerations but generates the same information specified by our assistants. As accelerations, Diagen only provides default templates that the designer has to complete and a guided procedure using different pop-up windows to fulfil the templates. The results confirm that the design time can be reduced, in average for all the assistants and tasks, in more than 45%, the number of keystrokes in 81%, and the number of clicks in 40%. Especially relevant is the high reduction (85%) obtained in the RMA considering that it is the main task in the design.

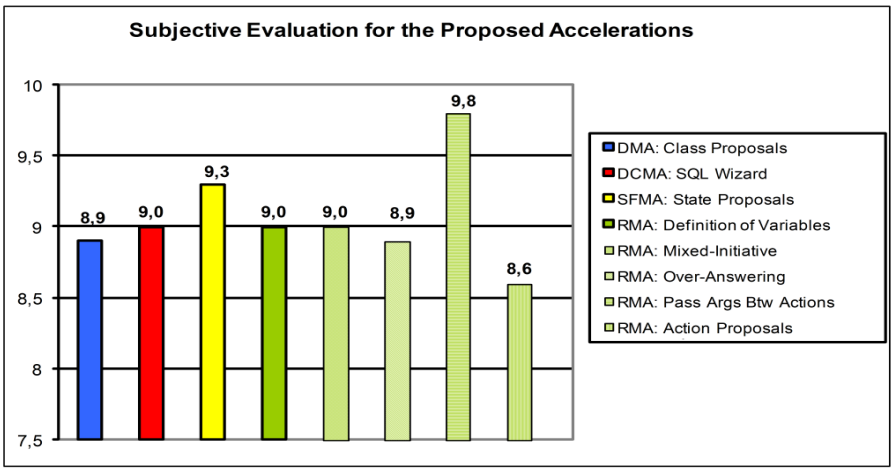
8 Conclusions and Future Work

In this paper, we have described the main accelerations incorporated into a complete platform for designing multimodal and multilingual dialogue applications. The proposed accelerations strategies are based on using information extracted from the contents of the backend database. The proposed accelerations include the creation of automatic state proposals, the unification of slots to be requested using mixed-initiative dialogues, and the semi-automatic creation and debugging of SQL statements for accessing the database, among others. Subjective and objective evaluations confirm that the proposed strategies are useful and contribute to simplify and accelerate the design.

As future work, we propose the extraction of new heuristic information, the creation of new rules for unifying slots for mixed-initiative dialogues. Considering the negative values in Figure 3b, we propose to improve the GUI for defining the connections among states in the SFMA, and to improve the DCMA by offering new automated methods for creating the prototypes.

9 Acknowledgements

This work has been supported by ROBONAUTA (DPI2007-66846-c02-02) and SD-TEAM (TIN2008-06856-C05-03).



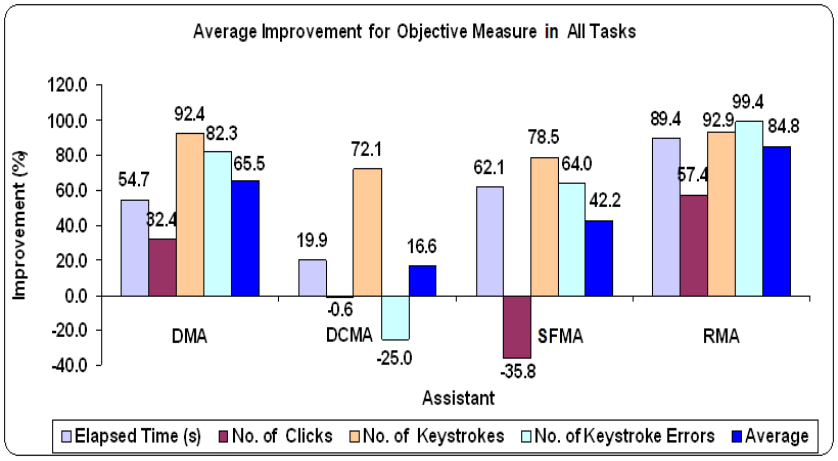


Figure 3. Average result for the: a) subjective evaluation for the accelerations, b) objective results

References

Chung, G. 2004. *Developing a Flexible Spoken Dialog System Using Simulation*. ACL 2004.

D'Haro. L. F. 2009. *Speed Up Strategies for the Creation of Multimodal and Multilingual Dialogue Systems*. PhD Thesis. Univ. Politécnica de Madrid.

D'Haro, L. F., Cordoba, R., et al. 2008. *Language Model Adaptation for a Speech to Sign Language Translation System Using Web Frequencies and a MAP framework*. Interspeech 2008, pp. 2119-2202.

D'Haro, L. F., Cordoba, R., et al. 2006. *An advanced platform to speed up the design of multilingual dialogue applications for multiple modalities* Speech Communication Vol. 48, Issue 8, pp. 863-887.

D'Haro, L. F., Cordoba, R., et al. 2004. *Strategies to reduce design time in multimodal/multilingual dialog applications*. ICSLP 2004, pp IV-3057–3060.

Feng, J., Bangalore, S., Rahim, M. 2003. *WEBTALK: Mining Websites for Automatically Building Dialog Systems.* ASRU 2003, pp. 168-173.

Hamerich, S. 2008. *From GEMINI to DiaGen: Improving Development of Speech Dialogues for Embedded Systems*. 9th SIGDIAL, pp. 92-95.

Jung, S., Lee, C., et. al. 2008. *DialogStudio : A Workbench for Data-driven Spoken Dialogue System Development and Management*. Speech Communications, 50 (8-9), pp. 683-697.

Pargellis, A. N., Kuo, H. J., Lee, C. 2004. *An automatic dialogue generation platform for personalized dialogue applications*. Speech Communication Vol. 42, pp. 329-351.

Polifroni, J. and Walker, M. 2006. *Learning Database Content for Spoken Dialogue System Design*. LREC 2006, pp. 143-148.

San-Segundo et al. 2008. *Speech to sign language translation system for Spanish*. Speech Communication Vol. 50, pp.1009–1020.

Appendix A. Additional Figures

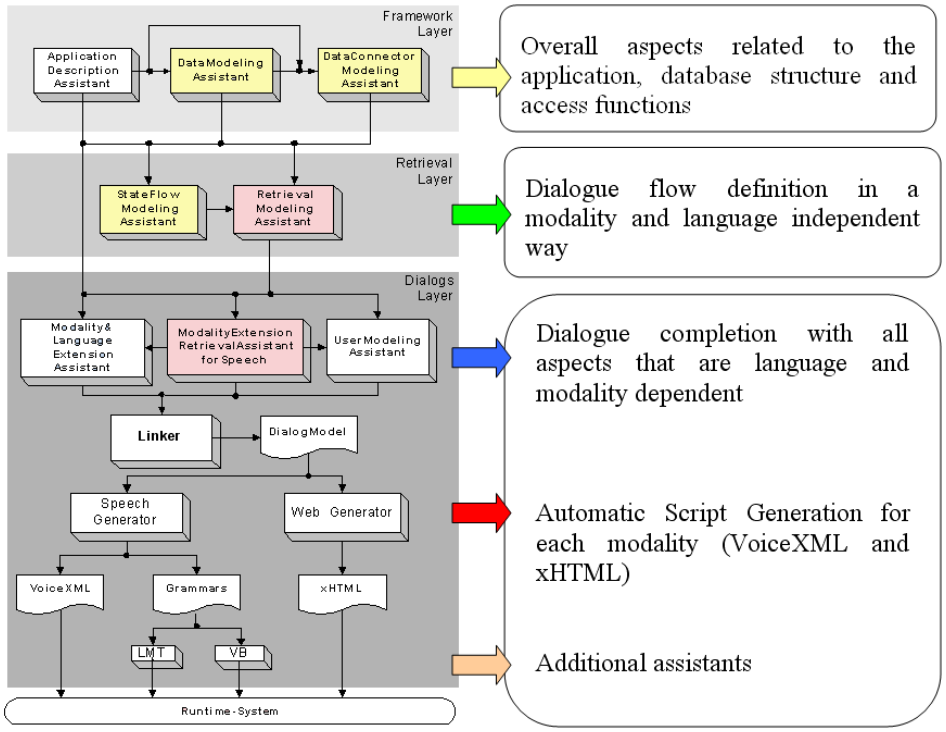


Figure 4. Platform architecture. In yellow colour the assistants with the new accelerations described in this paper. In pink colour assistants with previous accelerations (section 3)

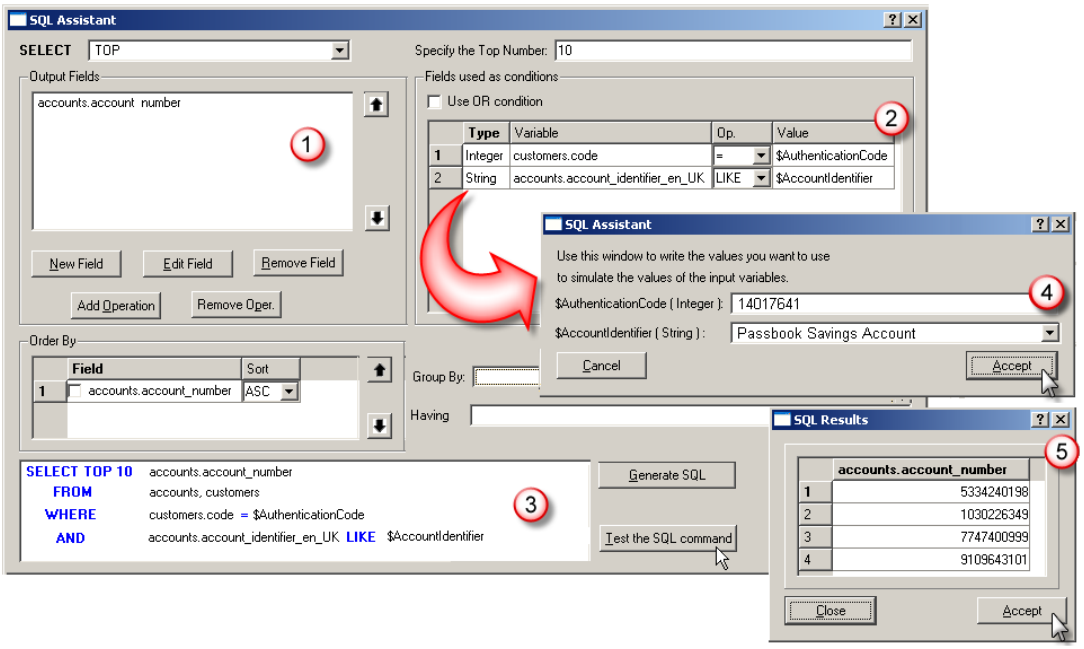


Figure 5. Wizard for creating and debugging the SQL statements for accessing the backend database. In the example, the proposed query allows the selection of all account numbers for a given customer (using his/her authentication code) and type of account (i.e. passbook saving accounts)

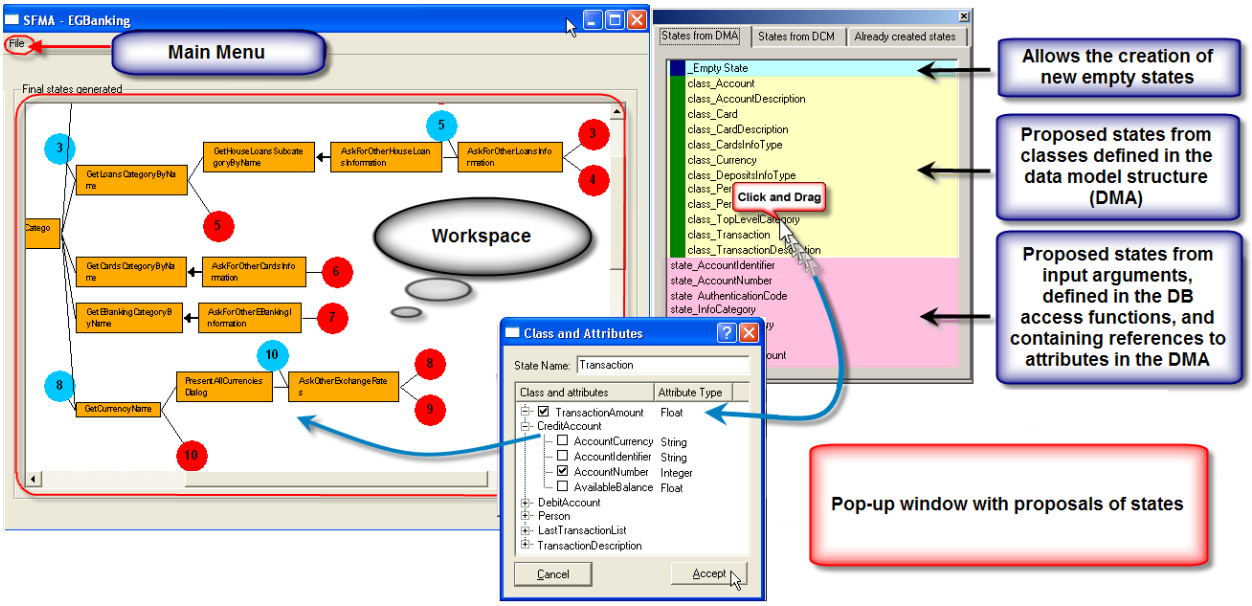


Figure 6. Workspace for creating the state transition network and pop up window with state proposals. In the example, the designer creates the state *Transaction* from the *Class_Transaction* template (created in the DMA, see Figure 2) and selects as slots the *TransactionAmount*, *CreditAccountNumber* and *DebitAccountNumber* (not shown)

Modeling User Satisfaction with Hidden Markov Models

Klaus-Peter Engelbrecht, Florian Gödde, Felix Hartard, Hamed Ketabdar, Sebastian Möller
Deutsche Telekom Laboratories, Quality & Usability Lab, TU Berlin,
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{Klaus-Peter.Engelbrecht,Florian.Goedde,
Hamed.Ketabdar,Sebastian.Moeller}@telekom.de
Felix.Hartard@Berlin.de

Abstract

Models for predicting judgments about the quality of Spoken Dialog Systems have been used as overall evaluation metric or as optimization functions in adaptive systems. We describe a new approach to such models, using Hidden Markov Models (HMMs). The user's opinion is regarded as a continuous process evolving over time. We present the data collection method and results achieved with the HMM model.

1 Introduction

Spoken Dialog Systems (SDSs) are now widely used, and are becoming more complex as a result of the increased solidity of advanced techniques, mainly in the realm of natural language understanding (Steimel et al. 2008). At the same time, the evaluation of such systems increasingly demands for testing the entire system, as components for speech recognition, language understanding and dialog management are interacting more deeply. For example, the system might search for web content on the basis of meaning extracted from an n-best list, and generate the reply and speech recognition grammars depending on the content found (Wootton et al. 2007). The performance of single components strongly depends on each other component in this case.

While performance parameters become less meaningful in such a system, the system's overall quality, which can only be measured by asking the user (Jekosch 2005), gains interest for the evaluation. Typically, users fill out questionnaires after the interaction, which cover various perceptional dimensions such as efficiency, dialog smoothness, or the overall evaluation of the system (Hone and Graham, 2001; ITU-T Rec. P.851, 2003; Möller 2005a). Judgments of the system's overall quality can be used to compare systems with respect to a single measure, which however comprises all relevant aspects of the interaction. Thus, the complexity of the evaluation task is reduced.

In addition, user simulation is increasingly used to address the difficulty of foreseeing all possible problems a user might encounter with the system (e.g. Ai and Weng, 2008; Engelbrecht et al., 2008a; Chung, 2004; López-Cózar et al., 2003). In order to evaluate results from such simulations, some approaches utilize prediction models of user judgments (e.g. Ai and Weng, 2008; Engelbrecht et al., 2008a).

Currently, prediction models for user judgments are based on the PARADISE framework introduced by Walker et al. (1997). PARADISE assumes that user satisfaction judgments describe the overall quality of the system, and are causally related to task success and dialog costs, i.e. efficiency and quality of the dialog. Therefore, a linear regression function can be trained with interaction parameters describing dialog costs and task success as predictors, and satisfaction ratings as the target. The resulting equation can then be used to predict user satisfaction with unseen dialogs.

In follow-up studies, it could be shown that such models are to some degree generalizable (Walker et al., 2000). However, also limitations of the models in predicting judgments for other user groups, or for systems with different levels of ASR performance, were reported (Walker et al., 1998). In the same study, prediction

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 170–177,
Queen Mary University of London, September 2009.

functions for user satisfaction were proposed to serve as optimization function in a system adapting its dialog strategy during the interaction. This idea is taken up by Rieser and Lemon (2008).

The prediction accuracy of PARADISE functions typically lies around an R^2 of 0.5, meaning that 50% of the variance in the judgments is explained by the model. While this number is not absolutely satisfying, it could be shown that mean values for groups of dialogs (e.g. with a specific system configuration) can be predicted more accurately than single dialogs with the same models (Engelbrecht and Möller, 2007). Low R^2 for the predictions of ratings of individual dialogs seems to be due to inter-rater differences at least to some degree. Such differences have been described, and may concern the actual perception of the judged issue (Guski, 1999), or the way the perception is described by the participant (Okun and Weir, 1990; Engelbrecht et al., 2008b)

We have tested the PARADISE framework extensively, using different classifier models and interaction parameters. Precise and general models are hard to achieve, even if the set of parameters describing the interaction is widely extended (Möller et al., 2008). In an effort to improve such prediction models, we developed two ideas:

- Predict the distribution of ratings which can be expected for a representative group of users given the same stimulus. This takes into account that in most cases the relevant user characteristics determining the judgment cannot be tracked, or even are unknown.
- Consider the time relations between events by modeling user opinion as a variable evolving over the course of the dialog. This way, time relations like co-occurrence of events, which affect quality perception, attention, or memory can be modeled most effectively.

In this paper, we present a new modeling approach considering these ideas. In Section 2, we introduce the topology of the model. Following this, we report how training data for the model were obtained from user tests in Section 3. Evaluation results are presented in Section 4 and discussed in Section 5, before we conclude with some remarks on follow-up research.

2 Modeling Judgments with HMMs

Hidden Markov Models (HMMs) are often used for classifying sequential stochastic processes, e.g. in computational linguistics or bio-informatics. An HMM models a sequence of events as a sequence of states, in which each state emits certain symbols with some probability. In addition, the transitions between states are probabilistic. The model is defined by a set of state symbols, a set of emission symbols, the probabilities for the initial state, the state transition matrix, and the emission matrix. The transition matrix contains the probabilities for transitions from each state to each other state or itself. The emission matrix contains the probabilities for each emission symbol to occur at each state.

While the sequence of emissions can be observed, the state sequence is hidden. However, given an emission sequence, standard algorithms defined for the HMM allow to calculate the probability of each state at each point in the sequence. The probability for the model to be in a state is dependent on the previous state and the emissions observed at the current state.

As illustrated by Figure 1, the development of the users' opinions can be modelled as an HMM. The user judgment about the dialog is modelled as states, each state representing a specific judgment (think of it as "emotional states"). A prediction is made at each dialog turn. In the model depicted, the user judgment can either be "bad" or "good". Each judgment has a probabilistic relation to the current events in the dialog. In the picture, the events are described in the form of understanding errors and confirmation types, i.e. there are two features which can take a number of different values, each with a certain probability.

Although the judgments do not "emit" the events at each turn (the causal relation is opposite), the probabilistic relation between them can be captured and evaluated with the HMM and the associated algorithms.

Apart from the dialog events, the current judgment is also determined by the previous judgment. For example, we expect that the judgments are varying smoothly, i.e. the probability for a transition becomes lower with increasing (semantic) distance between the state labels.

Although events in previous turns cannot impact the current judgment given this model topology, it is possible to incorporate dialog

history by creating features with a time lag. E.g., a feature could represent the understanding error in the previous turn. Also, simultaneity of different events affecting the quality perception can be evaluated by calculating probabilities for each judgment given the observed combination of features. If the features are interacting (i.e. the probability of one feature changes in dependence of another feature), this is modelled by directly specifying the emission probabilities for each combination of features. We call this a layer of emissions. Additional layers with other features can be created. In this case, the likelihood of each judgment given probabilities from each layer can be calculated by multiplication of the probabilities from each layer.

For the calculation of state probabilities, we can use forward recursion (Rabiner, 1989). The algorithm proceeds through the observed sequence, and at each step calculates the probability for each state given the probabilities of the observation, the probabilities of each state at the previous step, and the transition probabilities.

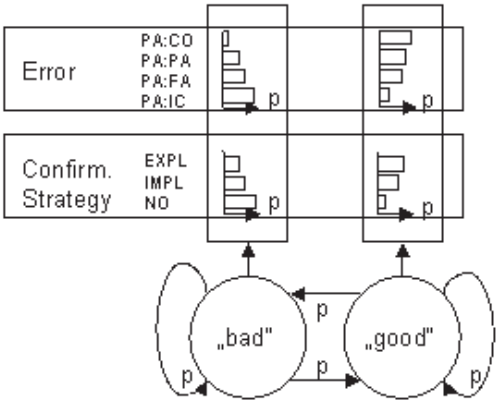


Figure 1. Topology of an HMM to model user judgments ("good" or "bad") in their probabilistic relation to dialog events (error and confirmation strategy) and the previous rating.

3 Data Collection

In order to train the model, data is needed in which the association between dialog events and judgments at each turn is represented. Therefore, we conducted an experiment in which users interacted with an SDS and judged the interaction after each dialog turn. In addition, all users followed the same five dialogs, which was possible by using the Wizard-of-Oz technique (Fraser and Gilbert 1991) and relatively strict scenario descriptions.

As the interactions were predefined, care was taken that a representative sample of relevant usability problems were passed in the interactions. Such problems were collected in a brainstorming session, and comprise understanding errors, task failure, prompt wording issues (such as incomplete information), missing control options, or not being able to input one of the criteria specified in the scenario description. As further features of the dialog, which might impact the quality judgment, the number of concepts which can be specified, as well as the confirmation strategy, should vary along the interactions.

We then designed a dialog system with a consistent strategy, which however allowed for the quality-relevant issues mentioned above to occur in the dialogs. The system was derived from the BoRIS restaurant information system (Möller 2005b), which enables the search of restaurants in the German city Bochum according to the criteria day, time, location, type of cuisine and pricing. For our system, we simplified the dialog strategy to system initiative, however, at the first turn, the users could say both day and time. If both were recognized, explicit confirmation was requested. If just one constraint was understood, confirmation was implicit. The last constraint specified (pricing) was not confirmed, and instead the restaurant information was output directly. If no restaurant was found according to the specified criteria, the user could change one of the constraints. Table 1 shows an example of a dialog with the system.

We also reformulated some prompts to fit the changed dialog strategy, and partly to incorporate potential wording problems. Prompts were recorded with a non-professional speaker,

S: Welcome to the telephone-based restaurant information system for Bochum. When would you like to eat?
U: Tomorrow night.
S: So you want to eat tonight?
U: No.
S: On which day would you like to eat?
….
U: Medium pricing
S: Unfortunately, I could not find a restaurant matching your query. Would you like to change time, location, cuisine or pricing?
U: Yes, cuisine.
S: Which type of cuisine are you searching for?
U: Italian.
S: The possible restaurants are: {name, address}

Table 1. Example dialog with the BoRIS restaurant information system, version as used in the experiment.

using high-quality audio equipment. During the interactions, the wizard simply replayed the prompt foreseen at the current state of the predefined dialog script. In addition to the foreseen prompts, the wizard had at hand no-input and help prompts in case the user would behave unexpectedly.

25 users (13 females, 12 males), recruited from the campus, but not all students, participated in the experiment. Participants were aged between 20 and 46 years (M=26.5; STD=6.6). Ratings were given on a 5-point scale, where the points were labeled "bad", "poor", "fair", "good", and "excellent". Ratings were input through a number pad attached to the scale. Each participant rehearsed the procedure with a test dialog. Before the experiment, all users filled out a questionnaire measuring their technical affinity.

As the data collected in the described experiment are all needed to train the prediction model for as many combinations of feature values as possible, we conducted a second experiment to generate test data. For this test, we asked 17 persons from our lab to conduct two dialogs with the system mock-up. The test setup was the same as in the previous experiment, except that new dialogs were created without particular requirements or restrictions.

In both experiments, not all users behaved as we hoped. Therefore, not all of the predefined dialog scripts were judged by all participants (N=15…23 for training corpus, N=9…13 for test corpus; N: number of valid dialogs). For one dialog script in the training corpus, the deviating interactions were all equal (N=9), so distributions of ratings per turn could be calculated for comparison with the predicted distributions for this dialog. For the training and calculation of initial state probabilities, all dialogs in the training corpus were used.

Feature	Values
understanding errors	PA:PA (partially correct) PA:FA (failed) PA:IC (incorrect)
confirmation strategy	explicit implicit none
system speech act	ask for 2 constraints ask for 1 constraint ask for selection of a constraint provide info
user speech act	provide info repeat info confirm meta communication no-input
contextual appropriateness (Grice's maxims)	manner quality quantity relevance
task success	success failure

Table 2. Annotated dialog features.

The model derived from the data includes five possible states (one for each rating). For a list of features annotated in the dialogs see Figure 2.

4 Results

In order to evaluate the modeling approach, we first searched for the best model given the training data from the first experiment. We then applied this model to the test data from the second experiment in order to evaluate the model accuracy given unseen data. Afterwards, we examined if another model trained on the training set can predict the test set better, i.e. we "optimized" the model on the test data. Finally, we cross-check how well the model optimized on the test data performs on the training data, which gives a glimpse at how much the model is biased towards the test data.

As the criterion for the optimization, we determined the mean squared error (*MSE*), and averaged across all dialog script in the corpus on which the model was optimized. For each dialog script, all 5 probabilities (ratings "bad" to "excellent") at each dialog turn were taken into account, i.e. the squared prediction errors were added. If *rate* is the rating, then

$$MSE_{dial} = \frac{\sum_{turn=1}^{n} \sum_{rate=1}^{5} \left[p_{emp}(rate) - p_{pred}(rate) \right]^2}{n}$$

As this measure, in the particular way we applied it here, is not easily comparable to other results, we add two pictures illustrating the accuracy represented either by a rather low or by a rather high *MSE*. In addition, we report the mean absolute error (MAE_{max}) of the models in predicting the most likely rating at each state (mean rating if two ratings with equal probability) and the baseline performance when the unconditional distribution of ratings is predicted.

We first optimized a model on the training data, meaning that we selected parameters, trained the HMM with these parameters on the training data and then predicted results for all 6 dialog scripts contained in the training set (top of

Table 3). The optimized model was chosen as the one returning the smallest *MSE* (mean of all tasks). The best model included understanding errors interacting with confirmation type at each turn, and interacting with task success. As we analyzed the prediction results, we found that whenever the system changed from asking two constraints at a time to just one (which is done in order to avoid multiple errors in a row), the predictions were too positive. We therefore introduced a new feature, which is annotated whenever the system asks for a single constraint which has been asked in a more complex question before ("dummy"). In the model optimized on training data, this parameter was included on a separate feature layer. That is, this feature impacts quality perception independent of the other features' values.

We then used this model to predict the test data collected in the second experiment (top of Table 4). As expected, the *MSE* clearly increases; however, this was partly due to the difference in the sample of participants. As in the second experiment participants were recruited from our lab, their technical affinity was relatively high. Therefore, we retrained the HMM with only those 50% of the users from the training set who got the highest score on the technical affinity questionnaire. With this model, the prediction of test data improved.

In a next step, we optimized the model on the test set meaning that we searched for the parameter combination achieving the best result on the two test dialogs. However, the model was still trained on the training data from the first experiment. As expected, the *MSE* could be improved. However, only minor changes in the feature configuration are necessary: Still, errors and confirmation type are interacting on the same layer. However, task success is included as independent variable on a second layer, and instead, the error in the previous turn determines the impact of errors and confirmation on the ratings. Again, we tested if the prediction can be

Predicted: training dialogs	**Dial 1**	**Dial 2**	**Dial 3**	**Dial 4**	**Dial 5**	**Dial 6**	**Mean (basel.)**
Optimized on training dialogs	*Layer 1: Error, Confirm, Task Success*						
	Layer 2: Dummy						
MSE:	0.0185	0.0307	0.0166	0.0216	0.0333	0.0477	0.0281 (0.1201)
MAE_{max}:	0.7000	0.5714	0.2857	0.0556	0.3636	0.3333	0.3849 (0.6167)
Optimized on test dialogs	*Layer 1: Errors, Errors_lag, Confirmation*						
	Layer 2: TaskSuccess						
MSE:	0.0272	0.0358	0.0247	0.0374	0.0400	0.0574	0.0371 (0.1201)
MAE_{max}:	0.5000	0.4286	0.4286	0.3889	0.4545	0.3333	0.4223 (0.6167)
Number of valid dialogs (N):	22	15	23	17	17	9	

Table 3. Evaluation of predictions of training dialogs (mean squared error and mean absolute error in predicting the most probable state at each turn). Baseline results are given in brackets. The feature combinations with which results were obtained are also reported.

Predicted: test dialogs	**Dial 1**	**Dial 2**	**Mean (baseline)**
Optimized on training dialogs	*Layer 1: Error, Confirm, Task Success*		
	Layer 2: Dummy		
MSE:	0.1039	0.0429	0.0734 (0.1583)
MAE_{max}:	0.4444	0.6250	0.5347 (0.6944)
Optimized on training dialogs (tah)	*Layer 1: Error, Confirm, Task Success*		
	Layer 2: Dummy		
MSE:	0.0957	0.0387	0.0672 (0.1636)
MAE_{max}:	0.3333	0	0.1667 (0.6944)
Optimized on test dialogs (rf)	*Layer 1: Errors, Errors_lag, Confirm*		
	Layer 2: TaskSuccess		
MSE:	0.0789	0.0349	0.0569 (0.1583)
MAE_{max}:	0.4444	0.6250	0.5347 (0.6944)
Optimized on test dialogs (tah; rf)	*Layer 1: Errors, Confirm*		
MSE:	0.0860	0.0374	0.0617 (0.1636)
MAE_{max}:	0.3333	0	0.1667 (0.6944)
Number of valid dialogs (N):	9	13	

Table 4. Evaluation of predictions of training dialogs (tah=model trained on users with high technical affinity; rf=user speech act feature exclude from analysis)

improved by considering differences between the users' technical affinity. However, repeating the procedure for only those users with high technical affinity did not improve the result this time. Concerning the parameters, error and confirmation type were confirmed to be significant predictors of quality judgments. The dummy parameter created to improve the accuracy on training data was not proven useful for the prediction of the test set ratings.

In order to cross-check the validity of the model optimized on test data, we finally predicted the ratings of the 6 dialogs from the training set with the same model (bottom of Table 3). As can be seen, the prediction is worse than that from the model optimized on the training set. However, the quality of the prediction is still reasonable, showing that the two datasets do not demand for completely different models. All predictions are above the baseline.

5 Discussion

In the previous section, we presented results achieved with our models in terms of *MSE*. In order to gain meaning to the values of *MSE*, we added the mean absolute error of predicting the most probable judgment at each state. A closer look at the relation between *MSE* and MAE_{max} reveals that both measures are not strictly correlated (see e.g. the first two models in Table 4). While the *MSE* measures the distance at each measurement point in the distribution, the MAE_{max} is a rough indicator of the similarity of the shape of the predicted and observed probability curves. The results for MAE_{max} are promising, as predictions of test data are in the range of predictions of training data and better than the baseline. Also, predictions made from participants with high technical affinity achieve better results on the test data in all cases, which was expected, but not found for the *MSE* results.

Figure 2 presents examples of prediction results graphically. We chose one example of an average, and one of a relatively bad prediction, to allow extrapolation to other results presented. The pictures show that even a relatively high *MSE* corresponds to a fair quality of the prediction. The probability curves are mostly similar, mainly smoother than the observed probability distributions. Sometimes the predictions are too optimistic, however, usually the change in judgments is predicted, just not the extent of this change. We can only hypothesize

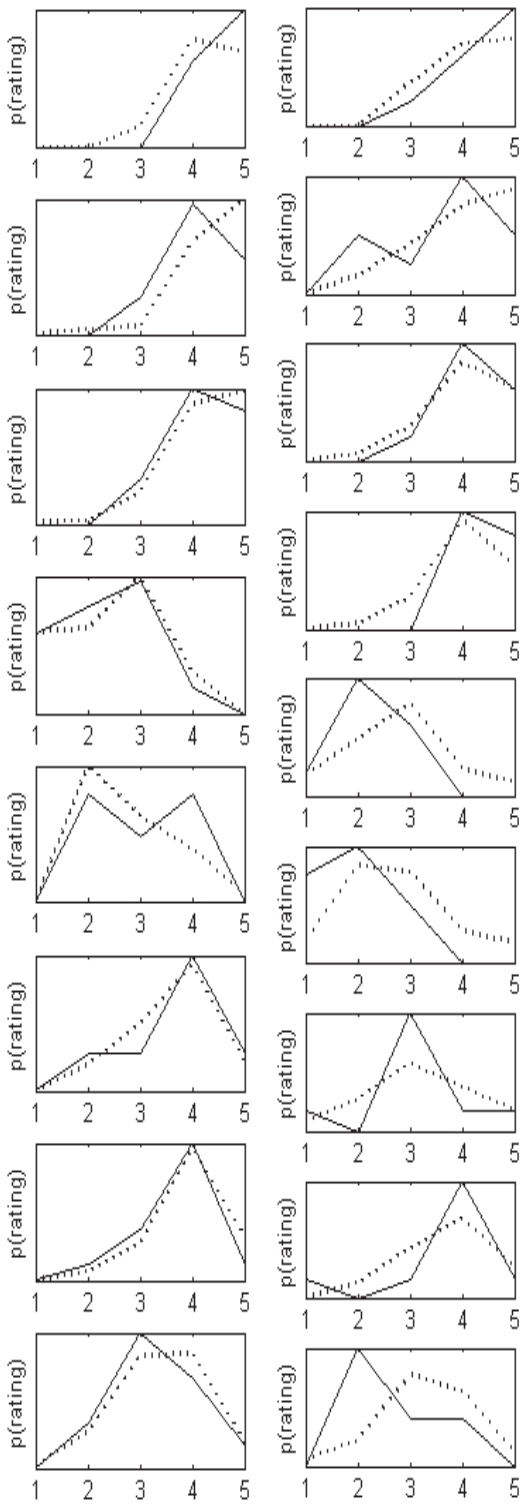


Figure 2. Examples of predictions on test data made with the model, illustrating the meaning of *MSE* values. Depicted are two dialogs (columns) with 9 (left) and 8 (right) turns (rows). For each turn, the empirical (solid line) and predicted (dotted line) rating distributions are given. Left: *MSE*=0.0957; N(emp)=9. Right: *MSE*=0.0349; N(emp)=13.

about the reasons for the participants to judge the respective dialog worse than predicted by the model. A possible reason is that users more easily decrease their judgments when the dialog has a longer history of problematic situations. According to our data, the users were relatively forgiving and increased their judgments if the dialog went well, even if previously errors had occurred. However, the errors might not really be forgot, and be reflected in the judgment of later problems and errors. Unfortunately, for reasons of data scarcity, the wider dialog history cannot be considered in the models.

Another source of prediction error might be the sample size available for the predicted dialogs. If sample size (N) and *MSE* values are compared among the dialogs, it can be observed that both values are correlated. This might be due to less smooth probability distribution curves if few ratings are available at each turn. While the curves depicted in Figure 2 are sometimes spiky, with increasing sample size normal distribution should be more likely. This might to some degree explain the clearly higher *MSE* for the test data predictions despite the relatively small error in predicting the most probable ratings.

6 Conclusion

In this paper, we presented a new approach to the prediction of user judgments about SDSs, using HMMs. The approach allows predicting the users' judgments at each step of a dialog. In predicting the distribution of ratings of many users, the approach takes into account differences between the users' judgment behaviors. This increases the usefulness of the model for a number of applications. E.g., in adaptive systems, the decision process can take into account differences between the users which cannot be attributed to user characteristics known to the system. If the model is applied to automatically generated dialogs, e.g. in the MeMo workbench (Engelbrecht et al., 2008a), a more detailed prediction of user satisfaction is enabled, allowing analysis on a turn-by-turn basis.

In addition, the approach facilitates the analysis of models and features affecting the quality ratings, as results can be compared to the empirical ratings with more detail. We hope to gain further insight into the relations between interaction parameters and user judgments by running simulations under different assumptions of relations between these entities.

A drawback of the approach is the generation of training data. The models presented in this paper cannot be assumed to be general, and in particular are lacking important parameters reflecting the timing in the dialogs. Therefore, as a next step the acquisition of judgments should be improved to be less disruptive for the interaction. In addition, it would be interesting to find a method for deriving the correct distributions of ratings at each dialog turn from a corpus of different dialogs, e.g. by grouping situations which are comparable. At the moment, we are also investigating if judgments can be acquired after the interactions without a loss in validity.

After all, the results we achieved with the model suggest that HMMs are suitable for modeling the users' quality perception of dialogs with SDSs. Further research on the topic will hopefully show if the dialog history has to be considered to a wider degree than in our present models.

Concerning dialog features and their relation to the judgments, the role of understanding errors in combination with the confirmation type could be established so far. More rich data are needed to work towards a general model for judgment predictions, including all relevant parameters. If judgments can be acquired after the interactions, we will be able to easily get the data needed for a better (and maybe complete) model. In any case, we are confident that the approach taken will allow a deeper analysis of the quality judgment process, which will enable progress by more analytical methods, such as formulating and testing hypotheses about this process.

References

Hua Ai, Fuliang Weng. 2008. *User Simulation as Testing for Spoken Dialog Systems.* Proc. of the 9th SIGdial Workshop on Discourse and Dialogue, Columbus, Ohio.

Grace Chung. 2004. *Developing a flexible spoken dialog system using simulation*. Proc. of the 42nd Annual Meeting on Association for Computational Linguistics, Barcelona, Spain.

Klaus-Peter Engelbrecht, Sebastian Möller. 2007. *Pragmatic Usage of Linear Regression Models for the Prediction of User Judgments*. Proc. of 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium.

Klaus-Peter Engelbrecht, Michael Kruppa, Sebastian Möller, Michael Quade. 2008a. *MeMo Workbench*

for Semi-Automated Usability Testing. Proc. of 9th Interspeech, Brisbane, Australia.

Klaus-Peter Engelbrecht, Sebastian Möller, Robert Schleicher, Ina Wechsung. 2008b. *Analysis of PARADISE Models for Individual Users of a Spoken Dialog System*. Proc. of ESSV 2008, Frankfurt/Main, Germany.

Klaus-Peter Engelbrecht, Felix Hartard, Florian Gödde, Sebastian Möller. 2009. *A Closer Look at Quality Judgments of Spoken Dialog Systems*, submitted to Interspeech 2009.

Norman M. Fraser, G. Nigel Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5:81–99.

Rainer Guski. 1999. Personal and Social Variables as Co-determinants of Noise Annoyance. *Noise & Health*, 3:45-56.

Kate S. Hone, Robert Graham. 2001. *Subjective Assessment of Speech-system Interface Usability*. Proc. of EUROSPEECH, Aalborg, Denmark.

ITU-T Rec. P.851, 2003. *Subjective Quality Evaluation of Telephone Services Based on Spoken Dialogue Systems*, International Telecommunication Union, Geneva, Switzerland.

Ute Jekosch. 2005. *Voice and Speech Quality Perception. Assessment and Evaluation*, Springer, Berlin, Germany.

Ramón López-Cózar, Ángel de la Torre, José C. Segura and Antonio J. Rubio. 2003. Assessment of Dialogue Systems by Means of a New Simulation Technique. *Speech Communication*, 40(3):387-407.

Sebastian Möller. 2005a. *Perceptual Quality Dimensions of Spoken Dialog Systems: A Review and New Experimental Results*, Proc. of Forum Acusticum, Budapest, Hungary.

Sebastian Möller. 2005b. *Quality of Telephone-based Spoken Dialog Systems*. Springer, New York.

Sebastian Möller, Klaus-Peter Engelbrecht, Robert Schleicher. 2008. Predicting the Quality and Usability of Spoken Dialogue Services, *Speech Communication*, 50:730-744.

Morris A. Okun, Renee M. Weir. 1990. Toward a Judgment Model of College Satisfaction. *Educational Psychological Review*, 2(1):59-76.

Lawrence R. Rabiner. 1989. A tutorial on HMM and selected applications in speech recognition. *Proc. IEEE*, 77(2):257-286.

Verena Rieser, Oliver Lemon. 2008. *Automatic Learning and Evaluation of User-Centered Objective Functions for Dialogue System Optimisation*. Proc. of LREC'08, Marrakech, Morocco.

Bernhard Steimel, Oliver Jacobs, Norbert Pfleger, Sebastian Paulke. 2008. *Testbericht VOICE Award 2008: Die besten deutschsprachigen Sprachapplikationen*. Initiative Voice Business, Bad Homburg, Germany.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, Alicia Abella. 1997. *PARADISE: A Framework for Evaluating Spoken Dialogue Agents.* Proc. of ACL/EACL 35th Ann. Meeting of the Assoc. for Computational Linguistics, Madrid, Spain.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, Alicia Abella. 1998. Evaluating Spoken Dialog Agents with PARADISE: Two Case Studies. *Computer Speech and Language*, 12:317-347.

Marilyn Walker, Candace Kamm, Diane Litman. 2000. Towards Developing General Models of Usability with PARADISE. *Natural Language Engineering*, 6(3-4):363-377.

Craig Wootton, Michael McTear, Terry Anderson. 2007. *Utilizing Online Content as Domain Knowledge in a Multi-Domain Dynamic Dialogue System*. Proc. of Interspeech 2007, Antwerp, Belgium.

Discourse Structure and Performance Analysis: Beyond the Correlation

Mihai Rotaru
Textkernel B.V.
Amsterdam, The Netherlands

mich.rotaru@gmail.com

Diane J. Litman
University of Pittsburgh
Pittsburgh, USA

litman@cs.pitt.edu

Abstract

This paper is part of our broader investigation into the utility of discourse structure for performance analysis. In our previous work, we showed that several interaction parameters that use discourse structure predict our performance metric. Here, we take a step forward and show that these correlations are not only a surface relationship. We show that redesigning the system in light of an interpretation of a correlation has a positive impact.

1 Introduction

The success of a spoken dialogue system (**SDS**) depends on a large number of factors and the strategies employed to address them. Some of these factors are intuitive. For example, problems with automated speech recognition can derail a dialogue from the normal course: e.g. non-understandings, misunderstandings, end-pointing, etc. (e.g. (Bohus, 2007; Raux and Eskenazi, 2008)). The strategies used to handle or avoid these situations are also important and researchers have experimented with many such strategies as there is no clear winner in all contexts (e.g. (Bohus, 2007; Singh et al., 2002)). However, other factors can only be inferred through empirical analyses.

A principled approach to identifying important factors and strategies to handle them comes from *performance analysis*. This approach was pioneered by the PARADISE framework (Walker et al., 2000). In PARADISE, the SDS behavior is quantified in the form of *interaction parameters*: e.g. speech recognition performance, number of turns, number of help requests, etc. (Möller, 2005).These parameters are then used in a multivariate linear regression to predict a SDS performance metric (e.g. task completion, user satisfaction: (Singh et al., 2002)). Finally, SDS redesign efforts are informed by the parameters that make it in the regression model.

Conceptually, this equates to investigating two properties of interaction parameters: **predictiveness** and **informativeness**[1]. Predictiveness looks at the connection between the parameter and system performance via predictive models (e.g. multivariate linear regression in PARADISE). Once the predictiveness is established, it is important to look at the parameter informativeness. Informally, informativeness looks at how much the parameter can help us improve the system. We already know that the parameter is predictive of performance. But this does not tell us if there is a causal link between the two. In fact, the main drive is not to prove a causal link but to show that the interaction parameter will inform a modification of the system and that this modification will improve the system.

This paper is part of our broader investigation into the utility of *discourse structure* for performance analysis. Although each dialogue has an inherent structure called the discourse structure (Grosz and Sidner, 1986), this information has received little attention in performance analysis settings. In our previous work (Rotaru and Litman, 2006), we established the predictiveness of several interaction parameters derived from discourse structure. Here we take a step further and demonstrate the informativeness of these parameters.

We show that one of the predictive discourse structure-based parameters (PopUp-Incorrect) informs a promising modification of our system.

[1] Although this terminology is not yet established in the SDS community, the investigations behind these properties are a common practice in the field.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 178–187,
Queen Mary University of London, September 2009.

We implement this modification and we compare it with the original version of the system through a user study. Our analyses indicate that the modification leads to objective improvements for our system (e.g. performance improvements for certain users but not at the population level and fewer system turns).

2 Background

ITSPOKE (**I**ntelligent **T**utoring **Spoke**n Dialogue System) (Litman et al., 2006) is a speech-enabled version of the text-based Why2-Atlas conceptual physics tutoring system (VanLehn et al., 2007). The interaction between ITSPOKE and users is mediated through a graphical web interface supplemented with a headphone-microphone unit. ITSPOKE first analyzes a user typed essay response to a physics problem for mistakes and omissions. Then it engages in a spoken dialogue to remediate the identified problems. Finally, users revise their essay and ITSPOKE either does another round of tutoring/essay revision if needed or moves on to the next problem.

While for most information access SDS performance is measured using task completion or user satisfaction, for the tutoring SDS the primary performance metric is learning. To measure learning, users take a knowledge test before and after interacting with ITSPOKE. The Normalized Learning Gain (**NLG**) is defined as (posttest-pretest)/(1-pretest) and measures the percentage improvement relative to the perfect improvement: an NLG of 0.0 means no improvement while an NLG of 1.0 means maximum improvement.

2.1 Discourse structure

We use the Grosz & Sidner theory of discourse (Grosz and Sidner, 1986). According to this theory, dialogue utterances naturally aggregate into discourse segments, with each segment having an associated purpose or intention. These segments are hierarchically organized forming the discourse structure hierarchy. This hierarchical aspect of dialogue has inspired several generic dialogue management frameworks (e.g. RavenClaw (Bohus, 2007)). We briefly describe our automatic annotation of this hierarchy and its use through discourse transitions. A sample example is shown in Appendix 1. For more details see (Rotaru and Litman, 2006).

Since dialogues with ITSPOKE follow a "tutor question - user answer - tutor response" format, which is hand-authored beforehand in a hierarchical structure, we can easily approximate the discourse structure hierarchy. After the essay analysis, ITSPOKE selects a group of questions which are asked one by one. These questions form the top-level discourse segment (e.g. DS1 in Appendix 1). For incorrect answers to more complex questions (e.g. applying physics laws), ITSPOKE will engage in a *remediation subdialogue* that attempts to remediate the student's lack of knowledge or skills. These subdialogues form the embedded discourse segments (e.g. DS2 in Appendix 2).

We define six *discourse transitions* in the discourse structure hierarchy and use them to label each system turn. A *NewTopLevel* label is used for the first question after an essay submission. If the previous question is at the same level with the current question we label the current question as *Advance*. The first question in a remediation subdialogue is labeled as *Push*. After a remediation subdialogue is completed, ITSPOKE will pop up and a heuristic determines whether to ask again the question that triggered the remediation dialogue. Reasking is labeled as a **PopUp**, while moving on to the next question is labeled as *PopUpAdv*. Rejections due to speech problems or timeouts are labeled as *SameGoal*.

Our transitions partially encode the hierarchical information of discourse structure: they capture the position of each system turn in this hierarchy relative to the previous system turn.

2.2 Discourse structure-based interaction parameters

To derive interaction parameters, we look at *transition–phenomena* and *transition–transition* bigrams. The first type of bigrams is motivated by our intuition that dialogue phenomena related to performance are not uniformly important but have more weight depending on their position in the dialogue. For example, it is more important for users to be correct at specific places in the dialogue rather than overall in the dialogue. We use two phenomena related to performance in our system/domain: user correctness (e.g. correct, incorrect) and user certainty (e.g. uncertain, neutral, etc.). For example, a PopUp-Incorrect event occurs whenever users are incorrect after being reasked the question that initially triggered the remediation dialogue. The second type of bigrams is motivated by our intuition that "good" and "bad" dialogues have different discourse structures. To compare two dialogues in terms of

the discourse structure we look at consecutive transitions: e.g. Push-Push.

For each bigram we compute 3 interaction parameters: a total (e.g. the number of PopUp-Incorrect events), a percentage (e.g. the number of PopUp-Incorrect relative to the number of turns) and a relative percentage (e.g. the percentage of times a PopUp is followed by an incorrect answer).

3 Predictiveness

In (Rotaru and Litman, 2006), we demonstrate the predictiveness of several discourse structure-based parameters. Here we summarize the results for parameters derived from the PopUp–Correct and **PopUp–Incorrect** bigrams (Table 1). These bigrams caught our attention as their predictiveness has intuitive interpretations and generalizes to other corpora. Predictiveness was measured by looking at correlations (i.e. univariate linear regression) between our interaction parameters and learning[2]. We used a corpus of 95 dialogues from 20 users (2334 user turns). For brevity, we report in Table 1 only the bigram, the best Pearson's Correlation Coefficient (R) associated with parameters derived from that bigram and the statistical significance of this coefficient (p).

Bigram	R	p
PopUp-Correct	**0.45**	**0.05**
PopUp-Incorrect	**-0.46**	**0.05**

Table 1. Several discourse structure-based parameters significantly correlated with learning (for complete results see (Rotaru and Litman, 2006))

The two bigrams shed light into user's learning patterns. In both cases, the student has just finished a remediation subdialogue and the system is popping up by reasking the original question again (a PopUp transition). We find that correct answers after a PopUp are positively correlated with learning. In contrast, incorrect answers after a PopUp are negatively correlated with learning. We hypothesize that these correlations indicate whether the user took advantage of the additional learning opportunities offered by the remediation subdialogue. By answering correctly the original system question (PopUp–Correct), the user demonstrates that he/she has absorbed the information from the remediation dialogue. This bigram is an indication of a successful learning event. In contrast, answering the original system question incorrectly (PopUp–Incorrect) is an indication of a missed learning opportunity; the more such events happen the less the user learns.

In (Rotaru and Litman, 2006) we also demonstrate that discourse structure is an important source for producing predictive parameters. Indeed, we found that simple correctness parameters (e.g. number of incorrect answers) are surprisingly not predictive in our domain. In contrast, parameters that look at correctness at specific places in the discourse structure hierarchy are predictive (e.g. PopUp–Incorrect).

4 Informativeness

We investigate the informativeness of the PopUp–Incorrect bigram as in (Rotaru, 2008) we also show that its predictiveness generalizes to two other corpora. We need 3 things for this: an interpretation of the predictiveness (i.e. an interpretation of the correlation), a new system strategy derived from this interpretation and a validation of the strategy.

As mentioned in Section 3, our interpretation of the correlation between PopUp–Incorrect events and learning is that these events signal failed learning opportunities. The remediation subdialogue is the failed learning opportunity: the system had a chance to correct user's lack of knowledge and failed to achieve that. The more such events we see, the lesser the system performance.

How can we change the system in light of this interpretation? We propose to *give additional explanations after a PopUp–Incorrect event* as the new strategy. To arrive at this strategy, we hypothesized why the failed opportunity has occurred. The simplest answer is that the user has failed to absorb the information from the remediation dialogue. It is possible that the user did not understand the remediation dialogue and/or failed to make the connection between the remediation dialogue and the original question. The current ITSPOKE strategy after a PopUp–Incorrect is to give away the correct answer and move on. The negative correlations indicate that this strategy is not working. Thus, maybe it would be better if the system will engage in additional explanations to correct the user. If we can make the user understand, then we transform the failed learning opportunity into a successful learning opportunity. This will be equivalent to a PopUp–Correct event which we have seen is *positively* correlated with learning (Section 3).

[2] As it is commonly done in the tutoring research (e.g. (Litman et al., 2006)), we use partial Pearson's correlations between our parameters and the posttest score that account for the pretest score.

While other interpretation and hypotheses might also be true, our results (Section 5) show that the new strategy is successful. This validates the interpretation, the strategy and consequently the informativeness of the parameter.

4.1 Modification

To modify the system, we had to implement the new PopUp–Incorrect strategy: provide additional explanations rather than simply giving away the correct answer and moving on. But how to deliver the additional explanations? One way is to engage in an additional subdialogue. However, this was complicated by the fact that we did not know exactly what information to convey and/or what questions to ask. It was crucial that the information and/or the questions were on target due to the extra burden of the new subdialogue.

Instead, we opted for a different implementation of the strategy: interrupt the conversation at PopUp–Incorrect events and offer the additional explanations in form of a *webpage* that the user will read (recall that ITSPOKE uses in addition a graphical web interface – Section 2). Each potential PopUp–Incorrect event had an associated webpage that is displayed whenever the event occurs. Because the information was presented visually, users can choose which part to read, which meant that we did not have to be on target with our explanations. To return to the spoken dialogue, users pressed a button when done reading the webpage.

All webpages included several pieces of information we judged to be helpful. We included the tutor question, the correct answer and a text summary of the instruction so far and of the remediation subdialogue. We also presented a graphical representation of the discourse structure, called the Navigation Map. Our previous work (Rotaru and Litman, 2007) shows that users prefer this feature over not having it on many subjective dimensions related to understanding. Additional information not discussed by the system was also included if applicable: intuitions and examples from real life, the purpose of the question with respect to the current problem and previous problems and/or possible pitfalls. See Appendix 2 for a sample webpage.

The information we included in the PopUp–Incorrect webpages has a "reflective" nature. For example, we summarize and discuss the relevant instruction. We also comment on the connection between the current problem and previous problems. The value of "reflective" information has been established previously e.g. (Katz et al., 2003).

All webpages and their content were created by one of the authors. All potential places for PopUp–Incorrect events (i.e. system questions) were identified and a webpage was authored for each question. There were 24 such places out of a total of 96 questions the system may ask during the dialogue.

5 Results

There are several ways to demonstrate the success of the new strategy. First, we can investigate if the correlation between PopUp–Incorrect and learning is broken by the new strategy. Our results (5.2) show that this is true. Second, we can show that the new system outperforms the old system. However, this might not be the best way as the new PopUp–Incorrect strategy directly affects only people with PopUp–Incorrect events. In addition, its effect might depend on how many times it was activated. Indeed, we find no significant effect of the new strategy in terms of performance at the population level. However, we find that the new strategy does produce a performance improvement for users that "needed" it the most: users with more PopUp–Incorrect events (5.3).

We begin by describing the user study and then we proceed with our quantitative evaluations.

5.1 User study

To test the effect of the new PopUp–Incorrect strategy, we designed and performed a between-subjects study with 2 conditions. In the control condition (***R***) we used the regular version of ITSPOKE with the old PopUp–Incorrect strategy (i.e. give the current answer and move on). In the experimental condition (***PI***), we had the regular version of ITSPOKE with the new PopUp–Incorrect strategy (i.e. give additional information).

The resulting corpus has 22 *R* users and 25 *PI* users and it is balanced for gender. There are 235 dialogues and 3909 user turns. The experiment took 2½ hours per user on average.

5.2 Breaking the correlation

The predictiveness of the PopUp–Incorrect bigram (i.e. its negative correlation with learning) means that PopUp–Incorrect events signal lower performance. One way to validate the effective-

ness of the new PopUp–Incorrect strategy is to show that it breaks down this correlation. In other words, PopUp–Incorrect events no longer signal lower performance. Simple correlation does not guarantee that this is true because correlation does not necessarily imply causality.

In our experiment, this translates to showing that that PopUp–Incorrect bigram parameters are still correlated with learning for *R* students but the correlations are weaker for *PI* students. Table 2 shows these correlations. As in Table 1, we show only the bigrams for brevity.

	R users		*PI* users	
Bigram	R	p	R	p
PopUp-Correct	**0.60**	**0.01**	0.18	0.40
PopUp-Incorrect	**-0.65**	**0.01**	-0.18	0.40

Table 2. Correlation with learning in each condition

We find that the connection between user behavior after a PopUp transition and learning continues to be strong for *R* users. PopUp–Incorrect events continue to signal lower performance (i.e. a strong significant negative correlation of -0.65). PopUp–Correct events signal increased performance (i.e. a strong significant positive correlation of +0.60). The fact that these correlations generalize across experiments/corpora further strengthens the predictiveness of the PopUp–Incorrect parameters.

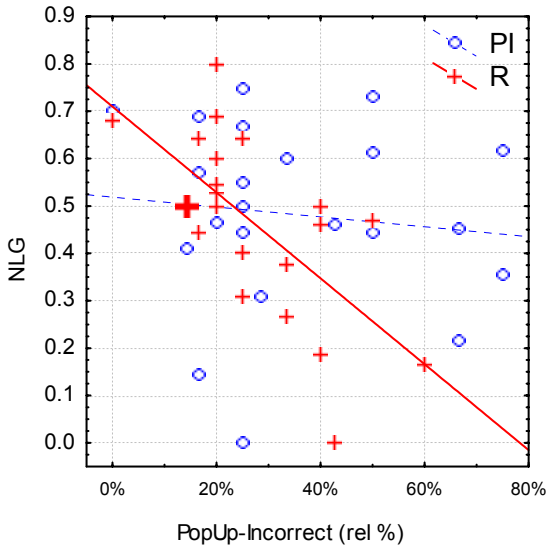


Figure 1. Correlations between a PopUp-Incorrect parameter and NLG

In contrast, for *PI* users these correlations are much weaker with non-significant correlation coefficients of -0.18 and 0.18 respectively. In other words the new PopUp–Incorrect strategy breaks down the observed correlation: PopUp–Incorrect events are no longer a good indicator of lower performance.

It is interesting to visualize these correlations graphically. Figure 1 shows a scatter plot of the PopUp–Incorrect relative percentage parameter and NLG for each *PI* and *R* user. The regression lines for the correlation between PopUp–Incorrect and NLG for *PI* and *R* are shown. The graph shows that users with less PopUp–Incorrect events (e.g. less than 30% relative) tend to have a higher NLG (0.5 or higher) regardless of the condition. However, for users with more PopUp–Incorrect events, the behavior depends on the condition: *R* users (crosses) tend to have lower NLG (0.5 or lower) while *PI* users (circles) tend to cover the whole NLG spectrum (0.2 to 0.73). Our next analysis will provide objective support for this observation.

5.3 Performance improvements

The simplest way to investigate the effect of the new PopUp–Incorrect strategy is to compare the two systems in terms of performance (i.e. learning). Table 3 shows in the second column the learning (NLG) in each condition. We find that the new strategy provides a small 0.02 performance improvement (0.48 vs. 0.46), but this effect is far from being significant. A one-way ANOVA test finds no significant effect of the condition on the NLG ($F(1,45)=0.12$, $p<0.73$).

		PI Split	
	All	Low	High
PI	0.48 (0.19)	0.49 (0.21)	0.48 (0.17)
R	0.46 (0.19)	0.56 (0.13)	0.30 (0.18)

Table 3. System performance (NLG) in each condition
(averages and standard deviation in parentheses)

There are several factors that contribute to this lack of significance. First, the new PopUp–Incorrect strategy is only activated by users that have PopUp–Incorrect events. Including users without such events in our comparison could weaken the effect of the new strategy. Second, the impact of the new strategy might depend on how many times it was activated. This relates back to our hypothesis that that a PopUp–Incorrect is an instance of a failed learning opportunity. If this is true and our new PopUp–Incorrect strategy is effective, then we should see a stronger impact on *PI* users with a higher number of PopUp–Incorrect events compared with the similar *R* users.

To test if the impact of the strategy depends on how many times it was engaged, we split users based on their PopUp–Incorrect (**PISplit**) behavior into two subsets: *Low* and *High*. We used the

mean split based on the PopUp–Incorrect relative percentage parameter (see the X axis in Figure 1): users with a parameter value less than 30% go into the Low subset (15 *PI* and 14 *R* users) while the rest go into the High subset (10 *PI* and 8 *R* users).

Results are shown in the third and the fourth columns in Table 3. To test the significance of the effect, we run a two-way factorial ANOVA with NLG as the dependent variable and two factors: PISplit (Low vs. High) and Condition (*PI* vs. *R*). We find a significant effect of the combination PISplit × Condition ($F(1,43)=5.13$, $p<0.03$). This effect and the results of the post-hoc tests are visualized in Figure 2. We find that *PI* users have a similar NLG regardless of their PopUp–Incorrect behavior while for *R*, High PISplit users learn less than Low PISplit users. Posthoc tests indicate that High PISplit *R* users learn significantly less than Low PISplit *R* users ($p<0.01$) and both categories of *PI* users ($p<0.05$). In other words, there is an inherent and significant performance gap between *R* users in the two subsets. The effect of the new PopUp–Incorrect strategy is to bridge this gap and bring High PISplit users to the performance level of the Low PISplit users. This confirms that the new PopUp–Incorrect strategy is effective where it is most needed (i.e. High PISplit users).

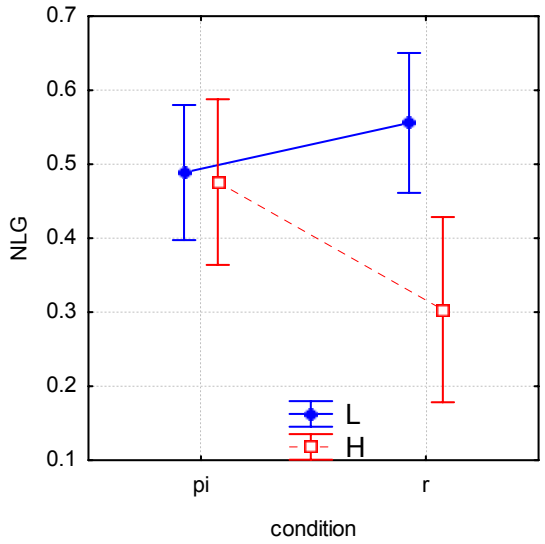


Figure 2. PISplit × Condition effect on NLG (bars represent 95% confidence intervals)

It is interesting to note that Low PISplit *R* users learn better than both categories of *PI* users although the differences are not significant. We hypothesize this happens because not all learning issues are signaled by PopUp–Incorrect events: a user might still have low learning even if he/she does not exhibit any PopUp–Incorrect events. Indeed, there are two *PI* users with a single PopUp–Incorrect event but with very low learning (NLG of 0.00 and 0.14 respectively). It is very likely that other things went wrong for these users rather than the activation of the new PopUp–Incorrect strategy (e.g. they might have other misconceptions that are not addressed by the remediation subdialogues). In fact, removing these two users results in identical NLG averages for the two low PISplit subsets.

5.4 Dialogue duration

We also wanted to know if the new PopUp–Incorrect strategy has an effect on measures of dialogue duration. The strategy delivers additional explanations which can result in an increase in the time users spend with the system (due to reading of the new instruction). Also, when designing tutoring systems researchers strive for learning efficiency: deliver increased learning as fast as possible.

	Total time (min)	No. of sys. turns
PI	44.2 (6.2)	86.4 (6.8)
R	45.5 (5.7)	90.9 (9.3)

Table 4. Dialogue duration metrics (averages and standard deviation in parentheses)

We look at two shallow dialogue metrics: dialogue time and number of turns. Table 4 shows that, in fact, the dialogue duration is shorter for *PI* users on both metrics. A one way ANOVA finds a non-significant effect on dialogue time ($F(1,45)=0.57$, $p<0.45$) but a trend effect for number of system turns ($F(1,45)=3.72$, $p<0.06$). We hypothesize that 2 factors are at play here. First, the additional information activated by the new PopUp–Incorrect strategy might have a positive effect on users' correctness for future system questions especially on questions that discuss similar topics. As a result, the system has to correct the user less and, consequently, finish faster. Second, the average total time *PI* users spend reading the additional information is very small (about 2 minutes) compared to the average dialogue time.

6 Related work

Designing robust, efficient and usable spoken dialogue systems (**SDS**) is a complex process that is still not well understood by the SDS research community (Möller and Ward, 2008). Typically, a number of evaluation/performance

metrics are used to compare multiple (versions of) SDS. But what do these metrics and the resulting comparisons tell us about designing SDS? There are several approaches to answering this question, each requiring a different level of supervision.

One approach that requires little human supervision is to use reinforcement learning. In this approach, the dialogue is modeled as a (partially observable) Markov Decision Process (Levin et al., 2000; Young et al., 2007). A reward is given at the end of the dialogue (i.e. the evaluation metric) and the reinforcement learning process propagates back the reward to learn what the best strategy to employ at each step is. Other semi-automatic approaches include machine learning and decision theoretic approaches (Levin and Pieraccini, 2006; Paek and Horvitz, 2004). However, these semi-automatic approaches are feasible only in small and limited domains though recent work has shown how more complex domains can be modeled (Young et al., 2007).

An approach that works on more complex domains but requires more human effort is through performance analysis: finding and tackling factors that affect the performance (e.g. PARADISE (Walker et al., 2000)). Central to this approach is the quality of the interaction parameters in terms of predicting the performance metric (predictiveness) and informing useful modifications of the system (informativeness). An extensive set of parameters can be found in (Möller, 2005).

Our use of discourse structure for performance analysis extends over previous work in two important aspects. First, we exploit in more detail the hierarchical information in the discourse structure through the domain-independent concept of discourse structure transitions. Most previous work does not use this information (e.g. (Möller, 2005)) or, if used, it is flattened (Walker et al., 2001). Also, to our knowledge, previous work has not employed parameters similar to our transition–phenomena (transition–correctness in this paper) and transition–transition bigram parameters. In addition, several of these parameters are predictive (Rotaru and Litman, 2006).

Second, in our work we also look at the informativeness while most of the previous work stops at the predictiveness step. A notable exception is the work by (Litman and Pan, 2002). The factor they look at is user's having multiple speech recognition problems in the dialogue. This factor is well known in the SDS field and it has been shown to be predictive of system performance by previous work (e.g. (Walker et al., 2000)). To test the informativeness of this factor, Litman and Pan propose a modification of the system in which the initiative and confirmation strategies are changed to more conservative settings whenever the event is detected. Their results show that the modified version leads to improvements in terms of system performance (task completion). We extend over their work by looking at a factor (PopUp–Incorrect) that was not known to be predictive of performance beforehand. We discover this factor through our empirical analyses of existing dialogues and we show that by addressing it (the new PopUp–Incorrect strategy) we also obtain performance improvements (at least for certain users). In addition, we are looking at a performance metric for which significant improvements are harder to obtain with small system changes (e.g. (Graesser et al., 2003)).

7 Conclusions

In this paper we finalize our investigation into the utility of discourse structure for SDS performance analysis (at least for our system). We use the discourse structure transition information in combination with other dialogue phenomena to derive a number of interaction parameters (i.e. transition–phenomena and transition–transition). Our previous work (Rotaru and Litman, 2006) has shown that these parameters are predictive of system performance. Here we take a step further and show that one of these parameters (the PopUp–Incorrect bigram) is also informative. From the interpretation of its predictiveness, we inform a promising modification of our system: offer additional explanations after PopUp–Incorrect events. We implement this modification and we compare it with the original system through a user study. We find that the modification breaks down the negative correlation between PopUp–Incorrect and system performance. In addition, users that need the modification the most (i.e. users with more PopUp–Incorrect events) show significant improvement in performance in the modified system over corresponding users in the original system. However, this improvement is not strong enough to generate significant differences at the population level. Even though the additional explanations add extra time to the dialogue, overall we actually see a small reduction in dialogue duration.

Our work has two main contributions. First, we demonstrate the utility of discourse structure

for performance analysis. In fact, our other work (Rotaru and Litman, 2007) shows that discourse structure is also useful for other SDS tasks. Second, to our knowledge, we are the first to show a complete application of the performance analysis methodology. We discover a new set of predictive interaction parameters in our system and we show how our system can be improved in light of these findings. Consequently, we validate performance analysis as an iterative, "debugging" approach to dialogue design. By analyzing corpora collected with an initial version of the system, we can identify semi-automatically problems in the dialogue design. These problems inform a new version of the system which will be tested for performance improvements. In terms of design methodology for tutoring SDS, our results suggest the following design principle: "do not give up but try other approaches". In our case, we do not give up after a PopUp-Incorrect but give additional explanations.

In the future, we would like to extend our work to other systems and domains. This should be relatively straightforward as the main ingredients, the discourse transitions, are domain independent.

Acknowledgments

This work is supported by the NSF grants 0328431 and 0428472. We would like to thank the ITSPOKE group.

References

D. Bohus. 2007. *Error Awareness and Recovery in Conversational Spoken Language Interfaces*. Ph.D. Dissertation, Carnegie Mellon University, School of Computer Science

A. Graesser, K. Moreno, J. Marineau, A. Adcock, A. Olney and N. Person. 2003. *AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head?* In Proc. of Artificial Intelligence in Education (AIED).

B. Grosz and C. L. Sidner. 1986. Attentions, intentions and the structure of discourse. *Computational Linguistics, 12*(3).

S. Katz, D. Allbritton and J. Connelly. 2003. Going Beyond the Problem Given: How Human Tutors Use Post-Solution Discussions to Support Transfer. *International Journal of Artificial Intelligence in Education (IJAIED), 13*.

E. Levin and R. Pieraccini. 2006. *Value-based optimal decision for dialogue systems.* In Proc. of IEEE/ACL Workshop on Spoken Language Technology (SLT).

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*.

D. Litman and S. Pan. 2002. Designing and Evaluating an Adaptive Spoken Dialogue System. *User Modeling and User-Adapted Interaction, 12*(2/3).

D. Litman, C. Rose, K. Forbes-Riley, K. VanLehn, D. Bhembe and S. Silliman. 2006. Spoken Versus Typed Human and Computer Dialogue Tutoring. *International Journal of Artificial Intelligence in Education, 16*.

S. Möller. 2005. *Parameters for Quantifying the Interaction with Spoken Dialogue Telephone Services.* In Proc. of SIGDial.

S. Möller and N. Ward. 2008. *A Framework for Model-based Evaluation of Spoken Dialog Systems.* In Proc. of Workshop on Discourse and Dialogue (SIGDial).

T. Paek and E. Horvitz. 2004. *Optimizing Automated Call Routing by Integrating Spoken Dialog Models with Queuing Models.* In Proc. of HLT-NAACL.

A. Raux and M. Eskenazi. 2008. *Optimizing Endpointing Thresholds using Dialogue Features in a Spoken Dialogue System.* In Proc. of 9th SIGdial Workshop on Discourse and Dialogue.

M. Rotaru. 2008. *Applications of Discourse Structure for Spoken Dialogue Systems*. Ph.D. Dissertation, University of Pittsburgh, Department of Computer Science

M. Rotaru and D. Litman. 2006. *Exploiting Discourse Structure for Spoken Dialogue Performance Analysis.* In Proc. of EMNLP.

M. Rotaru and D. Litman. 2007. *The Utility of a Graphical Representation of Discourse Structure in Spoken Dialogue Systems.* In Proc. of ACL.

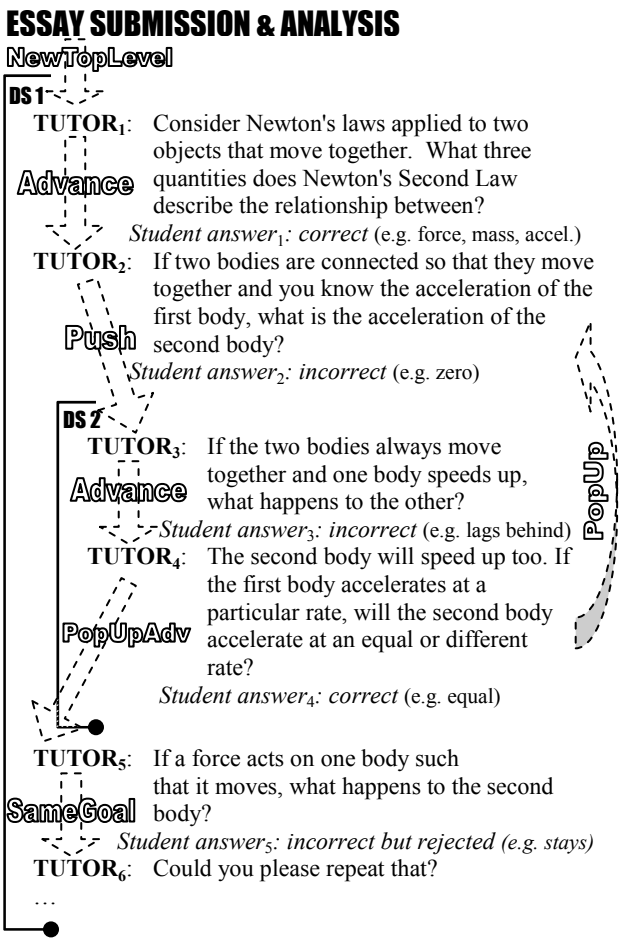
S. Singh, D. Litman, M. Kearns and M. Walker. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. *Journal of Artificial Intelligence Research,* (16).

K. VanLehn, A. C. Graesser, G. T. Jackson, P. Jordan, A. Olney and C. P. Rose. 2007. When are tutorial dialogues more effective than reading? *Cognitive Science, 31*(1).

M. Walker, D. Litman, C. Kamm and A. Abella. 2000. Towards Developing General Models of Usability with PARADISE. *Natural Language Engineering*.

M. Walker, R. Passonneau and J. Boland. 2001. *Quantitative and Qualitative Evaluation of Darpa Communicator Spoken Dialogue Systems.* In Proc. of ACL.

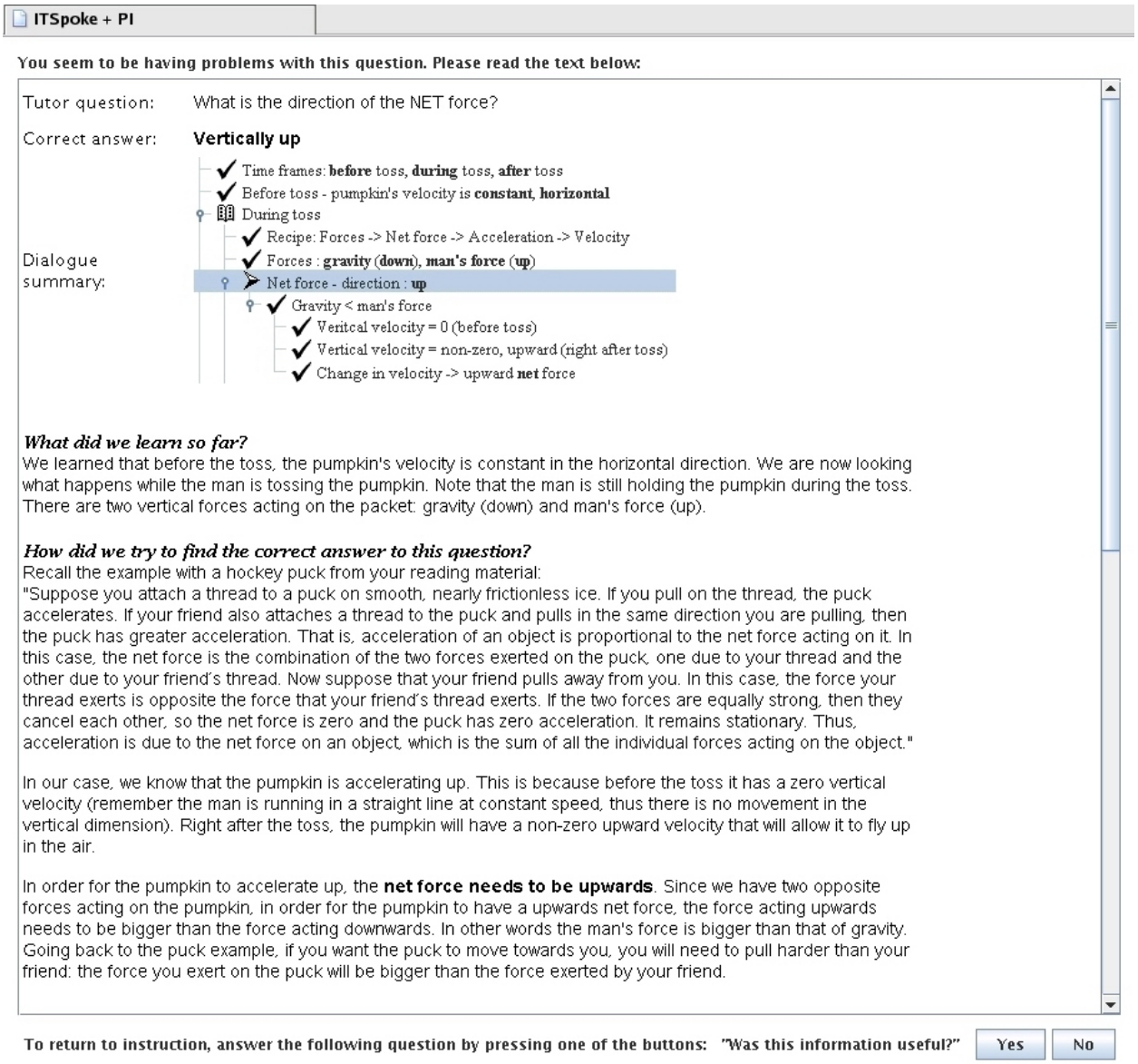
S. Young, J. Schatzmann, K. Weilhammer and H. Ye. 2007. *The Hidden Information State Approach to Dialog Management.* In Proc. of ICASSP.



Appendix 1. Automatic annotation of discourse structure hierarchy and of discourse structure transitions

Discourse structure hierarchy annotation: DS1 is the top level discourse segment. Its purpose is to correct misconceptions in user's essay and/or to elicit more complete explanations for the essay. DS2 is an embedded discourse segment which corresponds to the remediation subdialogue for question $Tutor_2$.

Discourse structure transition annotation: Each transition labels the system turn at the tip of the arrow (e.g. $Tutor_2$ is labeled with Advance). Please note that $Tutor_2$ will not be labeled with PopUp because, in such cases, an extra system turn will be created between Tutor4 and Tutor5 with the same content as Tutor2. This extra turn also includes variations of "Ok, back to the original question" to mark the discourse segment boundary transition.



Appendix 2. Sample additional instructions webpage

Problem discussed by ITSPOKE: Suppose a man is running in a straight line at constant speed. He throws a pumpkin straight up. Where will it land? Explain.

Location in the dialogue: For this problem, ITSPOKE discusses what happens during three time frames: before pumpkin toss, during pumpkin toss and after pumpkin toss. ITSPOKE is currently discussing the forces and the net force on the pumpkin during the toss.

The Role of Interactivity in Human-Machine Conversation for Automatic Word Acquisition

Shaolin Qu **Joyce Y. Chai**
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
{qushaoli,jchai}@cse.msu.edu

Abstract

Motivated by the psycholinguistic finding that human eye gaze is tightly linked to speech production, previous work has applied naturally occurring eye gaze for automatic vocabulary acquisition. However, unlike in the typical settings for psycholinguistic studies, eye gaze can serve different functions in human-machine conversation. Some gaze streams do not link to the content of the spoken utterances and thus can be potentially detrimental to word acquisition. To address this problem, this paper investigates the incorporation of interactivity in identifying the close coupling of speech and gaze streams for word acquisition. Our empirical results indicate that automatic identification of closely coupled gaze-speech streams leads to significantly better word acquisition performance.

1 Introduction

Spoken conversational interfaces have become increasingly important in many applications such as remote interaction with robots (Lemon et al., 2002), intelligent space station control (Aist et al., 2003), and automated training and education (Razzaq and Heffernan, 2004). As in any conversational system, one major bottleneck in conversational interfaces is robust language interpretation. To address this problem, previous multimodal conversational systems have utilized pen-based or deictic gestures (Bangalore and Johnston, 2004; Qu and Chai, 2006) to improve interpretation. Besides gestures, eye movements that naturally occur during interaction provide another important channel for language understanding, for example, reference resolution (Byron et al., 2005; Prasov and Chai, 2008). Recent work has also shown that what users look at on the interface (e.g., natural scenes or generated graphic displays) during speech production provides unique opportunities for word acquisition, namely automatically acquiring semantic meanings of spoken words by grounding them to visual entities (Liu et al., 2007) or domain concepts (Qu and Chai, 2008).

Psycholinguistic studies have shown that eye gaze indicates a person's attention (Just and Carpenter, 1976), and eye movement can facilitate spoken language comprehension (Tanenhaus et al., 1995; Eberhard et al., 1995). It has been found that users' eyes move to the mentioned object directly before speaking a word (Meyer et al., 1998; Rayner, 1998; Griffin and Bock, 2000). This parallel behavior of eye gaze and speech production motivates our previous work on word acquisition (Liu et al., 2007; Qu and Chai, 2008). However, in interactive conversation, human gaze behavior is much more complex than in the typical controlled settings used in psycholinguistic studies. There are different types of eye movements (Kahneman, 1973). The naturally occurring eye gaze during speech production may serve different functions, for example, to engage in the conversation or to manage turn taking (Nakano et al., 2003). Furthermore, while interacting with a graphic display, a user could be talking about objects that were previously seen on the display or something completely unrelated to any object the user is looking at. Therefore using every speech-gaze pair for word acquisition can be detrimental. The type of gaze that is mostly useful for word acquisition is the kind that reflects the underlying attention and tightly links to the content of the co-occurring speech. Thus, one important question is how to identify the closely coupled speech and gaze streams to improve word acquisition.

To address this question, we develop an approach that incorporates interactivity (e.g., speech,

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 188–195,
Queen Mary University of London, September 2009.

user activity, conversation context) with eye gaze to identify closely coupled speech and gaze streams. We further use the identified speech and gaze streams to acquire words with a translation model. Our empirical evaluation demonstrates that automatic identification of closely coupled gaze-speech streams can lead to significantly better word acquisition performance.

2 Related Work

Previous work has explored word acquisition by grounding words to visual entities. In (Roy and Pentland, 2002), given speech paired with video images of objects, mutual information between auditory and visual signals was used to acquire words by associating acoustic phone sequences with the visual prototypes (e.g., color, size, shape) of objects. Given parallel pictures and description texts, generative models were used to acquire words by associating words with image regions in (Barnard et al., 2003). Different from this previous work, in our work, the visual attention foci accompanying speech are indicated by eye gaze. As an implicit and subconscious input, eye gaze brings additional challenges in word acquisition.

Eye gaze has been explored for word acquisition in previous work. In (Yu and Ballard, 2004), given speech paired with eye gaze and video images, a translation model was used to acquire words by associating acoustic phone sequences with visual representations of objects and actions. Word acquisition from transcribed speech and eye gaze during human-machine conversation has been investigated recently. In (Liu et al., 2007), a translation model was developed to associate words with visual objects on a graphical display. In our previous work (Qu and Chai, 2008), enhanced translation models incorporating speech-gaze temporal information and domain knowledge were developed to improve word acquisition. However, none of these previous works has investigated the role of interactivity in word acquisition, which is the focus of this paper.

3 Data Collection

We collected speech and eye gaze data through user studies. This data set is different from the data set used in our previous work (Qu and Chai, 2008). The difference lies in two aspects: 1) the data for this investigation was collected during mixed initiative human-machine conversation whereas the data in (Qu and Chai, 2008) was based only on question and answering; 2) user studies were conducted in a more complex domain for this investigation, which resulted in a richer data set that contains a larger vocabulary.

3.1 Domain

Figure 1: Treasure hunting domain

Figure 1 shows the 3D treasure hunting domain used in our work. In this application, the user needs to consult with a remote "expert" (i.e., an artificial system) to find hidden treasures in a castle with 115 3D objects. The expert has some knowledge about the treasures but can not see the castle. The user has to talk to the expert for advice regarding finding the treasures. The application is developed based on a game engine and provides an immersive environment for the user to navigate in the 3D space. During the experiment, each user's speech was recorded, and the user's eye gaze was captured by a Tobii eye tracker.

3.2 Data Preprocessing

From 20 users' experiments, we collected 3709 utterances with accompanying gaze fixations. We transcribed the collected speech. The vocabulary size of the speech transcript is 1082, among which 227 are either nouns or adjectives. The user's speech was also automatically recognized online by the Microsoft speech recognizer with a word error rate (WER) of 48.1% for the 1-best recognition. The vocabulary size of the 1-best speech recognition is 3041, among which 1643 are either nouns or adjectives.

The collected speech and gaze streams were automatically paired together by the system. Each time the system detected a sentence boundary (indicated by a long pause of 500 milliseconds) of the user's speech, it paired the recognized speech with the gaze fixations that the system had been accumulating since the previously detected sentence

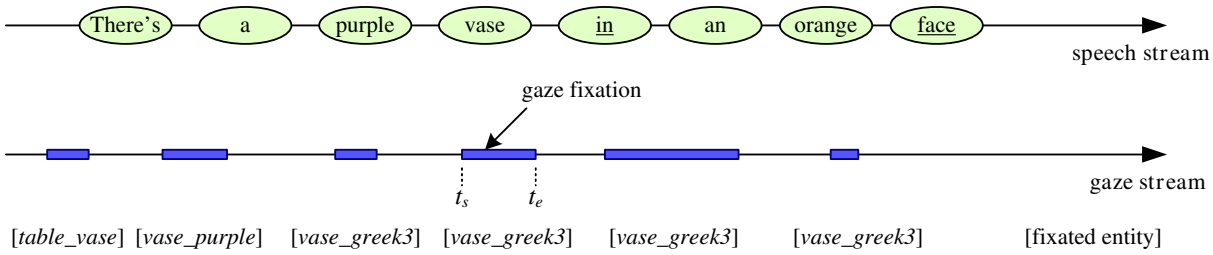


Figure 2: Accompanying gaze fixations and the 1-best recognition of a user's utterance "There's a purple vase and an orange vase." (There are two incorrectly recognized words "in" and "face" in the 1-best recognition)

boundary. Figure 2 shows a pair of user speech and accompanying stream of gaze fixations. In the speech stream, each spoken word was timestamped by the speech recognizer. In the gaze stream, each gaze fixation has a starting timestamp t_s and an ending timestamp t_e provided by the eye tracker. Each gaze fixation results in a fixated entity (3D object). When multiple entities are fixated by one gaze fixation due to the overlapping of entities, the one in the forefront is chosen.

Given the paired speech and gaze streams, we build a set of parallel word sequence and gaze fixated entity sequence $\{(\mathbf{w}, \mathbf{e})\}$ for the task of word acquisition. In section 6, we will evaluate word acquisition in two settings: 1) word sequence $\mathbf{w}$ contains all of the nouns/adjectives in the speech transcript, and 2) $\mathbf{w}$ contains all of the recognized nouns/adjectives in the 1-best speech recognition.

4 Word Acquisition With Eye Gaze

The task of word acquisition in our application is to ground words to the visual entities. Specifically, given the parallel word and entity sequences $\{(\mathbf{w}, \mathbf{e})\}$, we want to find the best match between the words and the entities. Following our previous work (Qu and Chai, 2008), we formulate word acquisition as a translation problem and use translation models for word acquisition. For each entity e, we first estimate the word-entity association probability $p(w|e)$ with a translation model, then choose the words with the highest probabilities as acquired words for e.

Inspired by the psycholinguistic findings that users' eyes move to the mentioned object before speaking a word (Meyer et al., 1998; Rayner, 1998; Griffin and Bock, 2000), in our previous work (Qu and Chai, 2008), we have incorporated the gaze-speech temporal information in the translation model as follows (referred as Model-2t through the rest of this paper):

$$p(\mathbf{w}|\mathbf{e}) = \prod_{j=1}^{m} \sum_{i=0}^{l} p_t(a_j = i|j, \mathbf{e}, \mathbf{w}) p(w_j|e_i)$$

where l and m are the lengths of entity and word sequences respectively. In this equation, $p_t(a_j = i|j, \mathbf{e}, \mathbf{w})$ is the temporal alignment probability representing the probability that w_j is aligned with e_i, which is further defined by:

$$p_t(a_j = i|j, \mathbf{e}, \mathbf{w}) = \begin{cases} 0 & d(e_i, w_j) > 0 \\ \frac{\exp[\alpha \cdot d(e_i, w_j)]}{\sum_i \exp[\alpha \cdot d(e_i, w_j)]} & d(e_i, w_j) \leq 0 \end{cases}$$

where α is a scaling factor, and $d(e_i, w_j)$ is the temporal distance between e_i and w_j. Based on the psycholinguistic finding that eye gaze happens before a spoken word, w_j is not allowed to be aligned with e_i when w_j happens earlier than e_i (i.e., $d(e_i, w_j) > 0$). When w_j happens no earlier than e_i (i.e., $d(e_i, w_j) \leq 0$), the closer they are, the more likely they are aligned. An EM algorithm is used to estimate $p(w|e)$ and α in the model.

Our evaluation in (Qu and Chai, 2008) has shown that Model-2t that incorporates temporal alignment between speech and eye gaze achieves significantly better word acquisition performance compared to the model where no temporal alignment is introduced. Therefore, this model is used for the investigation in this paper.

5 Identification of Closely Coupled Gaze-Speech Pairs

Successful word acquisition with the translation models relies on the tight coupling between the gaze fixations and the speech content. As mentioned earlier, not all gaze-speech pairs have this tight coupling. In a gaze-speech pair, if the speech

does not have any word that relates to any of the gaze fixated entities, this instance only adds noise to word acquisition. Therefore, we should identify the closely coupled gaze-speech pairs and only use them for word acquisition.

In this section, we first describe the feature extraction, then evaluate the application of a logistic regression classifier to predict whether a gaze-speech pair is a **closely coupled gaze-speech instance** – an instance where at least one noun or adjective in the speech stream describes some entity fixated by the gaze stream. For the training of the classifier, we manually labeled each instance as either a coupled instance or not based on the speech transcript and the gaze fixations.

5.1 Feature Extraction

For a gaze-speech instance, the following sets of features are automatically extracted.

5.1.1 Speech Features (S)

The following features are extracted from speech:

- c_w – count of nouns and adjectives.
  More nouns and adjectives are expected in the user's utterance describing entities.
- c_w/l_s – normalized noun/adjective count.
  The effect of speech length l_s on c_w is considered.

5.1.2 Gaze Features (G)

For each fixated entity e_i, let l_e^i be its temporal fixation length. Note that several gaze fixations may have the same fixated entity, l_e^i is the total length of all the gaze fixations that fixate on entity e_i. We extract the following features from gaze stream:

- c_e – count of different gaze fixated entities.
  Fewer fixated entities are expected when the user is describing entities while looking at them.
- c_e/l_s – normalized entity count.
  The effect of temporal spoken utterance length l_s on c_e is considered.
- $\max_i(l_e^i)$ – maximal fixation length.
  At least one fixated entity's fixation is expected to be long enough when the user is describing entities while looking at them.
- $mean(l_e^i)$ – average fixation length.
  The average gaze fixation length is expected to be longer when the user is describing entities while looking at them.
- $var(l_e^i)$ – variance of fixation lengths.
  The variance of the fixation lengths is expected to be smaller when the user is describing entities while looking at them.

The number of gaze fixated entities is not only determined by the user's eye gaze, but also affected by the visual scene. Let c_e^s be the count of all the entities that have been visible during the time period concurrent with the gaze stream. We also extract the following scene related feature:

- c_e/c_e^s – scene-normalized fixated entity count.
  The effect of the visual scene on c_e is considered.

5.1.3 User Activity Features (UA)

While interacting with the system, the user's activity can also be helpful in determining whether the user's eye gaze is tightly linked to the content of the speech. The following features are extracted from the user's activities:

- *maximal distance of the user's movements* – the maximal change of user position (3D coordinates) during speech.
  The user is expected to move within a smaller range while looking at entities and describing them.
- *variance of the user's positions*
  The user is expected to move less frequently while looking at entities and describing them.

5.1.4 Conversation Context Features (CC)

While talking to the system (i.e., the "expert"), the user's language and gaze behavior are influenced by the state of the conversation. For each gaze-speech instance, we use the previous system response type as a nominal feature to predict whether this is a closely coupled gaze-speech instance.

In our treasure hunting domain, there are 8 types of system responses in 2 categories:

System Initiative Responses:

- *specific-see* – the system asks whether the user sees a certain entity, e.g., "Do you see another couch?".
- *nonspecific-see* – the system asks whether the user sees anything, e.g., "Do you see anything else?", "Tell me what you see".

- *previous-see* – the system asks whether the user has previously seen something, e.g., "Have you previously seen a similar object?".
- *describe* – the system asks the user to describe in detail what the user sees, e.g., "Describe it", "Tell me more about it".
- *compare* – the system asks the user to compare what the user sees, e.g., "Compare these objects".
- *repair-request* – the system asks the user to make clarification, e.g., "I did not understand that", "Please repeat that".
- *action-request* – the system asks the user to take action, e.g., "Go back", "Try moving it".

User Initiative Responses:

- *misc* – the system hands the initiative back to the user without specifying further requirements, e.g., "I don't know", "Yes".

5.2 Evaluation of Gaze-Speech Identification

Given the extracted features and the "closely coupled" label of each instance in the training set, we train a logistic regression classifier (le Cessie and van Houwelingen, 1992) to predict whether an instance is a closely coupled gaze-speech instance.

Since the goal of identifying closely coupled gaze-speech instances is to improve word acquisition and we are only interested in acquiring nouns and adjectives, only the instances with recognized nouns/adjectives are used for training the logistic regression classifier. Among the 2969 instances with recognized nouns/adjectives and gaze fixations, 2002 (67.4%) instances are labeled as "closely coupled". The prediction is evaluated by a 10-fold cross validation.

Feature sets	Precision	Recall
Null (*baseline*)	0.674	1
S	0.686	0.995
G	0.707	0.958
UA	0.704	0.942
CC	0.688	0.936
G + UA	0.719	0.948
G + UA + S	0.741	0.908
G + UA + CC	0.731	0.918
G + UA + CC + S	**0.748**	0.899

Table 1: Gaze-speech prediction performance for the instances with 1-best speech recognition

Table 1 shows the prediction precision and recall when different sets of features are used. As seen in the table, as more features are used, the prediction precision goes up and the recall goes down. It is important to note that prediction precision is more critical than recall for word acquisition when sufficient amount data is available. *Noisy* instances where the gaze is not coupled with the speech content will only hurt word acquisition since they will guide the translation models to ground words to the wrong entities. Although higher recall can be helpful, its effect is expected to be reduced when more data becomes available.

The results show that speech features (S) and conversation context features (CC), when used alone, do not improve prediction precision much compared to the baseline of predicting all instances as closely coupled (with a precision of 67.4%). When used alone, gaze features (G) and user activity features (UA) are the two most useful feature sets for increasing prediction precision. When they are used together, the prediction precision is further increased. Adding either speech features or conversation context features to gaze and user activity features (G + UA + S/CC) further increases the prediction precision. Using all features (G + UA + CC + S) achieves the highest prediction precision, which is significantly better than the baseline: $z = 5.93, p < 0.001$. Therefore, we choose to use all feature sets to identify the closely coupled gaze-speech instances for word acquisition.

To compare the effects of the automatic gaze-speech identification on word acquisition from various speech input (1-best speech recognition, speech transcript), we also use the logistic regression classifier with all feature sets to identify the closely coupled gaze-speech instances for the instances with speech transcript. For the instances with speech transcript, there are 2948 instances with nouns/adjectives and gaze fixations, 2128 (72.2%) of them being labeled as "closely coupled". The prediction precision is 77.9% and the recall is 93.8%. The prediction precision is significantly better than the baseline of predicting all instances as coupled: $z = 4.92, p < 0.001$.

6 Evaluation of Word Acquisition

Every conversational system has an initial vocabulary where words are associated with domain concepts of entities. In our evaluation, we assume that

the system's vocabulary has one default word for each entity that indicates the semantic type of the entity. For example, the word "barrel" is the default word for the entity *barrel*. For each entity, we only evaluate those new words that are not in the system's vocabulary.

The acquired words are evaluated against the "gold standard" words that were manually compiled for each entity and its properties based on all users' speech transcripts. For the 115 entities in our domain, each entity has 1 to 20 "gold standard" words. The average number of "gold standard" words for an entity is 6.7.

6.1 Evaluation Metrics

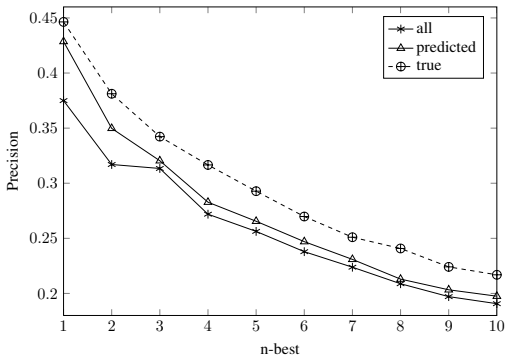
We evaluate the n-best acquired words (words grounded to domain concepts of entities) using precision, recall, and F-measure. When a different n is chosen, we will have different precision, recall, and F-measure.

We also evaluate the whole ranked candidate word list on Mean Reciprocal Rank Rate (MRRR) as in (Qu and Chai, 2008):

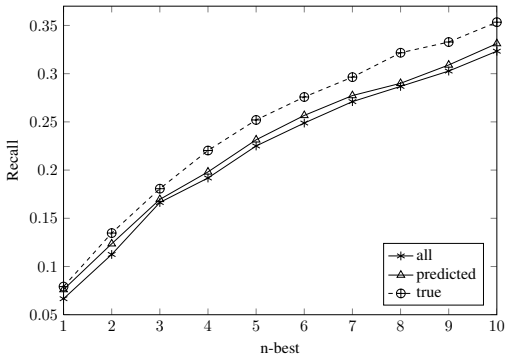
$$\text{MRRR} = \frac{\sum_e \frac{\sum_{i=1}^{N_e} 1/index(w_e^i)}{\sum_{i=1}^{N_e} 1/i}}{\#e}$$

where N_e is the number of all "gold standard" words $\{w_e^i\}$ for entity e, $index(w_e^i)$ is the index of word w_e^i in the ranked list of candidate words for entity e.

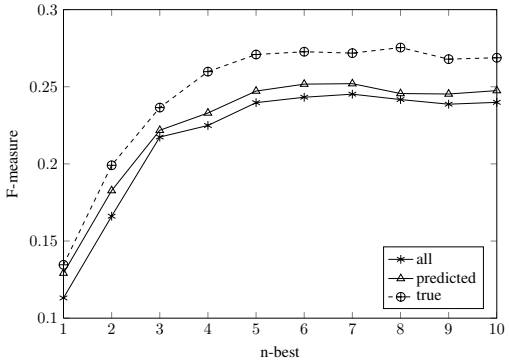
MRRR measures how close the ranks of the "gold standard" words in the candidate word lists are to the best-case scenario where the top N_e words are the "gold standard" words for e. The higher the MRRR, the better is the acquisition performance.



(a) precision



(b) recall



(c) F-measure

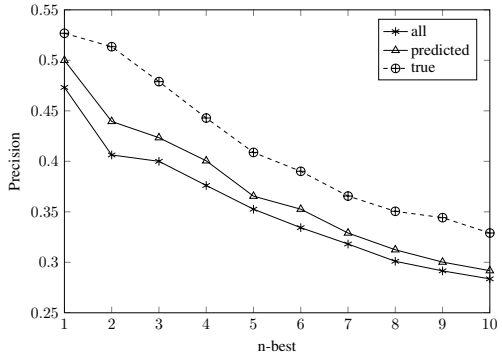
Figure 3: Performance of word acquisition on 1-best speech recognition

6.2 Evaluation Results

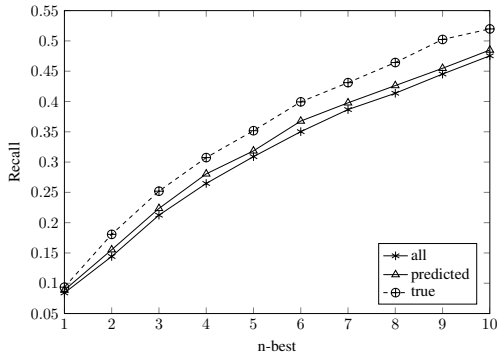
We evaluate the effect of the closely coupled gaze-speech instances on word acquisition from the 1-best speech recognition and speech transcript. The predicted closely coupled gaze-speech instances are generated by a 10-fold cross validation with the logistic regression classifier.

Figure 3 shows the precision, recall, and F-measure of the n-best words acquired from 1-best speech recognition by Model-2t using all instances (*all*), predicted coupled instances (*predicted*), and true (manually labeled) coupled instances (*true*). As shown in the figure, using predicted coupled instances achieves consistently better performance than using all instances. These results show that the identification of coupled gaze-speech prediction helps word acquisition. When the true coupled instances are used, the performance is further improved. This means that reliable identification of coupled gaze-speech instances can lead to better word acquisition.

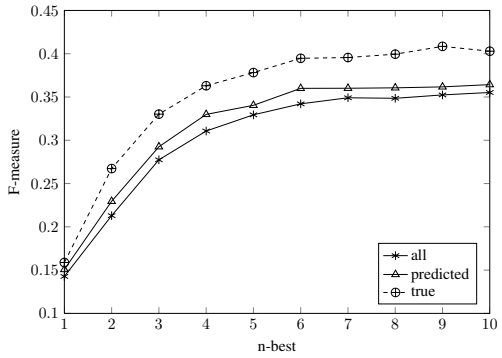
Figure 4 shows the precision, recall, and F-measure of the n-best words acquired from speech transcript by Model-2t using all instances, predicted coupled instances, and true coupled instances. Consistent with the performance based on the 1-best speech recognition, we can observe



(a) precision



(b) recall



(c) F-measure

Figure 4: Performance of word acquisition on speech transcript

that automatic identification of coupled instances results in better word acquisition performance and using the true coupled instances results in even better performance.

Table 2 presents the MRRRs achieved by Model-2t when words are acquired from different speech input (speech transcript, 1-best recognition) with different set of instances (all instances, predicted coupled instances, true coupled instances). These results also show the consistent behavior. Using predicted coupled instances achieves significantly better MRRR than using all instances no matter the words are acquired from 1-best speech recognition ($t = 2.59, p < 0.006$) or speech transcript($t = 3.15, p < 0.002$). When the true coupled instances are used, the performances are further improved for both 1-best recognition ($t = 2.29, p < 0.013$) and speech transcript ($t = 5.21, p < 0.001$) compared to using predicted coupled instances.

Instances	All	Predicted	True
Transcript	0.462	0.480	0.526
1-best reco	0.343	0.369	0.390

Table 2: MRRRs based on different data set

The quality of speech recognition is critical to word acquisition performance. Comparing word acquisition based on speech transcript and 1-best speech recognition, as expected, word acquisition performance on speech transcript is much better than on recognized speech. However, the acquisition performance based on speech transcript is still comparably low. For example, the recall of acquired words is still below 55% even when the 10 best word candidates are acquired for each entity. This is mainly due to the scarcity of words. Many words appear less than three times in the data, which makes them unlikely to be associated with any entity by the translation model. When more data is available, we expect to see better acquisition performance.

Note that our current evaluation is based on a two-stage approach, i.e., first identifying closely-coupled streams based on supervised classification and then automatically establishing mappings between words and entities in an unsupervised manner. There could be other approaches to address the word acquisition problem (e.g., supervised learning to directly identify whether a word is mapped to an object). Our two-stage approach has the advantage of requiring minimum supervision since the models learned from the first stage is application-independent and is potentially portable to different domains.

7 Conclusions

Unlike in the typical settings for psycholinguistic studies, human eye gaze can serve different functions during human machine conversation. Some gaze and speech streams may not be tightly coupled and thus can be detrimental to word acquisition. Therefore, this paper describes an approach that incorporates features from the interac-

tion context to identify closely coupled gaze and speech streams. Our empirical results indicate that the word acquisition based on these automatically identified gaze-speech streams achieves significantly better performance than the word acquisition based on all gaze-speech streams. Our future work will combine gaze-based word acquisition with multiple speech recognition hypotheses (e.g., word lattices) to further improve word acquisition and language interpretation performance.

Acknowledgments

This work was supported by grants IIS-0347548 and IIS-0535112 from the National Science Foundation. We thank anonymous reviewers for their valuable comments and suggestions.

References

G. Aist, J. Dowding, B. A. Hockey, M. Rayner, J. Hieronymus, D. Bohus, B. Boven, N. Blaylock, E. Campana, S. Early, G. Gorrell, and S. Phan. 2003. Talking through procedures: An intelligent space station procedure assistant. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

S. Bangalore and M. Johnston. 2004. Robust multimodal understanding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan. 2003. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135.

D. Byron, T. Mampilly, V. Sharma, and T. Xu. 2005. Utilizing visual attention for cross-modal coreference interpretation. In *Proceedings of the Fifth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT-05)*, pages 83–96.

K. Eberhard, M. Spivey-Knowiton, J. Sedivy, and M. Tanenhaus. 1995. Eye movements as a window into real-time spoken language comprehension in natural contexts. *Journal of Psycholinguistic Research*, 24:409–436.

Z. Griffin and K. Bock. 2000. What the eyes say about speaking. *Psychological Science*, 11:274–279.

M. Just and P. Carpenter. 1976. Eye fixations and cognitive processes. *Cognitive Psychology*, 8:441–480.

D. Kahneman. 1973. *Attention and Effort*. Prentice-Hall, Inc., Englewood Cliffs.

S. le Cessie and J. van Houwelingen. 1992. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201.

O. Lemon, A. Gruenstein, and S. Peters. 2002. Collaborative activities and multitasking in dialogue systems. *Traitement Automatique des Langues*, 43(2):131–154.

Y. Liu, J. Chai, and R. Jin. 2007. Automated vocabulary acquisition and interpretation in multimodal conversational systems. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*.

A. Meyer, A. Sleiderink, and W. Levelt. 1998. Viewing and naming objects: eye movements during noun phrase production. *Cognition*, 66(22):25–33.

Y. Nakano, G. Reinstein, T. Stocky, and J. Cassell. 2003. Towards a model of face-to-face grounding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Z. Prasov and J. Chai. 2008. What's in a gaze? the role of eye-gaze in reference resolution in multimodal conversational interfaces. In *Proceedings of ACM 12th International Conference on Intelligent User interfaces (IUI)*.

S. Qu and J. Chai. 2006. Salience modeling based on non-verbal modalities for spoken language understanding. In *Proceedings of the International Conference on Multimodal Interfaces (ICMI)*, pages 193–200.

S. Qu and J. Chai. 2008. Incorporating temporal and semantic information with eye gaze for automatic word acquisition in multimodal conversational systems. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 244–253.

K. Rayner. 1998. Eye movements in reading and information processing - 20 years of research. *Psychological Bulletin*, 124(3):372–422.

L. Razzaq and N. Heffernan. 2004. Tutorial dialog in an equation solving intelligent tutoring system. In *Proceedings of the Workshop on Dialog-based Intelligent Tutoring Systems: State of the art and new research directions*.

D. Roy and A. Pentland. 2002. Learning words from sights and sounds, a computational model. *Cognitive Science*, 26(1):113–146.

M. Tanenhaus, M. Spivey-Knowiton, K. Eberhard, and J. Sedivy. 1995. Integration of visual and linguistic information in spoken language comprehension. *Science*, 268:1632–1634.

C. Yu and D. Ballard. 2004. A multimodal learning interface for grounding spoken language in sensory perceptions. *ACM Transactions on Applied Perceptions*, 1(1):57–80.

Clarification Potential of Instructions

Luciana Benotti
TALARIS Team - LORIA (Université Henri Poincaré, INRIA)
BP 239, 54506 Vandoeuvre-lès-Nancy, France
Luciana.Benotti@loria.fr

Abstract

Our hypothesis is that conversational implicatures are a rich source of clarification questions. In this paper we do two things. First, we motivate the hypothesis in theoretical, practical and empirical terms. Second, we present a framework for generating the clarification potential of an instruction by inferring its conversational implicatures with respect to a particular context. General means-ends inference, beyond classical planning, turns out to be crucial.

1 Introduction

Practical interest in clarification requests (CRs) no longer needs to be awakened in dialogue system designers (Gabsdil, 2003; Purver, 2004; Rodríguez and Schlangen, 2004; Rieser and Moore, 2005; Skantze, 2007). In sociolinguistics and discourse analysis, repair has been an even more favored theme for almost three decades now; see (Schegloff, 1987) as a representative example. However, the theoretical scope of the phenomena and its implications for a theory of meaning are still being delineated. Recently, it has been proposed that clarification should be a basic component in an adequate theory of meaning:

*The basic criterion for adequacy of a theory of meaning is the ability to characterize for any utterance type the **update** that emerges in the aftermath of successful mutual understanding and the full range of **possible clarification requests** otherwise — this is the early 21st century analogue of truth conditions. (Ginzburg, 2009, p.4)*

In this view, repairs are not a necessary evil but an intrinsic mechanism of language. In fact, interpreting an utterance centrally involves characterizing **the space of possible requests of clarification** of the utterance, that is its **clarification potential**. We believe that Ginzburg's comment points in the right direction; we discuss the motivations from a theoretical perspective in Section 2.1. In Section 2.2 we review a state-of-the-art definition of the notion of clarification from the perspective of dialogue system designers. This review makes evident the necessity of further refining the notion of clarification if it is going to play such a central role in a theory of meaning. In Section 2.3 we present our findings in the corpus SCARE (Stoia et al., 2008) which empirically motivates our work.

We believe that it is crucial to redefine the notion of clarification in functional terms. Because we know that the task is difficult, we restrict ourselves to one utterance type, **instructions**, and to a particular interaction level, the **task-level**. In the rest of the paper (Sections 3 and 4), we present a framework that generates the task-level clarification potential of an instruction by inferring its particularized conversational implicatures.

The following exchange illustrate the kinds of interactions our framework models:

(1) A(1): Turn it on.
B(2): By pushing the red button?
(Rodríguez and Schlangen, 2004, p.102)

Roughly speaking, our framework takes as input sentences like A(1) and explains how B(2) can be generated. In particular, the framework indicates what kinds of information resources and what kind of inferences are involved in the process of generating utterances like B(2). In other words, the goal of the framework is to explain why A(1) and B(2) constitute a coherent dialogue by saying how B(2) is relevant to A(1).

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 196–205,
Queen Mary University of London, September 2009.

2 Background and motivation

In this section, we motivate our framework from the **theoretical perspective** of pragmaticists interested in the relevance of clarifications for a theory of meaning, from the **practical perspective** of dialogue system designers, and from the **empirical perspective** of a human-human corpus that provides evidence for the necessity of such a framework.

2.1 Theoretical: Relevance of clarifications

Modeling how listeners draw inferences from what they hear, is a basic problem for theories of understanding natural language. An important part of the information conveyed is inferred in context, given the nature of conversation as a goal-oriented enterprise; as illustrated by the following classical example by Grice:

(2) A: I am out of petrol.
B: There is a garage around the corner.
$\rightsquigarrow$ B thinks that the garage is open.
(Grice, 1975, p.311)

B's answer *conversationally implicates* ($\rightsquigarrow$) information that is relevant to A. In Grice's terms, B made a relevance implicature: he would be flouting the conversational maxim of relevance unless he believes that it's possible that the garage is open. A conversational implicature (CI) is different from an entailment in that it is *cancelable* without contradiction. B can append material that is inconsistent with the CI — "but I don't know whether it's open". Since the CI can be canceled, B knows that it does not necessarily hold and then both B or A are able to *reinforce* or *clarify* it without repetition.

It is often controversial whether something is actually a CI or not (people have different intuitions, which is not surprising given that people have different background assumptions). In dialogue, CRs provide good evidence of the implicatures that have been made simply because they make implicatures explicit. Take for instance the clarification request which can naturally follow Grice's example.

(3) A: and you think it's open?

B will have to answer and support the implicature (for instance with "yes, it's open till midnight") if he wants to get it added to the common ground; otherwise, if he didn't mean it, he can well reject it without contradiction with "well, you have a point there, they might have closed".

Our hypothesis is that CIs are a rich source of clarification requests. And our method for generating the potential CRs of an utterance will be then to infer (some of) the CIs of that utterance with respect to a particular context.

2.2 Practical: Kinds of clarifications

Giving a precise definition of a clarification request is more difficult than might be thought at first sight. Rodríguez and Schlangen (2004) recognize this problem by saying:

Where we cannot report reliability yet is for the task of identifying CRs in the first place. This is not a trivial problem, which we will address in future work. As far as we can see, Purver, Ginzburg and Healey have not tested for reliability for doing this task either. (Rodríguez and Schlangen, 2004, p.107)

One of the most developed classifications of CRs is the one presented in (Purver, 2004). However, Purver's classification relies mainly on the surface form of the CRs. The attempts found in the literature to give a classification of CRs according to their functions (Rodríguez and Schlangen, 2004; Rieser and Moore, 2005) are based on the four-level model of communication independently developed by Clark (1996) and Allwood (1995). The model is summarized in Figure 1 (from the point of view of the hearer).

Level	Clark	Allwood
4	consideration	reaction
3	understanding	understanding
2	identification	perception
1	attention	contact

Figure 1: The four levels of communication

Most of the previous work on clarifications has concentrated on levels 1 to 3 of communication. For instance, Schlangen (2004) proposed a fined-grained classification of CRs but only for level 3. Gabsdil (2003) proposes a test for identifying CRs. The test says that CRs cannot be preceded by explicit acknowledgements. But in the following example, presented by Gabsdil himself, the CR uttered by F can well start with an explicit "ok".

(4) G: I want you to go up the left hand side of it towards the green bay and make it a slightly diagonal line, towards, sloping to the right.
F: So you want me to go above the carpenter? (Gabsdil, 2003, p.30)

The kind of CR showed in 4, also called **clarification of intentions** or **task level clarifications**, are in fact very frequent in dialogue; they have been reported to be the second or third most common kind of CR (the most common being reference resolution). (Rodríguez and Schlangen, 2004) reports that 22% of the CRs found by them in a German task-oriented spoken dialogue belonged to level 4, while (Rieser and Moore, 2005) reports 8% (a high percentage considering that the channel quality was poor and caused a 31% of acoustic problems).

Fourth level CRs are not only frequent but there are studies that show that the hearer in fact prefers them. That is, if the dialogue shows a higher amount of task related clarifications (instead of, conventional CRs such as "what?") hearers qualitative evaluate the task as more successful (Skantze, 2007). (Gabsdil, 2003) and (Rieser and Moore, 2005) also agree that for task-oriented dialogues the hearer should present a task-level reformulation to be confirmed rather than asking for repetition, thereby showing his subjective understanding to the other dialogue participants. Gabsdil briefly suggests a step in this direction:

Task-level reformulations might benefit from systems that have access to effects of action operators or other ways to compute task-level implications. (Gabsdil, 2003, p.29 and p.34)

In the rest of the paper we propose a framework that formalizes how to compute task-level implicatures and that suggests a finer-grained classification for CRs in level 4. But first, in Section 2.3 we present empirical findings that motivate such a framework.

2.3 Empirical: The SCARE corpus

The SCARE corpus (Stoia et al., 2008) consists of fifteen English spontaneous dialogues situated in an instruction giving task[1]. It was collected using the Quake environment, a first-person virtual reality game. The task consists of a direction giver (DG) instructing a direction follower (DF) on how to complete several tasks in a simulated game world. The corpus contains the collected audio and video, as well as word-aligned transcriptions.

The DF had no prior knowledge of the world map or tasks and relied on his partner, the DG, to guide him on completing the tasks. The DG had a map of the world and a list of tasks to complete (detailed in Appendix A.3). The partners spoke to each other through headset microphones; they could not see each other. As the participants collaborated on the tasks, the DG had instant feedback of the DF's location in the simulated world, because the game engine displayed the DF's first person view of the world on both the DG's and DF's computer monitors.

We analyzed the 15 transcripts that constitute the SCARE corpus while watching the associated videos to get familiarized with the experiment and evaluate its suitability for our purposes. Then, we randomly selected one dialogue; its transcript contains 449 turns and its video lasts 9 minutes and 12 seconds. Finally, we classified the clarification requests according to the levels of communication (see Figure 1). We found 29 clarification requests; so 6.5% of the turns are CRs. From these 29 CRs, 65% belong to the level 4 of Table 1, and 31% belonged to level 3 (most of them related to reference resolution). Only 4% of the CRs were acoustic (level 2) since the channel used was very reliable.

In fact we only found one CR of the form "what?" and it was a signal of incredulity of the effect of an action as can be seen below:

DG(1): and then cabinet should open
DF(2): did it
DF(3): nothing in it
DG(4): what?
DG(5): There should be a silencer there

Interestingly, the "what?" form of CR was reported as the most frequently found in "ordinary" dialogue in (Purver et al., 2003). This is not the case in the SCARE corpus. Furthermore, "what?" is usually assumed to be a CR that indicates a low level of coordination and is frequently classified as belonging to level 1 or 2. However, this is not the case in our example in which the CR is evidently related to the task structure and thus belongs to level 4. This is an example of why surface form is not reliable when classifying CRs.

[1]The corpus is freely available for research in http://slate.cse.ohio-state.edu/quake-corpora/scare/

2.4 Preliminary conclusions

In this preliminary study, the SCARE corpus seems to present more CRs than the corpus analyzed by previous work (which reports that 4% of the dialogue turns are CR). Furthermore, in distinction to results reported in Ginzburg (2009), most CRs occur at level 4. We believe this is naturally explained in politeness theory (Brown and Levinson, 1987).

The participants were punished if they performed steps of the task that they were not supposed to (see the instructions in Appendix A.1). This punishment might take precedence over the dispreference for CRs that is universal in dialogue due to politeness. CRs are perceived as a form of disagreement which is universally dispreferred according to politeness theory. The pairs of participants selected were friends so the level of intimacy among them was high, lowering the need of politeness strategies; a behavior that is also predicted by politeness theory. Finally, the participants received a set of instructions before the task started (see Appendix A) that includes information on the available actions in the simulated world and their expected effects. The participants make heavy use of this to produce high level clarification requests, instead of just signaling misunderstanding.

From these observations we draw the preliminary conclusion that clarification strategies depend on the information that is available to the dialogue participants (crucially including the information available before the dialogue starts) and on the constraints imposed on the interaction, such as politeness constraints. In Section 3 we describe the four information resources of our framework whose content depends on the information available to the dialogue participants. In Section 4 we introduce the reasoning tasks that use the information resources to infer the clarification potential of instructions. The study of the interaction between politeness constraints and clarification strategies seems promising, and we plan to address it in future work.

3 The information resources

The inference framework uses four information resources whose content depends on the information available to the dialogue participants. We describe each of them in turn and we illustrate their content using the SCARE experimental setup.

3.1 The world model

Since the kind of utterance that the framework handles are instructions that are supposed to be executed in a simulated world, the first required information resource is a model of this world. The world model is a knowledge base that represents the physical state of the simulated world. This knowledge base has complete and accurate information about the world that is relevant for completing the task at hand. It specifies properties of particular individuals (for example, an individual can be a *button* or a *cabinet*). Relationships between individuals are also represented here (such as the relationship between an object and its location). Such a knowledge base can be thought as a first-order model.

The content of the world model for the SCARE setup is a representation of the factual information provided to the DG before the experiment started, namely, a relational model of the map he received (see Figure 3 in Appendix A.3). Crucially, such a model contains all the functions associated with the buttons in the world and the contents of the cabinets (which are indicated on the map).

3.2 The dialogue model

Usually, this knowledge base starts empty; it is assumed to represent what the DF knows about the world. The information learned, either through the contributions made during the dialogue or by navigating the simulated world, are incrementally added to this knowledge base. The knowledge is also represented as a relational model and in fact this knowledge base will usually (but not necessarily) be a submodel of the world model.

The DF initial instructions in the SCARE setup include almost no factual information (as you can verify looking at his instructions in Appendix A.2). The only factual information that he received were pictures of some objects in the world so that he is able to recognize them. Such information is relevant mainly for referent resolution and this is not the focus of the current paper. Therefore, for our purposes we can assume that the dialogue model of the SCARE experiment starts empty.

3.3 The world actions

Crucially, the framework also includes the definitions of the actions that can be executed in the world (such as the actions *take* or *open*). Each ac-

tion is specified as a STRIPS-like operator (Fikes et al., 1972) detailing its arguments, preconditions and effects. The preconditions indicate the conditions that the world scenario must satisfy so that the action can be executed; the effects determine how the action changes the world when it is executed. These actions specify complete and accurate information about how the world behaves and together with the world model is assumed to represent what the DG knows about the world.

The SCARE world action database will contain a representation of the specification of the quake controls (see Appendix A.1) received by both participants and the extra action information that the DG received. First, he received a specification of the action *hide* that was not received by the DF. Second, if the DG read the instructions carefully, he knows that pressing a button can also cause things to move. The representation of this last action schema is shown in Appendix A.3.1.

3.4 The potential actions

The potential actions include representation of actions that the DF learned from the instructions he received before beginning the task. This includes the quake controls (see Appendix A.1) and also the action knowledge that he acquired during his learning phase (see appendix A.2). In the learning phase the direction follower learned that the effect of pressing a button can open a cabinet (if it was closed) or close it (if it was opened). Such knowledge is represented as a STRIPS-like operator like one showed in Appendix A.2.1.

3.5 Preliminary conclusions

An **action language** like PDDL (Gerevini and Long, 2005) can be used to specify the two action databases introduced above (in fact, the STRIPS fragment is enough). PDDL is the official language of the International Conference on Automated Planning and Scheduling since 1998. This means that most off-the-shelf planners that are available nowadays support this language, such as FF (Hoffmann and Nebel, 2001) and SGPlan (Hsu et al., 2006).

As we said in the previous section, the world model and the dialogue model are just relational structures like the one showed in Figure 3 (in the appendix). These relational structures can be directly expressed as a set of literals which is the format used to specify the **initial state** of a planning problem.

The information resources then constitute almost everything that is needed in order to specify a complete **planning problem**, as expected by current planners, the only element that the framework is missing is the **goal**. With a set of action schemas (i.e. action operators), an initial state and a goal as input, a planner is able to return a sequence of actions (i.e. a plan) that, when executed in the initial state, achieves the goal.

Planning is a **means-end inference task**, a kind of **practical inference** as defined by Kenny in (Kenny, 1966); and is a very popular inference task indeed as evidenced by the amount of work done in the area in the last two decades. However, *planning is not the only interesting means-end inference task*. One of the goals of the next section is to show exactly this: there is more to practical inference than planning.

4 The inference tasks

In this section we do two things. First, we say how current off-the-shelf planners can be used to infer part of the clarification potential of instructions. In particular we define what the missing element, the goal, is and we illustrate this with fragments of human-human dialogue of the SCARE corpus. Incidentally, we also show that clarification potential can not only be used for generating and interpreting CRs but also for performing acceptance and rejection acts. Second, we motivate and start to define one means-ends inference task that is not currently implemented, but that is crucial for inferring the clarification potential of instructions.

In order to better understand the examples below you may want to read the Appendix A first. The information in the Appendix was available to the participants when they performed the experiments and it's heavily used in the inferences they draw.

4.1 Planning: A means-end inference task

Shared-plan recognition —and *not* artificial intelligence planning— has been used for utterance interpretation (Lochbaum, 1998; Carberry and Lambert, 1999; Blaylock and Allen, 2005). In such plan recognition approaches each utterance adds a constraint to the plan that is partially filled out, and the goal of the conversation has to be inferred during the dialogue; that is, a *whole dialogue* is mapped to one shared plan. In our approach, *each instruction* is interpreted as a plan instead; that is,

we use planning at the utterance level and not at dialogue level.

Artificial intelligence planning has been used at utterance level (called micro-planning) for *generation* (Koller and Stone, 2007). We use artificial intelligence planning for *interpretation* of instructions instead.

In our framework, the goal of the planning problem are the *preconditions of instruction* for which the clarification potential is being calculated. Now, the planning problem has a goal, but there are two action databases and two initial states. Which one will be used for finding the clarification potential? In fact, all four.

When the DG gives an instruction, the DF has to interpret it in order to know what actions he has to perform (step 1 of the inference). The interpretation consists in trying to construct a plan that, when executed in the current state of the game world, achieves the goals of the instruction. The specification of such planning problem is as follows. The preconditions of the instruction are the *goal* of the planning problem, the dialogue model is the *initial state* and the potential actions are the *action operators*. With this information the off-the-shelf planner will find a *plan*, a sequence of actions that are the implicatures of the instruction.

Then (step 2 of the inference), an attempt to execute the plan on the the world model and using the world actions occurs. *Whenever the plan fails, there is a potential clarification.*

Using clarification potential to clarify: In the dialogue below, the participants are trying to move a picture from a wall to another wall (task 1 in Appendix A.3). The instruction that is being interpreted is the one uttered by the DG in (1). Using the information in the potential action database, the DF infers a plan that involves two implicatures, namely *picking up the picture* (in order to achieve the precondition of holding the picture), and *going to the wall* (inference step 1). However, this plan will fail when executed on the world model because the picture is *not takeable* and thus it cannot be picked, resulting in a potential clarification (inference step 2). This potential clarification, foreshadowed by (3), is finally made explicit by the CR in (4).

DG(1): well, put it on the opposite wall
DF(2): ok, control picks the .
DF(3): control's supposed to pick things up and .
DF(4): am I supposed to pick this thing?

A graphical representation of both steps of inference involved in this example is shown in Section B of the Appendix[2].

But also to produce evidence of rejection: In the dialogue below, the DG utters the instruction (1) knowing that the DF will not be able to follow it; the DG is just thinking aloud. If taken seriously, this instruction would involve the action *resolve the reference "cabinet nine"*. A precondition of this action is that the DF knows the numbers of the cabinets, but both participants know this is not the case, only the DG can see the map. That's why the rejection in (2) is received with laughs and the DG continues his loud thinking in (3) while looking at the map.

DG(1): we have to put it in cabinet nine .
DF(2): yeah, they're not numbered [laughs]
DG(3): [laughs] where is cabinet nine .

And to produce evidence of acceptance: The following dialogue fragment continues the fragment above. Now, the DG finally says where cabinet nine is in (4). And the DF comes up with the plan that he incrementally grounds making it explicit in (5), (7), and (9) while he is executing it; the plan achieves the precondition of the instruction *put* of being near the destination of the action, in this case "near cabinet nine". Uttering the steps of the plan that were not made explicit by the instruction is indeed a frequently used method for performing acceptance acts.

DG(4): it's . kinda like back where you started . so
DF(5): ok . so I have to go back through here .
DG(6): yeah
DF(7): and around the corner .
DG(8): right
DF(9): and then do I have to go back up the steps
DG(10): yeah
DF(11): alright, this is where we started
DG(12): ok . so your left ca- . the left one
DF(13): alright, so how do I open it?

In (13) the DF is not able to find a plan that achieves another precondition of the action *put*, namely that the destination container is opened, so he directly produces a CR about the precondition.

[2]The correct plan to achieve (1) involves pressing button 12, as you (and the DG) can verify on the map (in the Appendix).

4.2 Beyond classical planning: Other important means-end inference tasks

Consider the following example, here the DG just told the DF to press a button, in turn (1), with no further explanation. As a result of the action a cabinet opened, and the DF predicted that the following action requested would be (5). In (6) the DG confirms this hypothesis.

DG(1): press the button on the left [pause]
DG(2): and . uh [pause]
DF(3): [pause]
DG(4): [pause]
DF(5): put it in this cabinet?
DG(6): put it in that cabinet, yeah

The inference that the DF did in order to produce (5) can be defined as another means-end inference task which involves finding the **next relevant actions**. The input of such task would also consist of an initial state, a set of possible actions but it will contain one observed action (in the example, action (1)). Inferring the next relevant action consists in inferring the affordabilities (i.e. the set of executable actions) of the initial state and the affordabilities of the state after the observed action was executed. The **next relevant actions** will be those actions that were activated by the observed action. In the example above, the next relevant action that will be inferred is "put the thing you are carrying in the cabinet that just opened", just what the DF predicted in (5).

The definition of this inference task needs refining but it already constitutes an interesting example of a new form of means-ends reasoning.

There are further examples in the corpus that suggest the need for means-end inferences in situations in which a classical planner would just say "there is no plan". These are cases in which no complete plan can be found but the DF is anyway able to predict a possible course of action. For instance, in the last dialogue of Section 4.1, the DF does not stops in (13) and waits for an answer but he continues with:

DF(14): one of the buttons?
DG(15): yeah, it's the left one

Other CRs similar to this one, where a parameter of the action is ambiguous, is missing or is redundant, were also found in the corpus.

4.3 Preliminary Conclusions

The inference-tasks we discussed or just hinted to in this paper do not give a complete characterization of the kinds of clarification requests of level 4. It covers 14 of the 19 CRs in the SCARE dialogue analyzed in Section 2.3. CRs not covered at all have to do mainly with the fact that people do not completely remember (or trust) the instructions during the experiments or what themselves (or their partner) said a few turns before, such as the following one:

DG(1): you've to . like jump on it or something .
DF(2): I don't know if I can jump

Here, the DF does not remember that he can jump using the Spacebar as stated in the instructions he received (Appendix A.1).

In order to account for these cases it is necessary to consider how conversation is useful for overcoming also this issue. The fact that people's memory is non reliable is intrinsic to communication and here again, communication must provide intrinsic mechanisms to deal with it. Modeling such things are challenges that a complete theory of communication will have to face.

5 Conclusions

Conversational implicatures are negotiable, this is the characteristic that distinguishes them from other kinds of meanings (like entailments). Dialogue provides an intrinsic mechanism for carrying out negotiations of meaning, namely clarifications. So our hypothesis is that conversational implicatures are a rich source of clarification requests.

In order to investigate this hypothesis, we reviewed theoretical work from pragmatics, practical work from the dialogue system community and we presented empirical evidence from spontaneous dialogues situated in an instruction giving task. Also, we presented a framework in which (part of) the clarification potential of an instruction is generated by inferring its conversational implicatures. We believe that this is a step towards defining a clear functional criteria for identifying and classifying the clarification requests at level 4 of communication.

But much more remains to be done. The empirical results we present here are suggestive but preliminary; we are currently in the process of evaluating their reliability measuring inter-annotator

agreement. Moreover, in the course of this work we noticed a promising link between clarification strategies and politeness constraints which we plan to develop in future work. Also, we are particularly interested in means-ends reasoning other than planning, something we have merely hinted at in this paper; these tasks still need to be formally defined, implemented and tested. Finally, we are considering the GIVE challenge (Byron et al., 2009) as a possible setting for evaluating our work (our framework could predict potential clarification requests from the users).

There is lot to do yet, but we believe that the interplay between conversational implicatures and clarification mechanisms will play a crucial role in future theories of communication.

References

Jens Allwood. 1995. An activity based approach to pragmatics. In *Abduction, Belief and Context in Dialogue: Studies in Computational Pragmatics*, pages 47–80. University of Göteborg.

Nate Blaylock and James Allen. 2005. A collaborative problem-solving model of dialogue. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 200–211, Lisbon, Portugal.

Penelope Brown and Stephen Levinson. 1987. *Politeness: Some universals in language usage*. Studies in Interactional Sociolinguistics.

Donna Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments (GIVE). In *Proc. of the 12th European Workshop on Natural Language Generation*, pages 165–173, Athens, Greece. ACL.

Sandra Carberry and Lynn Lambert. 1999. A process model for recognizing communicative acts and modeling negotiation subdialogues. *Computational Linguistics*, 25(1):1–53.

Herbert Clark. 1996. *Using Language*. Cambridge University Press, New York.

Richard Fikes, Peter Hart, and Nils Nilsson. 1972. Learning and executing generalized robot plans. *Artificial Intelligence*, 3:251–288.

Malte Gabsdil. 2003. Clarification in spoken dialogue systems. In *Proc of the AAAI Spring Symposium. Workshop on Natural Language Generation in Spoken and Written Dialogue*, pages 28–35.

Alfonso Gerevini and Derek Long. 2005. Plan constraints and preferences in PDDL3. Technical Report R.T. 2005-08-47, Brescia University, Italy.

Jonathan Ginzburg. 2009. *The interactive Stance: Meaning for Conversation*. CSLI Publications.

Paul Grice. 1975. Logic and conversation. In P. Cole and J. L. Morgan, editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press.

Jörg Hoffmann and Bernhard Nebel. 2001. The FF planning system: Fast plan generation through heuristic search. *JAIR*, 14:253–302.

Chih-Wei Hsu, Benjamin W. Wah, Ruoyun Huang, and Yixin Chen. 2006. New features in SGPlan for handling soft constraints and goal preferences in PDDL3.0. In *Proc of ICAPS*.

Anthony Kenny. 1966. Practical inference. *Analysis*, 26:65–75.

Alexander Koller and Matthew Stone. 2007. Sentence generation as planning. In *Proc. of ACL-07*, Prague.

Karen E. Lochbaum. 1998. A collaborative planning model of intentional structure. *Comput. Linguist.*, 24(4):525–572.

Matthew Purver, Jonathan Ginzburg, and Patrick Healey. 2003. On the means for clarification in dialogue. In *Current and New Directions in Discourse and Dialogue*, pages 235–255. Kluwer Academic Publishers.

Matthew Purver. 2004. *The Theory and Use of Clarification Requests in Dialogue*. Ph.D. thesis, King's College, University of London.

Verena Rieser and Johanna Moore. 2005. Implications for generating clarification requests in task-oriented dialogues. In *Proc of ACL*, pages 239–246.

Kepa Rodríguez and David Schlangen. 2004. Form, intonation and function of clarification requests in german task oriented spoken dialogues. In *Proc of SEMDIAL*, pages 101–108.

Emanuel Schegloff. 1987. Some sources of misunderstanding in talk-in-interaction. *Linguistics*, 8:201–218.

David Schlangen. 2004. Causes and strategies for requesting clarification in dialogue. In *Proc of SIGDIAL*.

Gabriel Skantze. 2007. *Error Handling in Spoken Dialogue Systems*. Ph.D. thesis, KTH - Royal Institute of Technology, Sweden.

Laura Stoia, Darla Shockley, Donna Byron, and Eric Fosler-Lussier. 2008. SCARE: A situated corpus with annotated referring expressions. In *Proc of LREC*.

Laura Stoia. 2007. *Noun Phrase Generation for Situated Dialogs*. Ph.D. thesis, Ohio State University, USA.

A Instructions for the DG and DF

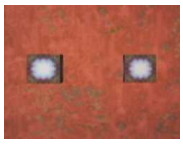
In this section, we specify the information that was available to the DG and the DF before the SCARE experiment started (adapted from (Stoia, 2007)). These instructions are crucial for our study since they define the content of the information resources of the inference framework described in this paper.

A.1 Instructions for both

The following specification of the Quake controls, that is, the possible actions in the simulated world, were received by all participants.

1. Use the arrow keys for **movement**:
   - Walk forward: ↑
   - Walk backward: ↓
   - Turn right: →
   - Turn left: ←
2. To **jump**: use Spacebar.
3. To **press a button**: Walk over the button. You will see it depress.
4. To **pick up an object**: Step onto the item then press Ctrl (Control key).
5. To **drop an object**: Hit TAB to see the list of items that you are currently carrying. Press the letter beside the item you wish to drop. Press TAB again to make the menu go away.

The participants also received the following pictures of possible objects in the simulated world so that they are able to recognize them.

Buttons	Cabinet
	

The following things were indicated as being objects that the DF can pick up and move:

Quad damage	Rebreather	Silencer
		

They also received the following warning: You will not be timed, but penalty points will be taken for pushing the wrong buttons or placing things in the wrong cabinets.

A.2 Instructions for the Direction Follower

Only the DF received the following information:

Phase 1: Learning the controls First you will be put into a small map with no partner, to get accustomed to the quake controls (detailed in Section A.1). Practice moving around using the arrow keys. Practice these actions:

1. Pick up the Rebreather or the Quad Damage.
2. Push the blue button to open the cabinet.
3. Drop the Quad Damage or the Rebreather inside the cabinet and close the door by pushing the button again.

Phase 2: Completing the task In this phase you will be put in a new location. Your partner will direct you in completing 5 tasks. He will see the same view that you are seeing, but you are the only one that can move around and act in the world.

A.2.1 Implications for the Potential Actions

In phase 1, when the DF is learning the controls, he learns that buttons can have the effect of opening closed cabinets and closing open cabinets. Such action is formalized as follows in PDDL (Gerevini and Long, 2005) and is included in the possible action database:

```
(:action press_button
   :parameters (?x ?y)
   :precondition
        (button ?x)
        (cabinet ?y)
        (opens ?x ?y)
   :effects
        (when (open ?y) (closed ?y))
        (when (closed ?y) (open ?y)))
```

Notice that this action operator has conditional effects in order to specify the action more succinctly. However, it is not mandatory for the action language to support conditional effects. This action could be specified with two actions in which the antecedent of the conditional effect is now a precondition.

A.3 Instructions for the Direction Giver

Only the DG received the following information:

Phase 1: Planning the task Your packet contains a **map** of the quake world with **5 objectives** that you have to direct your partner to perform. Read the instructions and take your time to plan the directions you want to give to your partner.

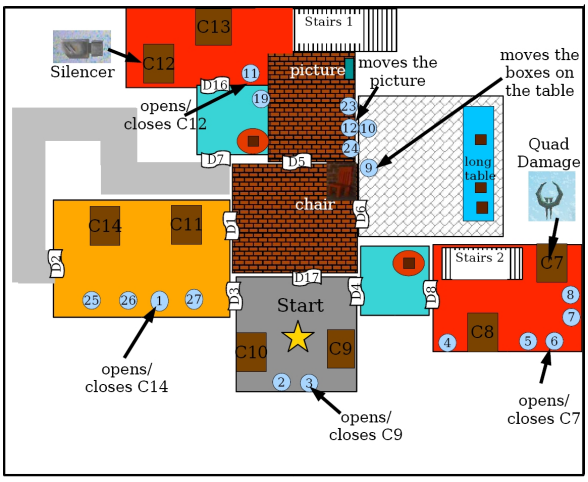


Figure 2: Map received by the DG (upper floor)

Phase 2: Directing the follower In this phase your partner will be placed into the world in the start position. Your monitor will show his/her view of the world as he/she moves around. He/she has no knowledge of the tasks, and has not received a map. You have to direct him/her through speech in order to complete the tasks. The objective is to complete all 5 tasks, but the order does not matter.

The tasks are:

1. Move the picture to the other wall.
2. Move the boxes on the long table so that the final configuration matches the picture below.

Picture	Long table
	

3. Hide the Rebreather in Cabinet9. To **hide** an item you have to find it, pick it up, drop it in the cabinet and close the door.
4. Hide the Silencer in Cabinet4.
5. Hide the Quad Damage in Cabinet14.
6. At the end, return to the starting point.

A.3.1 Implications for the World Actions

The functions of the buttons that can move things can be represented in the following action schema. If the thing is in it's original location (its location when the game starts), we say that is thing is `not-moved`. If the thing is in the goal position then we say that the thing is `moved`.

```
(:action press_button
   :parameters (?x ?y)
   :precondition
        (button ?x)
        (thing ?y)
        (moves ?x ?y)
   :effects
        (when (moved ?y) (not-moved ?y))
        (when (not-moved ?y) (moved ?y)))
```

A.3.2 Implications for the World Model

The world model is a relational model that represents the information provided by the map, including the functions of the buttons and the contents of the cabinets.

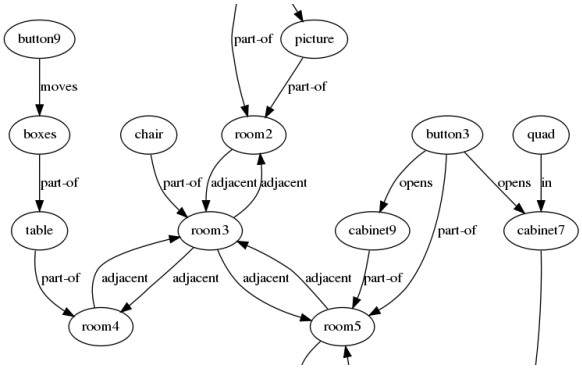


Figure 3: Fragment of the SCARE world model

B Clarification Potential Inference Steps

The following pictures illustrate how the implicatures of the instruction "put the picture on the opposite wall" are calculated using the dialogue model (Figure 4) and used to predict the CR "Am I supposed to pick up this thing?" (Figure 5).

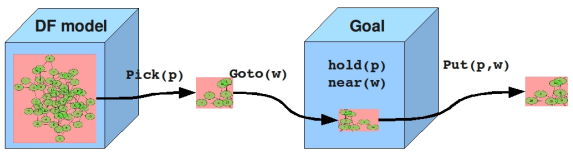


Figure 4: Step 1 - Calculating the implicatures

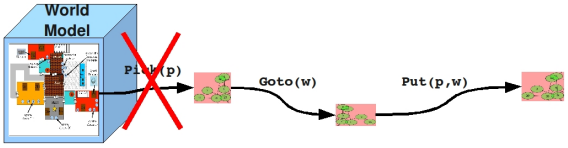


Figure 5: Step 2 - Predicting the CR

What do We Know about Conversation Participants: Experiments on Conversation Entailment

Chen Zhang **Joyce Y. Chai**
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824, USA
{zhangch6, jchai}@cse.msu.edu

Abstract

Given the increasing amount of conversation data, techniques to automatically acquire information about conversation participants have become more important. Towards this goal, we investigate the problem of conversation entailment, a task that determines whether a given conversation discourse entails a hypothesis about the participants. This paper describes the challenges related to conversation entailment based on our collected data and presents a probabilistic framework that incorporates conversation context in entailment prediction. Our preliminary experimental results have shown that conversation context, in particular dialogue act, plays an important role in conversation entailment.

1 Introduction

Conversation is a joint activity between its participants (Clark, 1996). Their goals and their understanding of mutual beliefs of each other shape the linguistic discourse of conversation. In turn, this linguistic discourse provides tremendous information about conversation participants. Given the increasing amount of available conversation data (e.g., conversation scripts such as meeting scripts, court records, and online chatting), an important question is *what do we know about conversation participants?* The capability to automatically acquire such information can benefit many applications, for example, development of social networks and discovery of social dynamics.

Related to this question, previous work has developed techniques to extract profiling information about participants from conversation interviews (Jing et al., 2007) and to automatically identify dynamics between conversation participants such as agreement/disagreement from multiparty meeting scripts (Galley et al., 2004). We approach this question from a different angle as a *conversation entailment* problem: given a conversation discourse D and a hypothesis H concerning its participant, the goal is to identify whether D entails H. For instance, in the following example, the first hypothesis can be entailed from the dialogue segment while the second hypothesis cannot.

Example 1:

Dialogue Segment:

A: And where about were you born?
B: Up in Person Country.

Hypothesis:

(1) B was born in Person Country.
(2) B lives in Person Country.

Inspired by textual entailment (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007), conversation entailment provides an intermediate step towards acquiring information about conversation participants. What we should know or would like to know about a participant can be rather open. The type of information needed about participants is also application-dependent and difficult to generalize. In conversation entailment, we will not face this problem since hypotheses can be used to express any type of information about a participant one might be interested in. Although hypotheses are currently given in our investigation, they can potentially be automatically generated based on information needs and/or theories on cognitive status/mental models of conversation participants. The capability to make correct entailment judgements based on these hypotheses will benefit many applications such as information extraction, question answering, and summarization.

As a first step in our investigation, we collected a corpus of conversation entailment data from nineteen human annotators. Our data showed that conversation entailment is more challenging than

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 206–215,
Queen Mary University of London, September 2009.

the textual entailment task due to unique characteristics about conversation and conversational implicature. To predict entailment, we developed a probabilisitic framework that incorporates semantic representation of conversation context. Our preliminary experimental results have shown that conversation context, in particular dialogue acts, play an important role in conversation entailment.

2 Related Work

Recent work has applied different approaches to acquire information about conversation participants based on human-human conversation scripts, for example, to extract profiling information from conversation interviews (Jing et al., 2007) and to identify agreement/disagreement between participants from multiparty meeting scripts (Galley et al., 2004). In human-machine conversation, inference about conversation participants has been studied as a part of user modeling. For example, earlier work has investigated inference of user intention from utterances to control clarification dialogue (Horvitz and Paek, 2001) and recognition of user emotion and attitude from utterances for intelligent tutoring systems (Litman and Forbes-Riley, 2006). In contrast to previous work, we propose a new angle to address information acquisition about conversation participants, namely, through conversation entailment.

This work is inspired by a large body of recent work on textual entailment initiated by the PASCAL RTE Challenge (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007). Nevertheless, conversation discourse is very different from written monologue discourse. The conversation discourse is shaped by the goals of its participants and their mutual beliefs. The key distinctive features include turn-taking between participants, grounding between participants, and different linguistic phenomena of utterances (e.g., utterances in conversation tend to be shorter, with disfluency, and sometimes incomplete or ungrammatical). It is the goal of this paper to explore how techniques developed for textual entailment can be extended to address these unique behaviors in conversation entailment.

3 Experimental Data

The first step in our investigation is to collect entailment data to help us better understand the problem and facilitate algorithm development and evaluation.

3.1 Data Collection Procedure

We selected 50 dialogues from the Switchboard corpus (Godfrey and Holliman, 1997). In each of these dialogues, two participants discuss a topic of interest (e.g., sports activities, corporate culture, etc.). To focus our work on the entailment problem, we use the transcribed scripts of the dialogues in our experiments. We also make use of available annotations such as syntactic structures, disfluency markers, and dialogue acts.

We had 15 volunteer annotators read the selected dialogues and create hypotheses about participants. As a result, a total of 1096 entailment examples were created. Each example consists of a snippet from the dialogue (referred to as *dialogue segment* in the rest of this paper), a hypothesis statement, and a truth value indicating whether the hypothesis can be inferred from the snippet given the whole history of that dialogue session. During annotation, we asked the annotators to provide balanced examples for each dialogue. That is, roughly half of the hypotheses are truly entailed and half are not. Special attention was given to negative entailment examples. Since any arbitrary hypotheses that are completely irrelevant can be negative examples, a special criteria is enforced that any negative examples should have a majority word overlap with the snippet. In addition, inspired by previous work (Jing et al., 2007; Galley et al., 2004), we particularly asked annotators to provide hypotheses that address the profiling information of the participants, their opinions and desires, as well as the dynamic communicative relations between participants.

A recent study shows that for many NLP annotation tasks, the reliability of a small number of non-expert annotations is on par with that of an expert annotator (Snow et al., 2008). It also found that for tasks such as affection recognition, an average of four non-expert labels per item are capable of emulating expert-level label quality. Based on this finding, in our study the entailment judgement for each example was further independently annotated by four annotators (who were not the original contributors of the hypotheses). As a result, on average each entailment example (i.e., a pair of snippet and hypothesis) received five judgements.

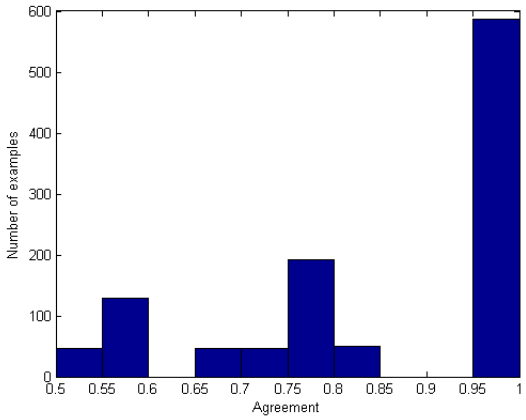


Figure 1: Agreement histogram of entailment judgements

3.2 Data and Examples

Figure 1 shows a histogram of the agreements of collected judgements. It indicates that conversation entailment is in fact a quite difficult task even for humans. Only 53% of all the examples (586 out of 1096) are agreed upon by all human annotators. The disagreement between users sometimes is caused by language ambiguity since conversation scripts are often short and without clear sentence boundaries. For example,

Example 2:

Dialogue Segment:

A: Margaret Thatcher was prime minister, uh, uh, in India, so many, uh, women are heads of state.

Hypothesis:

A believes that Margaret Thatcher was prime minister of India.

In the utterance of speaker A, the prepositional phrase *in India* is ambiguous because it can either be attached to the preceding sentence, which sufficiently entails the hypothesis; or it can be attached to the succeeding sentence, which leaves it unclear which country A believes Margaret Thatcher was prime minister of.

Difference in recognition and handling of conversational implicature is another issue that led to disagreement among annotators. For example:

Example 3:

Dialogue Segment:

A: Um, I had a friend who had fixed some, uh, chili, buffalo chili and, about a week before we went to see the movie.

Hypothesis:

A ate some buffalo chili.

Example 4:

Dialogue Segment:

B: Um, I've visited the Wyoming area. I'm not sure exactly where *Dances with Wolves* was filmed.

Hypothesis:

B thinks *Dances with Wolves* was filmed in Wyoming.

In the first example, a listener could assume that A follows the maxim of relevance. Therefore, a natural inference that makes "fixing of buffalo chili" relevant is that A ate the buffalo chili. Similarly, in the second example, the speaker A mentions a visit to Wyoming, which can be considered relevant to the filming place of DANCES WITH WOLVES. Some annotators recognized such relevance and some did not.

Given the discrepencies between annotators, we selected 875 examples which have at least 75% agreement among the judgements in our current investigation. We further selected one-third of this data (291 examples) as our development data. The experiments reported in Section 5 are based on this development set.

3.3 Types of Hypotheses

The hypotheses collected from our study can be categorzied into the following four types:

Fact. Facts about the participants. This includes: (1) profiling information about individual participants (e.g., occupation, birth place, etc.); (2) activities associated with individual participants (e.g., A bikes to work everyday); and (3) social relations between participants (e.g., A and B are coworkers, A and B went to college together).

Belief. Participants' beliefs and opinions about the physical world. Any statement about the physical world in fact is a belief of the speaker. Technically, the state of the physical world that involves the speaker him/herself is also a type of belief. However, here we assume a statement about oneself is true and is considered as a *fact*.

Desire. Participants' desire of certain actions or outcomes (e.g., A wants to find a university job). These desires represent the states of the world the participant finds pleasant (although they could be conflicting to each other).

Intent. Participants' deliberated intent, in particular communicative intention which captures the intent from one participant on the other participant such as whether A agrees/disagrees with B

on some issue, whether A intends to convince B on something, etc.

Most of these types are motivated by the Belief-Desire-Intention (BDI) model, which represents key mental states and reflects the *thoughts* of a conversation participant. *Desire* is different from *intention*. The former arises subconsciously and the latter arise from rational deliberation that takes into consideration desires and beliefs (Allen, 1995). The *fact* type represents the facts about a participant. Both thoughts and facts are critical to characterize a participant and thus important to serve many other downstream applications. The above four types account for 47.1%, 34.0%, 10.7%, and 8.2% of our development set respectively.

4 A Probabilistic Framework

Following previous work (Haghighi et al., 2005; de Salvo Braz et al., 2005; MacCartney et al., 2006), we approach conversation entailment using a probabilistic framework. To predict whether a hypothesis statement H can be inferred from a dialogue segment D, we estimate the probability

$$P(D \vDash H|D, H)$$

Suppose we have a representation of a dialogue segment D in m clauses $d_1, \ldots, d_m$ and a representation of the hypothesis H in n clauses $h_1, \ldots, h_n$. Since a hypothesis is the conjunction of the decomposed clauses, whether it can be inferred from a segment is equivalent to whether all of its clauses can be inferred from the segment. We further simplify the problem by assuming that whether a clause is entailed from a dialogue segment is conditionally independent from other clauses. Note that this conditional independence assumption is an over-simplification, but it gets things started. Therefore:

$$\begin{aligned}
&P(D \vDash H|D, H) \\
&\quad = P(d_1 \ldots d_m \vDash h_1 \ldots h_n|d_1, \ldots, d_m, h_1, \ldots, h_n) \\
&\quad = P(D \vDash h_1, \ldots, D \vDash h_n|D, h_1, \ldots, h_n) \\
&\quad = \prod_{j=1}^{n} P(D \vDash h_j|D = d_1 \ldots d_m, h_j) \\
&\quad = \prod_{j=1}^{n} P(d_1 \ldots d_m \vDash h_j|d_1, \ldots, d_m, h_j) \qquad (1)
\end{aligned}$$

If this likelihood is above a certain threshold (e.g., 0.5 in our experiments), then H is considered as a true entailment from D.

Given this framework, two important questions are: (1) how to represent and automatically create the clauses from each pair of dialogue segment and hypothesis; and (2) how to estimate probabilities as shown in Equation 1?

4.1 Clause Representation

Our clause representation is inspired by previous work on textual entailment (Dagan et al., 2005; Bar-Haim et al., 2006; Giampiccolo et al., 2007). Clause representation has several advantages. First, it can be acquired automatically from a parse tree (e.g., dependency parser). Second, it can be used to facilitate both logic-based reasoning as in (Tatu and Moldovan, 2005; Bos and Markert, 2005; Raina et al., 2005) or probabilistic reasoning as in (Haghighi et al., 2005; de Salvo Braz et al., 2005; MacCartney et al., 2006). The key difference between our work and previous work on textual entailment is the representation of conversation discourse, which has not been considered in previous work but is important for conversation entailment, as we will see later.

More specifically, a clause is made up by two components: **Term** and **Predicate**.

Term: A term can be an *entity* or an *event*. An *entity* refers to a person, a place, an organization, or other real world entities. This follows the concept of *mention* in the Automatic Content Extraction (ACE) evaluation (Doddington et al., 2004). An *event* refers to an action or an activity. For example, from the sentence "John married Eva in 1940" we can identify an event of marriage. Following the neo-Davidsonian representation (Parsons, 1990), all the events are reified as terms in our representation.

Predicate: A predicate represents either a *property* (i.e., unary) for a term or a *relation* (i.e., binary) between two terms. For example, an entity *company* has a property of *Russian* as in the phrase "a Russian company" (i.e., $Russian(company)$). An event *visit* has a property of *recently* (i.e., $recently(visit)$) as in the phrase "visit Brazil recently". From the phrase "Prime Minister recently visited Brazil", there are binary relations: *Prime Minister* is the subject of the event *visit* (i.e., $subj(visit, Prime\ Minister)$) and *Brazil* is the object of the *visit* (i.e., $obj(visit, Brazil)$).

This representation is a direct conversion from the dependency structure and can be used to represent the semantics of utterances in the dialogue

segments and the semantics of hypotheses. For example,

Example 5:

Dialogue Segment:

B: Have you seen *Sleeping with the Enemy*?
A: No. I've heard that's really great, though.
B: You have to go see that one.

Hypothesis:

B suggests A to watch *Sleeping with the Enemy*.

Appendix A shows the dependency structure of the dialogue utterances and the hypothesis from Example 5. Appendix B shows the corresponding clause representation of the dialogue segment and the hypothesis. Note that in this representation, *you* and *I* are replaced with the respective participants. Since the clauses are generated based on parse trees, most relational predicates are syntactic-driven.

To facilitate conversation entailment, we further augment the representation of a dialogue segment by incorporating conversation context. Appendix C shows the augmented representation for Example 5. It represents the following additional information:

- **Utterance:** A group of pseudo terms u_1, u_2, ... are used to represent individual utterances.
- **Participant:** A relational clause $speaker(\cdot,\cdot)$ is used to represent the speaker of this utterance, e.g., $speaker(u_1, B)$.
- **Content:** A relational clause $content(\cdot,\cdot)$ is used to represent the content of an utterance where the second term is the *head* of the utterance as identified in the parsing structure. e.g., $content(u_3, heard)$
- **Dialogue act:** A relational clause $act(\cdot,\cdot)$ is used to represent the dialogue act of the speaker for a particular utterance. e.g., $act(u_2, no\_answer)$. A set of 42 dialogue acts from the Switchboard annotation are used here (Godfrey and Holliman, 1997).
- **Utterance flow:** A relational clause $follow(\cdot,\cdot)$ is used to connect each pair of adjacent utterances. e.g., $follow(u_2, u_1)$. We currently do not consider overlap in utterances, but our representation can be modified to handle this situation by introducing additional predicates.

4.2 Entailment Prediction

Given the clause representation for a conversation segment and a hypothesis, the next step is to make an entailment prediction (as in Equation 1) based on two models: an *Alignment Model* and an *Inference Model*.

4.2.1 Alignment Model

The alignment model is to find alignments (or matches) between terms in the clause representation for a hypothesis and those in the clause representation for a conversation segment. We define an **alignment** as a mapping function g between a term x in the dialogue segment and a term y in the hypothesis. $g(x, y) = 1$ if x and y are aligned; otherwise $g(x, y) = 0$. Note that a verb can be aligned to a noun as in $g(sell, sale) = 1$. It is also possible that there are multiple terms from the segment mapped to one term in the hypothesis, or vice versa.

For any two terms x and y, the problem of predicting the alignment function $g(x, y)$ can be formulated as a binary classification problem. We used several features to train the classifier, which include whether x and y are the same (or have the same stem), whether one term is an acronym of the other, and their WordNet and distributional similarities (Lin, 1998).

Given an augmented representation with conversation context (as in Appendix C), we also align event terms in the hypothesis (e.g., *suggest* in Example 5) to (pseudo) utterance terms in the dialogue segment. We call it a *pseudo alignment*. This is currently done by a set of rules which associate event terms in the hypotheses with dialogue acts. For example, the event term *suggest* may be aligned to an utterance with dialogue act of *opinion*. Appendix D gives a correct alignment for Example 5, in which $g(u_4, x_1) = 1$ is a pseudo alignment.

4.2.2 Inference Model

As shown in Equation 1, to predict the inference of the entire hypothesis, we need to calculate the probability that the dialogue segment entails each clause from the hypothesis. More specifically, given a clause from the hypothesis h_j, a set of clauses from the dialogue segment $d_1, \ldots, d_m$, and an alignment function g between them derived by the method described in Section 4.2.1, we predict whether $d_1, \ldots, d_m$ entails h_j under the alignment g using two different classification models,

depending on whether h_j is a property or a relation (i.e. whether it takes one argument ($h_j(\cdot)$) or two arguments ($h_j(\cdot,\cdot)$)):

Given a property clause from the hypothesis, $h_j(x)$, we look for all the property clauses in the dialogue segment that describes the same term as x, i.e. a clause set $D' = \{d_i(x')|d_i(x') \in D, g(x', x) = 1\}$. Then we predict whether $h_j(x)$ can be inferred from the clauses in D' by binary classification, using a set of features similar to those used in the alignment model.

Given a relational clause from the hypothesis, $h_j(x, y)$, we look for the relation between the counterparts of x and y in the dialogue segment. That is, we find the set of terms $X' = \{x'|x' \in D, g(x', x) = 1\}$ and the set of terms $Y' = \{y'|y' \in D, g(y', y) = 1\}$ and look for the closest relation between these two sets of terms in the dependency structure. If there is a path between any $x' \in X'$ and any $y' \in Y'$ in the dependency structure with a length smaller than a threshold λ_L, we predict that $h_j(x, y)$ can be inferred. Note that our current handling of the relational clauses is rather simplified. It only captures whether two terms from an hypothesis are connected by any relation in the dialogue segment.

Appendix E shows the inference procedure of the four hypothesis clauses in Example 5. For each relational clause $h_j(x, y)$, the shortest path between the corresponding X' and Y' has a length of 3 or less, so each of these four clauses is entailed from the dialogue segment. Based on Equation 1 we can conclude that the overall hypothesis is entailed.

We trained the alignment model and the inference model (e.g.,the threshold λ_L) based on the development data provided by the PASCAL 3 challenges on textual entailment.

5 Experimental Results

To understand unique behaviors of conversation entailment, we focused our current experiments on the development dataset (see Section 3.2). We are particularly interested in how the techniques for textual entailment can be improved for conversation entailment. To do so, we applied our entailment framework on the test data of the PASCAL-3 RTE Challenge (Giampiccolo et al., 2007). Among 800 testing examples, our approach achieved an accuracy of 60.6%. This result is on par with the performance of the median system of accuracy 61.8% (z-test, p=0.63) in the PASCAL-3 RTE Challenge. Our current approach is very lean on the use of external knowledge. Its competitive performance sets up a reasonable baseline for our investigation on conversation entailment. This same system, modified to tailor linguistic characteristics of conversation (e.g., removal of disfluency), was used as the baseline in our experiments.

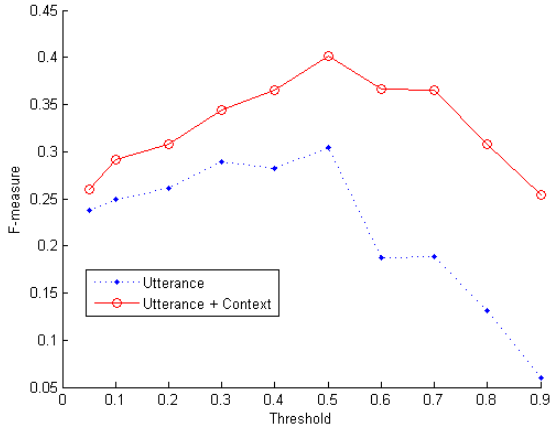
5.1 Event Alignment

To understand the effect of conversation context in the event alignment, we compared two configurations of alignment model for events. The first configuration is based on the clause representation of semantics of utterances (as shown in Appendix B). This is the same configuration as used in textual entailment. The second configuration is based on representation of both semantics from utterances and conversation context (as shown in Appendix C). We evaluate how well each configuration aligns the event terms based on the pairwise alignment decision: for any event term t_H in the hypothesis and any term t_D in the dialogue, whether the model can correctly predict that the two terms should be aligned.

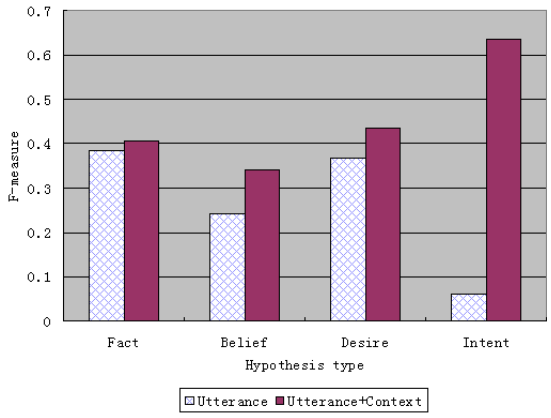
Figure 2(a) shows the comparison of F-measure between the two models. Depending on the threshold of alignment prediction, the precision and recall of the prediction vary. When the threshold is lower, the models tend to give more alignments, resulting in lower precision and higher recall. When the threshold is higher, the models tend to give fewer alignments, thus resulting in higher precision but lower recall. When the threshold is around 0.5, the alignment reaches its best F-measure. Regardless of what threshold is chosen, the model based on both utterance and context consistently works better. Figure 2(b) shows the breakdown based on the types of hypothesis (at threshold 0.5). The model that incorporates conversation context consistently performs better for all types. Its improvement is particularly significant for the *intent* type of hypothesis.

These results are not surprising. Many event terms in hypotheses (e.g., suggest, think, etc.) do not have their counterparts directly expressed in utterances in the dialogue discourse. Only through the modeling of dialog acts, these terms can be aligned to potential pseudo terms in the dialogue

segment. For the *fact* type hypotheses, the event terms in the hypotheses generally have their counterparts in the dialogue discourse. That explains why the improvement for the *fact* type using conversation context is minimal.



(a) Overall comparison on F-measure



(b) Comparison for different types of hypothesis

Figure 2: Experimental results on event alignment

5.2 Entailment Prediction

Given correct alignments, we further evaluated entailment prediction based on three configurations of the inference model: (1) the same inference model learned from the textual entailment data and tested on the PASCAL-3 RTE Challenge (Text); (2) an improved model incorporating a number of features relevant to dialogues (especially syntactic structure of utterances) based on representations without conversation context as in Appendix B (+Dialogue); (3) a further improved model based on augmented representations of conversation context and using dialogue acts during the prediction of entailment as in Appendix C (+Context).

System	Acc	Prec	Recall	F
Text	53.6%	71.6%	29.3%	41.6%
+Dialogue	58.4%	84.1%	32.3%	46.7%
+Context	67.7%	91.7%	47.0%	62.1%

Table 1: Experimental results on entailment prediction

For each configuration we present two evaluation metrics: an accuracy of the overall prediction and a precision-recall measurement for the positive entailment examples. All the evaluations are performed on our development data, which has 56.4% of positive examples and 43.6% of negative examples.

The evaluations results are shown in Table 1. The system learned from textual entailment performs lower than the prediction based on the majority class (56.4%). Incorporating syntactic features of dialogues did better but the difference is not statistically significant. Incorporating conversation context, especially dialogue acts, achieves significantly better performance (z-test, $p < 0.005$).

Table 2 shows the comparison of the three configurations based on different types of hypothesis. As expected, the basic system trained on textual entailment is not capable for any *intent* type of hypotheses. Modeling conversation context with dialogue acts improves inference for all types of hypothesis, with most significant improvement for the *belief*, *desire*, and *intent* types of hypothesis.

6 Conclusion

This paper describes our initial investigation on conversation entailment to address information acquisition about conversation participants. Since there are so many variables involved in the prediction, our experiments have been focused on a set of development data where most of the features are annotated. This allowed us to study the effect of conversation context in both alignment and entailment. Our future work will enhance the current approach by training the models based on our development data and evaluate them on the testing data. Conversation entailment is an important task. Although the current exercise is targeted to process conversation scripts from human-human conversation, it can potentially benefit human machine conversation by enabling automated agents to gain better understanding of their conversation

System	Fact		Belief		Desire		Intent	
	Acc	F	Acc	F	Acc	F	Acc	F
Text	58.4%	51.3%	52.5%	37.3%	51.6%	34.8%	33.3%	0
+Dialogue	68.6%	62.6%	53.5%	36.1%	48.4%	33.3%	33.3%	0
+Context	70.8%	64.9%	67.7%	62.8%	58.1%	47.8%	62.5%	60.9%

Table 2: Experimental results on entailment prediction for different types of hypotheses

partners.

Acknowledgments

This work was partially supported by IIS-0347548 and IIS-0840538 from the National Science Foundation. We thank the anonymous reviewers for their valuable comments and suggestions.

References

James Allen. 1995. *Natural language understanding*. The Benjamin/Cummings Publishing Company, Inc.

Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, Venice, Italy.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In *Proceedings of HLT-EMNLP*, pages 628–635.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *PASCAL Challenges Workshop on Recognising Textual Entailment*.

Rodrigo de Salvo Braz, Roxana Girju, Vasin Punyakanok, Dan Roth, and Mark Sammons. 2005. An inference model for semantic entailment in natural language. In *Proceedings of AAAI*.

G. Doddington, A. Mitchell, M. Przybocki, and L. Ramshaw. 2004. The automatic content extraction (ace) programctasks, data, and evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*.

Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. 2004. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *Proceedings of ACL*, pages 669–676.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9.

John J. Godfrey and Edward Holliman. 1997. *Switchboard-1 Release 2*. Linguistic Data Consortium, Philadelphia.

Aria Haghighi, Andrew Ng, and Christopher Manning. 2005. Robust textual inference via graph matching. In *Proceedings of HLT-EMNLP*, pages 387–394.

Eric Horvitz and Tim Paek. 2001. Harnessing models of users' goals to mediate clarification dialog in spoken language systems. In *Proceedings of the 8th International Conference on User Modeling*, pages 3–13.

Hongyan Jing, Nanda Kambhatla, and Salim Roukos. 2007. Extracting social networks and biographical facts from conversational speech transcripts. In *Proceedings of ACL*, pages 1040–1047.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*, pages 296–304.

Diane Litman and Katherine Forbes-Riley. 2006. Recognizing student emotions and attitudes on the basis of utterances in spoken tutoring dialogues with both human and computer tutors. *Speech Communication*, 48(5):559–590.

Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D. Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of HLT-NAACL*, pages 41–48.

Terence Parsons. 1990. *Events in the Semantics of English. A Study in Subatomic Semantics*. MIT Press.

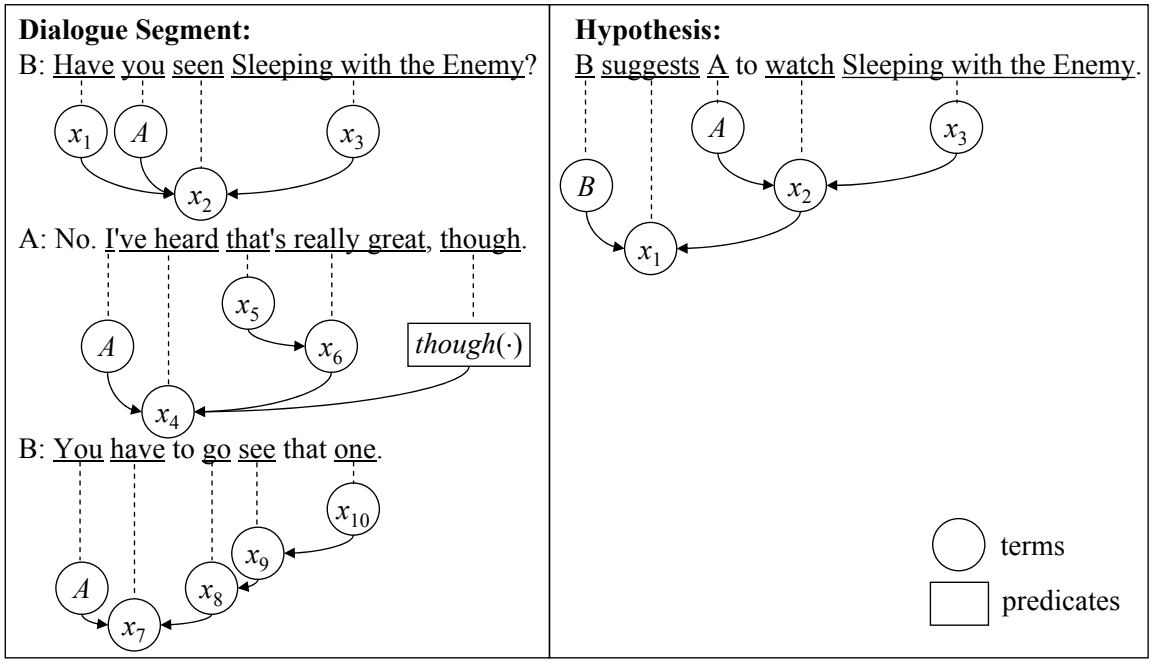
Rajat Raina, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of AAAI*, pages 1099–1105.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP*, pages 254–263.

Marta Tatu and Dan Moldovan. 2005. A semantic approach to recognizing textual entailment. In *Proceedings of HLT-EMNLP*, pages 371–378.

APPENDIX

A Dependency Structure of Dialogue Utterances and Hypothesis in Example 5



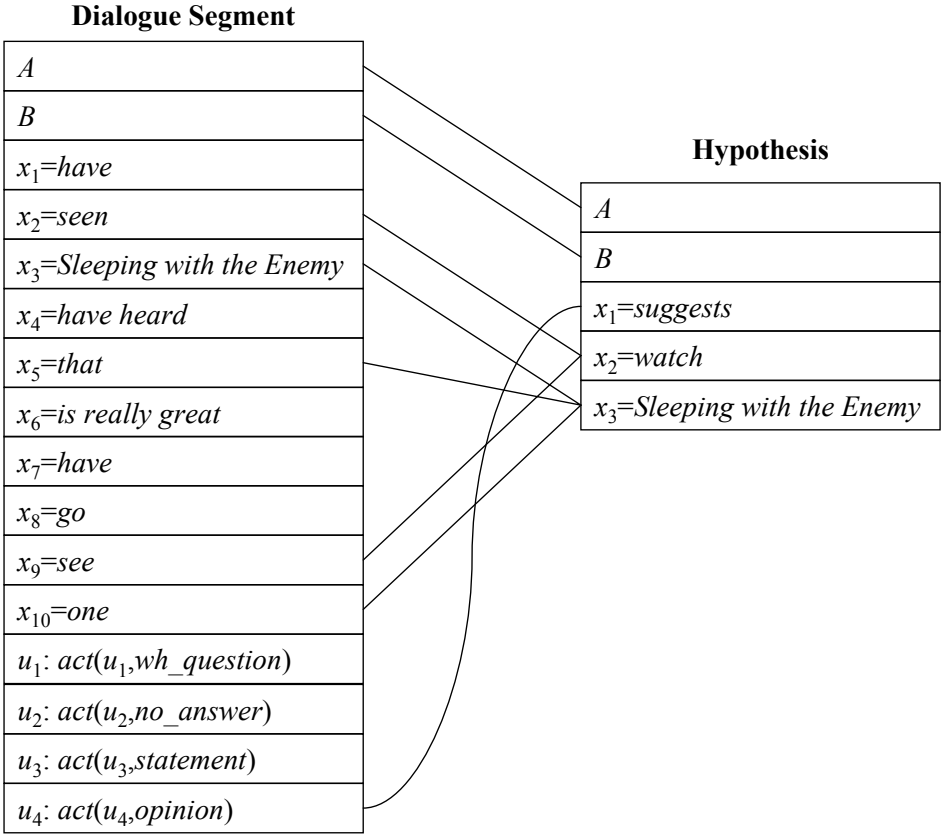
B Clause Representation of Dialogue Segment and Hypothesis for Example 5

Dialogue Segment:		
	Terms	Clauses
B:	x_1=*have*, x_2=*seen*, x_3=*Sleeping with the Enemy*, *A*	*subj*(x_2,*A*), *obj*(x_2,x_3), *aux*(x_2,x_1)
A:	x_4=*have heard*, x_5=*that*, x_6=*is really great*, *A*	*subj*(x_4,*A*), *obj*(x_4,x_6), *subj*(x_6,x_5), *though*(x_4)
B:	x_7=*have*, x_8=*go*, x_9=*see*, x_{10}=*one*, *A*	*subj*(x_7,*A*), *obj*(x_7,x_8), *obj*(x_8,x_9), *obj*(x_9,x_{10})
Hypothesis:		
	x_1=*suggests*, x_2=*watch*, x_3=*Sleeping with the Enemy*, *A*, *B*	*subj*(x_1,*B*), *obj*(x_1,*A*), *obj*(x_1,x_2), *obj*(x_2,x_3)

C Augmented Clause Representation of Dialogue Segment in Example 5

Dialogue Segment (with context representation):		
	Terms	Clauses
B:	u_1,x_1=*have*, x_2=*seen*, x_3=*Sleeping with the Enemy*, *A*, *B*	*speaker*(u_1,*B*), *content*(u_1,x_2), *act*(u_1,*wh_question*), *subj*(x_2,*A*), *obj*(x_2,x_3), *aux*(x_2,x_1)
A:	u_2, u_3, x_4=*have heard*, x_5=*that*, x_6=*is really great*, *A*	*speaker*(u_2,*A*), *content*(u_2,-), *act*(u_2,*no_answer*), *speaker*(u_3,*A*), *content*(u_3,x_4), *act*(u_3,*statement*), *subj*(x_4,*A*), *obj*(x_4,x_6), *subj*(x_6,x_5), *though*(x_4)
B:	u_4,x_7=*have*, x_8=*go*, x_9=*see*, x_{10}=*one*, *A*, *B*	*speaker*(u_4,*B*), *content*(u_4,x_7), *act*(u_4,*opinion*), *subj*(x_7,*A*), *obj*(x_7,x_8), *obj*(x_8,x_9), *obj*(x_9,x_{10})
		follow(u_2,u_1), *follow*(u_3,u_2), *follow*(u_4,u_3)

D The Alignment for Example 5



E The Prediction of Inference for the Hypothesis Clauses in Example 5

Hypothesis Clause	*subj*(x_1,*B*)		*obj*(x_1,*A*)		*obj*(x_1,x_2)		*obj*(x_2,x_3)	
Clause Type	relation		relation		relation		relation	
Terms in this Clause	x_1	*B*	x_1	*A*	x_1	x_2	x_2	x_3
Aligned Terms in the Dialogue Segment	u_4	*B*	u_4	*A*	u_4	x_2, x_9	x_2, x_9	x_3, x_5, x_{10}
Shortest Path between the Aligned Terms in the Dependency Structure of Dialogue Segment	*speaker*(u_4,*B*)		*content*(u_4,x_7), *subj*(x_7,*A*)		*content*(u_4,x_7), *obj*(x_7,x_8), *obj*(x_8,x_9)		*obj*(x_9,x_{10})	
Path Length	1		2		3		1	
Hypothesis Clause Entailed?	yes		yes		yes		yes	

Invited Talk

Artificial Companions as Dialogue Agents

Yorick Wilks
Department of Computer Science
University of Sheffield
Sheffield S1 4DP, UK
www.dcs.shef.ac.uk/~yorick
yorick@dcs.sheffield.ac.uk

COMPANIONS is an EU project that aims to change the way we think about the relationships of people to computers and the Internet by developing a virtual conversational 'Companion'. This is intended as an agent or 'presence' that stays with the user for long periods of time, developing a relationship and 'knowing' its owners preferences and wishes. The Companion communicates with the user primarily through speech. This paper describes the functionality and system modules of the Senior Companion, one of two initial prototypes built in the first two years of the project. The Senior Companion provides a multimodal interface for eliciting and retrieving personal information from the elderly user through a conversation about their photographs. The Companion will, through conversation, elicit their life memories, often prompted by discussion of their photographs; the aim is that the Companion should come to know a great deal about its user, their tastes, likes, dislikes, emotional reactions etc, through long periods of conversation. It is a further assumption that most life information will be stored on the internet (as in the Memories for Life project: http://www.memoriesforlife.org/) and the SC is linked directly to photo inventories in Facebook, to gain initial information about people and relationships, as well as to Wikipedia to enable it to respond about places mentioned in conversations about images. The overall aim of the SC, not yet achieved, is to produce a coherent life narrative for its user from these materials, although its short term goals are to assist, amuse, entertain and gain the trust of the user. The Senior Companion uses Information Extraction to get content from the speech input, rather than conventional parsing, and retains utterance content, extracted internet information and ontologies all in RDF formalism over which it does primitive reasoning about people. It has a dialogue manager virtual machine intended to capture mixed initiative, between Companion and user, and which can be a basis for later replacement by learned components.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, page 216,
Queen Mary University of London, September 2009.

Effects of Conversational Agents on Human Communication in Thought-Evoking Multi-Party Dialogues

Kohji Dohsaka
NTT Communication Science Laboratories
NTT Corporation
2-4, Hikaridai, Seika-cho,
Kyoto 619-0237, Japan

Ryota Asai
Graduate School of
Information Science and Technology
Osaka University, 1-1 Yamadaoka,
Suita, Osaka 565-0871, Japan

Ryuichiro Higashinaka and **Yasuhiro Minami** and **Eisaku Maeda**
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Kyoto 619-0237, Japan
{dohsaka,rh,minami,maeda}@cslab.kecl.ntt.co.jp

Abstract

This paper presents an experimental study that analyzes how conversational agents activate human communication in thought-evoking multi-party dialogues between multi-users and multi-agents. A thought-evoking dialogue, which is a kind of interaction in which agents act on user willingness to provoke user thinking, has the potential to stimulate multi-party interaction. In this paper, we focus on quiz-style multi-party dialogues between two users and two agents as an example of a thought-evoking multi-party dialogue. The experiment results showed that the presence of a peer agent significantly improved user satisfaction and increased the number of user utterances. We also found that agent empathic expressions significantly improved user satisfaction, raised user ratings of a peer agent, and increased user utterances. Our findings will be useful for stimulating multi-party communication in various applications such as educational agents and community facilitators.

1 Introduction

Conversational interfaces including dialogue systems and conversational agents have been typically used as a single interface to a single user (Zue et al., 1994; Allen et al., 2001; Cassell et al., 2000). On the other hand, a new area of research in conversational interfaces is dealing with multi-party interaction (Traum and Rickel, 2002; Liu and Chee, 2004; Zheng et al., 2005). Multi-party conversational interfaces have been applied to such tasks as training decision-making in team activities (Traum and Rickel, 2002), collaborative learning (Liu and Chee, 2004), and coordinating and facilitating interaction in a casual social group (Zheng et al., 2005).

The advantage of such multi-party dialogues over two-party cases is that the multi-party case encourages group interaction and collaboration among human users. This advantage can be exploited to foster such human activities as student learning in more social settings and to build and maintain social relationships among people. However, unless users actively engage in the interaction, these multi-party dialogue qualities cannot be adequately exploited. Our objective is to stimulate human communication in multi-party dialogues between multi-users and multi-agents by raising user willingness to engage in the interaction and increasing the number of user utterances.

As the first step toward this objective, we exploit a new style of dialogue called thought-evoking dialogue and experimentally investigate the impact of a peer agent's presence and agent emotional expressions on communication activation in thought-evoking multi-party dialogues. A thought-evoking dialogue, an interaction in which agents act on the willingness of users to provoke user thinking and encourage involvement in the dialogue, has the potential to activate interaction among participants in multi-party dialogues.

Previous work proposed a quiz-style information presentation dialogue system (hereafter quiz-style dialogue system) (Higashinaka et al., 2007a) that is regarded as a kind of thought-evoking dialogue system. This system conveys contents as biographical facts of famous people through quiz-style interaction with users by creating a "Who is this?" quiz and individually presenting hints.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 217–224,
Queen Mary University of London, September 2009.

The hints are automatically created from the biographical facts of people and ordered based on the difficulty naming the people experienced by the users (Higashinaka et al., 2007b). Since the user has to consider the hints to come up with reasonable answers, the system stimulates user thinking. This previous work reported that, for interaction between a single user and a computer, a quiz-style dialogue improved user understanding and willingness to engage in the interaction. In this paper, we focus on a quiz-style information presentation multi-party dialogue (hereafter quiz-style multi-party dialogue) as an example of a thought-evoking multi-party dialogue.

A peer agent acts as a peer of the users and participates in the interactions in the same way that the users do. We are interested in the peer agent's role in quiz-style multi-party dialogues since the positive effects of a peer agent on users have been shown in the educational domain (Chou et al., 2003; Maldonado et al., 2005), which is a promising application area for quiz-style dialogues. In the educational domain, a user could benefit not only from direct communication with a peer agent but also from overhearing dialogues between a peer agent and a tutor. Learning by observing others who are learning is called vicarious learning and positively affects user performance (Craig et al., 2000; Stenning et al., 1999). To the best of our knowledge, detailed experimental investigations on the effect of a peer agent on communication activation have not been reported in multi-party dialogues between multi-users and multi-agents, which are our main concern in this paper.

The topic of emotion has gained widespread attention in human-computer interaction (Bates, 1994; Picard, 1997; Hudlicka, 2003; Prendinger and Ishizuka, 2004). The impact of an agent's emotional behaviors on users has also recently been studied (Brave et al., 2005; Maldonado et al., 2005; Prendinger et al., 2005). However, these previous studies addressed scenario-based interaction in which a user and an agent acted with predetermined timing. In this paper, we investigate the impact of agent emotional expressions on users in multi-party dialogues in which multiple users and agents can make utterances with more flexible timing.

Resembling work by Brave *et al.* (2005), we classify agent emotional expressions into empathic and self-oriented ones and investigate their impact on users in a thought-evoking multi-party dialogue system. As stated above, Brave *et al.* (2005) addressed scenario-based Black-jack interaction, but we deal with multi-party dialogues that enable more flexible turn-taking. Previous studies (Bickmore and Picard, 2005; Higashinaka et al., 2008) showed that agent empathic expressions have a positive psychological impact upon users, but they only examined two-party cases. Although Traum *et al.* (2002) and Gebhard *et al.* (2004) exploited the role of agent emotion in multi-party dialogues, they did not adequately examine the effects of agent emotion on communication activation by experiment.

In this work, we deal with disembodied agents and focus on their linguistic behaviors. We believe that our results are useful for designing embodied conversational agents using other modalities.

This paper presents an experimental study that analyzes how agents stimulate human communication in quiz-style multi-party dialogues between two users and two agents. We are especially interested in how the presence of a peer agent and agent emotional expressions improve user satisfaction, enhance user opinions about the peer agent, and increase the number of user utterances. Our findings will be useful for stimulating human communication in various applications such as educational agents and community facilitators.

In the following, Section 2 shows an overview of our quiz-style multi-party dialogue system. Section 3 explains the experiment design, and Section 4 describes the results. Section 5 concludes the paper.

2 Thought-Evoking Multi-Party Dialogue System

We implemented a quiz-style multi-party dialogue system between multi-users and multi-agents. The system is a Japanese keyboard-based dialogue system with a chat-like interface. The users can make utterances any time they want. A user utterance is completed and displayed on the chat window when the Enter key is pressed.

Our experiment dealt with cases where two users and two agents engaged in a dialogue. The two agents are a quizmaster and a peer. The quizmaster agent creates a "Who is this?" quiz about a famous person and presents hints one by one to the users and the peer agent who guess the correct answer.

1	*Whowho*	Who is this? First hint: Graduated from the University of Tokyo.
2	Mary	I don't know.
3	*Kinoko*	Yoshida Shigeru.
4	*Whowho*	No, not even close!
5	Jane	That's very difficult.
6	*Kinoko*	Difficult for me, too.
7	*Whowho*	Second hint: Born in Ushigome, Edo.
	...	
8	*Whowho*	Third hint: Novelist and scholar of British literature.
9	Mary	Murakami Haruki.
10	*Whowho*	Close!
11	*Kinoko*	You are close. Excellent.
12	Jane	Well then, who is it?
13	*Whowho*	Fourth hint: Familiar with Haiku, Chinese poetry, and calligraphy.
14	Mary	Natsume Soseki.
15	*Whowho*	That's right. Wonderful.
16	*Kinoko*	Mary, excellent. I'm happy for you.
17	Jane	Mary, that's the right answer. Good job.

Figure 1: Sample dialogue

Figure 1 shows a sample dialogue. Mary and Jane are human users. *Whowho* is the quizmaster agent, and *Kinoko* is the peer agent. Quizmaster agent *Whowho* presents hints in lines 1, 7, 8, and 13. Users Mary and Jane and peer agent *Kinoko* give answers in lines 3, 9, and 14.

The hints were automatically created using biographical facts (in Japanese) of people in Wikipedia [1] based on a previously reported method (Higashinaka et al., 2007b).

2.1 Dialogue acts

The users and the two agents perform several dialogue acts based on the dialogue context.

Present-hint: The quizmaster agent presents hints one by one (lines 1, 7, 8, and 13) in the sample dialogue shown in Figure 1.

Give-ans: Users and the peer agent give answers (lines 3, 9, and 14).

Show-difficulty: Users and the peer agent offer opinions about the quiz difficulty (lines 2, 5, 6, and 12).

[1] http://ja.wikipedia.org/

Evaluate-ans: When the answer is wrong, the quizmaster agent evaluates the answer based on the person-name similarity score (Higashinaka et al., 2007a) and utters "very close!," "close!," "a little close!," "a little far," "far," or "not even close!" (lines 4 and 10).

Complete-quiz-with-success: When the right answer is given, the quizmaster agent informs the dialogue participants that the current quiz is completed (line 15).

Complete-quiz-with-failure: If all hints have been generated and no right answer is given, the quizmaster agent gives the right answer, and the current quiz is completed.

Feedback-on-wrong-ans: Users and the peer agent give feedback when their own or the other's answers are wrong during the current quiz (line 11).

Feedback-on-success: Users and the peer agent give feedback when their own or the other's answers are right and the current quiz session is completed (lines 16 and 17).

Feedback-on-failure: Users and the peer agent give feedback when the current quiz is completed without the right answer.

Address-hearer: Users and the two agents specify an intended addressee by uttering the other's name (lines 16 and 17).

When a user utterance is input, the system separates it into word tokens using a Japanese morphological analyzer and converts it into dialogue acts using hand-crafted grammar. The system can recognize 120,000 proper names of persons.

2.2 Utterance generation

Surface realization forms were prepared for each dialogue act by the agents. Agent utterances are generated by randomly selecting one of the forms.

Some agent dialogue acts can be generated with emotional expressions. Agent emotional expressions are categorized into empathic and self-oriented ones (Brave et al., 2005). The agent self-oriented emotional expressions (self-oriented expressions) are oriented to their own state, and the agent empathic expressions are oriented to the other's state and are congruent with the other's

Dialog act	Emotion	Expressions
Show-difficulty	EMP	Difficult for me, too.
Show-difficulty	SELF	I don't remember. That's so frustrating.
Show-difficulty	NONE	I don't know.
Feedback-on-success	EMP	You're right. I'm happy for you.
Feedback-on-success	SELF	I'm really glad I got the correct answer.
Feedback-on-success	NONE	You're right / I'm right.
Feedback-on-failure	EMP	Too bad you didn't know the right answer.
Feedback-on-failure	SELF	I'm disappointed that I didn't know the right answer.
Feedback-on-failure	NONE	I/You didn't know the right answer.

Table 1: Examples of agent expressions. EMP shows empathic expressions, SELF shows self-oriented expressions, and NONE shows neutral expressions when neither emotion is present.

welfare. As explained in 3.1, we prepared different experimental conditions to determine the presence/absence of agent empathic and self-oriented expressions. Based on the conditions, we controlled the agent emotional expressions. Table 1 shows examples of agent empathic, self-oriented, and neutral expressions.

2.3 Dialogue management

The system maintains a dialogue state in which the history of the participant's dialogue acts is recorded with the time of each act. We prepared preconditions of each dialogue act by the agents. For example, the quizmaster agent's *Evaluate-ans* can be executed after the users or the peer agent provides a wrong answer. The peer agent's *Feedback-on-success* can be executed after the quizmaster agent performs Complete-quiz-with-success. We also used the following turn-taking rules:

1. Either agent must talk when neither the users nor the agents make utterances within a given time (4 sec.).

Condition	Peer agent	Empathic	Self-oriented
(0)	Absent	Absent	Absent
(1)	Present	Absent	Absent
(2)	Present	Present	Absent
(3)	Present	Absent	Present
(4)	Present	Present	Present

Table 2: Experimental conditions based on presence/absence of peer agent and agent empathic and self-oriented expressions

2. Agents must not talk for a given time (0.5 sec.) after the others talk.

3. The quizmaster agent must move to the next hint when neither the users nor the peer agent give a correct answer within a given time (30 sec.).

Based on the dialogue state, the preconditions of the dialogue acts and the turn-taking rules, the system chooses the next speaker and its dialogue act.

3 Experiment

3.1 Experimental conditions

To evaluate the effects of the presence of the peer agent and the agent emotional expressions, we prepared five systems under different experimental conditions, (0), (1), (2), (3), and (4), based on the presence/absence of the peer agent and agent empathic and self-oriented expressions. They are shown in Table 2. In condition (0), the peer agent was absent, and only the quizmaster agent was present. In other conditions, both the quizmaster and peer agents were present. In conditions (0) and (1), neither empathic nor self-oriented expressions were exhibited. In condition (2), only empathic expressions were exhibited. In condition (3), only self-oriented expressions were exhibited. In condition (4), both empathic and self-oriented expressions were exhibited.

We evaluated the effects of the presence of the peer agent by comparing conditions (0) and (1). We evaluated the effects of agent empathic and self-oriented expressions by comparing conditions (1), (2), (3), and (4).

3.2 Measures

We used three measures: user satisfaction, user opinions about the peer agent, and the number of

	Questionnaire items
Q1	Did you want to converse with this system again? (Willingness to engage in dialogue)
Q2	Was the dialogue enjoyable? (Pleasantness of dialogue)
Q3	Did you feel satisfied using the dialogue system? (Satisfaction of system usage)
Q4	Was the peer agent friendly? (Agent's closeness)
Q5	Did you feel that the peer agent cared about you? (Agent's caring)
Q6	Was the peer agent likable? (Agent's likability)
Q7	Did the peer agent support you? (Agent's support)

Table 3: Questionnaire items to evaluate user satisfaction (Q1, Q2, and Q3) and user opinions about the peer agent (Q4, Q5, Q6, and Q7)

user utterances. Among these measures, we regarded the number of user utterances as an objective measure to evaluate communication activation. User satisfaction and opinions about the peer agent are subjective measures based on the questionnaires (ten-point Likert scale). Table 3 shows the questionnaires used in the experiment. We expected that a high level of user satisfaction and positive opinions about the peer agent would lead to a high level of user engagement, which would promote user utterances.

User satisfaction was evaluated from different perspectives with three questions: Q1, Q2, and Q3. Q1 focused on user willingness to engage in the dialogue; Q2 focused on the user experience of the dialogue's pleasantness; Q3 focused on user satisfaction with the system. We evaluated user satisfaction with averages of the ratings of Q1, Q2, and Q3. Using the averaged ratings of Likert questions allows us to apply such parametric statistical tests as a multi-factor ANOVA since the summed or averaged responses to Likert questions tend to follow a normal distribution.

User opinions about the peer agent were evaluated in terms of how the user perceived the peer agent's closeness (Q4), its caring (Q5), its likability (Q6), and its support (Q7). We evaluated user opinions about the peer agent with the averaged ratings of these items. Previous studies showed that empathic behaviors exhibited by an agent improved user opinions about the agent in a Blackjack scenario (Brave et al., 2005) and in a social dialogue between a single user and an agent (Higashinaka et al., 2008). We examined these items in multi-party dialogues with flexible turn-taking.

3.3 Procedure

We recruited and paid 64 Japanese adults (32 males and 32 females) for their participation. The mean ages of the male and female groups were 32.0 and 36.2, respectively (male group: SD=9.2 , min=22, max=59, female group: SD=9.6, min=20, max=50). The participants were divided into 32 pairs of the same gender: 16 pairs of males and 16 pairs of females. The participants in each pair were unacquainted.

The experiment had a within-participants design. Each pair of participants successively engaged in dialogues using the five systems under different experimental conditions. The order of using the systems was counter-balanced to prevent order effect.

Before starting the experiment, the participants were informed that, after completing a dialogue with each system, they would fill out questionnaires. The questionnaires on user opinions about the peer agent were used only when it was present (conditions (1), (2), (3), and (4)). The participants were also told that the agents were computer programs and not human participants. During the experiment, each pair of participants was seated in separate rooms in front of a computer display, a keyboard, and a mouse, and they could only communicate with each other through the system.

In the dialogue with each system, five "Who is this?" quizzes about famous people were presented. The quiz subjects were chosen so that the difficulty level of the quizzes was approximately the same in all the systems. For this purpose, we first sorted people in Wikipedia in descending order by their PageRank ^{TM} score based on Wikipedia's hyper-link structure. We then extracted the top-50 people and divided them from the top into five groups of 10. Next we randomly selected five people from each group to make five sets of five people of approximately identical PageRank scores. Each set of five people was used for quizzes in each system.

On average, a pair of participants took 18 minutes to complete a dialogue with each system. The number of hints that were actually presented in a

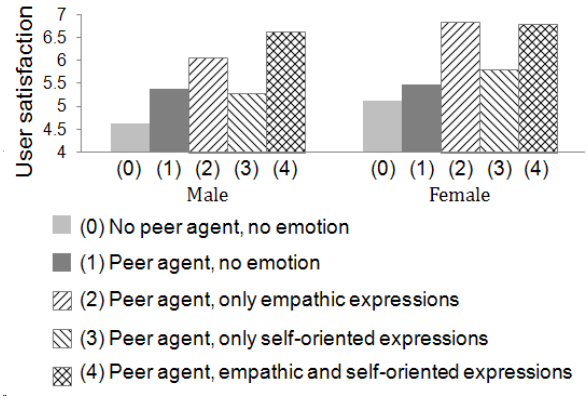


Figure 2: User satisfaction

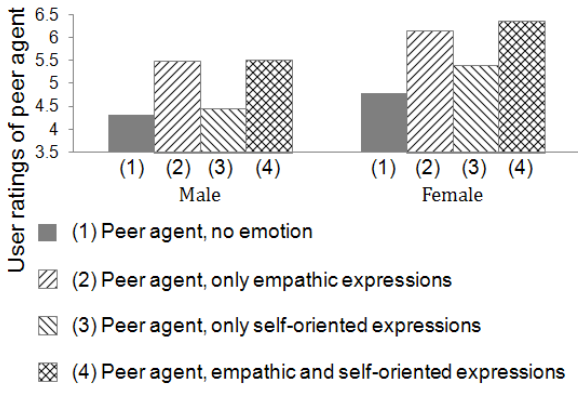


Figure 3: User ratings of peer agent

quiz averaged 7.5.

4 Results

4.1 User satisfaction

For questions Q1, Q2, and Q3, Cronbach's alpha was 0.83, which justified combining these items into a single index. Therefore we evaluated user satisfaction with averages of the ratings of these items. Figure 2 shows user satisfaction under each experimental condition.

To evaluate the effect of the peer agent's presence on user satisfaction, we compared conditions (0) and (1). The F-test results showed that variances were assumed to be equal across groups ($p > 0.2$), and the Kolmogorov-Smirnov test results showed that the assumption of normality was satisfied ($p > 0.6$). By applying the paired t-test to both the male and female groups, we found that the peer agent's presence significantly improved user satisfaction (male group: $t(31) = 4.2, p < 0.001$, female group: $t(31) = 2.8, p < 0.008$).

To evaluate the effect of the empathic and self-oriented expressions exhibited by the agents on user satisfaction, we compared conditions (1), (2), (3), and (4). A three-factor ANOVA was conducted with two within-participant factors of empathic and self-oriented expressions and one between-participant factor of gender. The F-test for the homogeneity of variances ($p > 0.1$) and the Kolmogorov-Smirnov normality test ($p > 0.1$) showed that the data met the ANOVA assumptions. As a result of the ANOVA, a significant main effect was found for empathic expressions with respect to user satisfaction, $F(1, 62) = 92.7, p < 0.001$. No significant main effects were found for either self-oriented expressions or gender, and there were no significant interactions.

These results showed that the peer agent's presence and the agent empathic expressions significantly improved user satisfaction in quiz-style multi-party dialogues.

4.2 User opinions about the peer agent

For questions Q4, Q5, Q6, and Q7, Cronbach's alpha was 0.92, which justified combining these items into a single index. Therefore we evaluated user opinions about the peer agent with the averaged ratings of these items under each experimental condition. Figure 3 shows the user ratings of the peer agent under each condition.

To evaluate the effect of agent empathic and self-oriented expressions on the user ratings of the peer agent, we compared conditions (1), (2), (3) and (4). A three-factor ANOVA was conducted with two within-participant factors of empathic and self-oriented expressions and one between-participant factor of gender. The F-test for the homogeneity of variances ($p > 0.3$) and the Kolmogorov-Smirnov normality test ($p > 0.2$) showed that the data met the ANOVA assumptions. As a result of the ANOVA, a significant main effect was found for empathic expressions with respect to the user ratings of the peer agent, $F(1, 62) = 77.4, p < 0.001$. There was a moderate main effect for self-oriented expressions with respect to the user ratings of the peer agent, $F(1, 62) = 4.38, p < 0.04$. There were no significant main effects for gender, and there were no significant interactions.

These results showed that agent empathic expressions significantly improved user ratings of the peer agent in quiz-style multi-party dialogues.

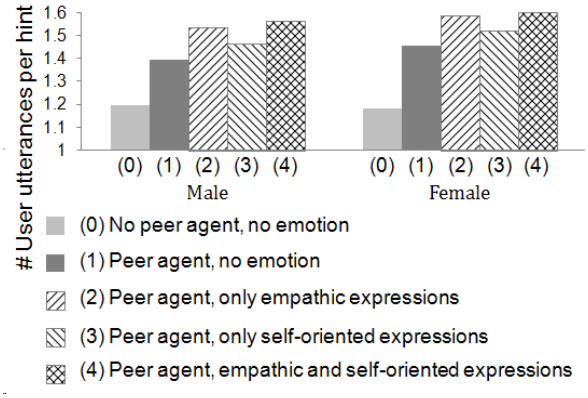


Figure 4: User utterances per quiz hint

4.3 Number of user utterances

Figure 4 shows the number of user utterances per quiz hint under each condition.

To evaluate the effect of the peer agent's presence on the number of user utterances per quiz hint, we compared conditions (0) and (1). Based on the F-test and the Kolmogorov-Smirnov test, the assumptions of variance homogeneity ($p > 0.6$) and normality ($p > 0.5$) were met. By applying the paired t-test to both the male and female groups, we found that the presence of the peer agent significantly increased the number of user utterances per hint (male group: $t(31) = 3.1, p < 0.004$, female group: $t(31) = 5.6, p < 0.001$).

To evaluate the effect of empathic and self-oriented expressions by agents on the number of user utterances, we compared conditions (1), (2), (3), and (4). A three-factor ANOVA was conducted with two within-participant factors of empathic and self-oriented expressions and one between-participant factor of gender. The F-test for the homogeneity of variances ($p > 0.05$) and the Kolmogorov-Smirnov normality test ($p > 0.6$) showed that the data met the ANOVA assumptions. As a result of the ANOVA, a significant main effect was found for empathic expressions with respect to the number of user utterances, $F(1, 62) = 18.9, p < 0.001$. No significant main effects were found for either self-oriented expressions or gender, and there were no significant interactions.

These results showed that the peer agent's presence and agent empathic expressions increased the number of user utterances and stimulated human communication in quiz-style multi-party dialogues.

5 Conclusion

This paper experimentally analyzed how conversational agents stimulate human communication in thought-evoking multi-party dialogues between multi-users and multi-agents. As an example of such multi-party dialogue, we focused on quiz-style multi-party dialogues between two users and two agents. We investigated how a peer agent's presence and agent emotional expressions influenced user satisfaction, the user ratings of the peer agent, and the number of user utterances. The user ratings of the peer agent included user's perceived closeness, likability and caring from the peer agent, and the user's feeling of being supported by the peer agent.

The experiment results showed that the peer agent's presence significantly improved user satisfaction and increased the number of user utterances. We also found significant effects that agent empathic expressions improved user satisfaction and user positive ratings of the peer agent and that they further increased the number of user utterances. These results indicate that employing a peer agent and agent empathic behaviors in thought-evoking multi-party dialogues will stimulate interaction among people in computer-mediated communication. Our findings will be useful for a broader class of applications such as educational agents and community facilitators.

Many directions for future work remain. First, we plan to extend our work to deal with various modalities such as speech, gestures, body posture, facial expressions, and the direction of eye gazes to investigate the effects of agent representation (embodied or disembodied) and other modalities in thought-evoking multi-party dialogues. Second, we will analyze how agent behaviors influence users and dialogues in more detail and develop a more sophisticated dialogue management method based on our detailed analysis. Learning optimal dialogue management strategies in multi-party dialogues is a challenging research topic. Third, examining the relationship between user personality traits and the impact of agents on users is valuable. Previous work reported that the effect of embodiment depended on user personalities (Lee et al., 2006). This direction is important to the stimulation of multi-party interaction for therapeutic and emotional support.

References

James Allen, Donna Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. 2001. Toward conversational human-computer interaction. *AI Magazine*, 22(4):27–37.

Joseph Bates. 1994. The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125.

Timothy W. Bickmore and Rosalind W. Picard. 2005. Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction*, 12(2):293–327.

Scott Brave, Clifford Nass, and Kevin Hutchinson. 2005. Computers that care: investigating the effects of orientation of emotion exhibited by an embodied computer agent. *International Journal of Human-Computer Studies*, 62(2):161–178.

Justine Cassell, Joseph Sullivan, Scott Prevost, and Elizabeth Churchill, editors. 2000. *Embodied Conversational Agents*. MIT Press, Cambridge, MA.

Chih-Yueh Chou, Tak-Wai Chan, and Chi-Jen Lin. 2003. Redefining the learning companion: the past, present, and future of educational agents. *Computers & Education*, 40(3):255–269.

Scotty D. Craig, Barry Gholson, Matthew Ventura, Arthur C. Graesser, and the Tutoring Research Group. 2000. Overhearing dialogues and monologues in virtual tutoring sessions: Effects on questioning and vicarious learning. *International Journal of Artificial Intelligence in Education*, 11:242–253.

Patrick Gebhard, Martin Klesen, and Thomas Rist. 2004. Coloring multi-character conversations through the expression of emotions. In *Lecture Notes in Computer Science (Tutorial and Research Workshop on Affective Dialogue Systems)*, volume 3068, pages 128–141.

Ryuichiro Higashinaka, Kohji Dohsaka, Shigeaki Amano, and Hideki Isozaki. 2007a. Effects of quiz-style information presentation on user understanding. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association*, pages 2725–2728.

Ryuichiro Higashinaka, Kohji Dohsaka, and Hideki Isozaki. 2007b. Learning to rank definitions to generate quizzes for interactive information presentation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (Poster Presentation)*, pages 117–120.

Ryuichiro Higashinaka, Kohji Dohsaka, and Hideki Isozaki. 2008. Effects of self-disclosure and empathy in human-computer dialogue. In *Proceedings of 2008 IEEE Workshop on Spoken Language Technology*, pages 109–112.

Eva Hudlicka. 2003. To feel or not to feel: The role of affect in human-computer interaction. *International Journal of Human-Computer Studies*, 59(1-2):1–32.

Kwan Min Lee, Younbo Jung, Jaywoo Kim, and Sang Ryong Kim. 2006. Are physically embodied social agents better than disembodied social agents?: Effects of embodiment, tactile interaction, and people's loneliness in human-robot interaction. *International Journal of Human-Computer Studies*, 64(10):962–973.

Yi Liu and Yam San Chee. 2004. Intelligent pedagogical agents with multiparty interaction support. In *Proceedings of International Conference on Intelligent Agent Technology*, pages 134–140.

Heidy Maldonado, Jong-Eun Roselyn Lee, Scott Brave, Cliff Nass, Hiroshi Nakajima, Ryota Yamada, Kimihiko Iwamura, and Yasunori Morishima. 2005. We learn better together: enhancing elearning with emotional characters. In *Proceedings of the 2005 Conference on Computer Support for Collaborative Learning*, pages 408–417.

Rosalind W. Picard. 1997. *Affective Computing*. MIT Press, Cambridge, MA.

Helmut Prendinger and Mitsuru Ishizuka, editors. 2004. *Life-Like Characters: Tools, Affective Functions, and Applications*. Springer, Berlin.

Helmut Prendinger, Junichiro Mori, and Mitsuru Ishizuka. 2005. Using human physiology to evaluate subtle expressivity of a virtual quizmaster in a mathematical game. *International Journal of Human-Computer Studies*, 62(2):231–245.

Keith Stenning, Jean McKendree, John Lee, Richard Cox, Finbar Dineen, and Terry Mayes. 1999. Vicarious learning from educational dialogue. In *Proceedings of the 1999 Conference on Computer Support for Collaborative Learning*, pages 341–347.

David Traum and Jeff Rickel. 2002. Embodied agents for multi-party dialogue in immersive virtual worlds. In *Proceedings of the 1st International Joint Conference on. Autonomous Agents and Multi-Agent Systems*, pages 766–773.

Jun Zheng, Xiang Yuan, and Yam San Chee. 2005. Designing multiparty interaction support in Elva, an embodied tour guide. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 929–936.

Victor Zue, Stephanie Seneff, Joseph Polifroni, Michael Phillips, Christine Pao, David Goodine, David Goddeau, and James Glass. 1994. PEGASUS: a spoken dialogue interface for on-line air travel planning. *Speech Communication*, 15:331–340.

Models for Multiparty Engagement in Open-World Dialog

Dan Bohus
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
dbohus@microsoft.com

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
horvitz@microsoft.com

Abstract

We present computational models that allow spoken dialog systems to handle multi-participant engagement in open, dynamic environments, where multiple people may enter and leave conversations, and interact with the system and with others in a natural manner. The models for managing the engagement process include components for (1) sensing the engagement state, actions and intentions of multiple agents in the scene, (2) making engagement decisions (*i.e.* whom to engage with, and when) and (3) rendering these decisions in a set of coordinated low-level behaviors in an embodied conversational agent. We review results from a study of interactions "in the wild" with a system that implements such a model.

1 Introduction

To date, nearly all spoken dialog systems research has focused on the challenge of engaging single users on tasks defined within a relatively narrow context. Efforts in this realm have led to significant progress including large-scale deployments that now make spoken dialog systems common features in the daily lives of millions of people. However, research on dialog systems has largely overlooked important challenges with the initiation, maintenance, and suspension of conversations that are common in the course of natural communication and collaborations among people. In (Bohus and Horvitz, 2009) we outlined a set of core challenges for extending traditional *closed-world* dialog systems to systems that have competency in *open-world dialog*. The work described here is part of a larger research effort aimed at addressing these challenges, and constructing computational models to support the core interaction skills required for open-world dialog. In particular, we focus our attention in this paper on the challenges of managing *engagement* – "the process by which two (or more) participants establish, maintain and end their perceived connection during interactions they jointly undertake", *cf.* Sidner et al. (2004) in open-world settings.

We begin by reviewing the challenges of managing engagement in the open-world in the next section. In Section 3, we survey the terrain of related efforts that provides valuable context for the new work described in this paper. In Section 4, we introduce a computational model for multiparty situated engagement. The model harnesses components for sensing the engagement state, actions, and intentions of people in the scene for making high-level engagement decisions (whom to engage with, and when), and for rendering these decisions into a set of low-level coordinated behaviors (*e.g.*, gestures, eye gaze, greetings, etc.). Then, we describe an initial observational study with the proposed model, and discuss some of the lessons learned through this experiment. Finally, in Section 6, we summarize this work and outline several directions for future research.

2 Engagement in Open-World Dialog

In traditional, single-user systems the engagement problem can often be resolved in a relatively simple manner. For instance, in telephony-based applications, it is typically safe to assume that a user is engaged with a dialog system once a call has been received. Similarly, push-to-talk buttons are often used in multimodal mobile applications. Although these solutions are sufficient and even natural in closed, single-user contexts, they become inappropriate for open-world systems that must operate continuously in open, dynamic environments, such as robots, interactive billboards, or embodied conversational agents.

Interaction in the open-world is characterized by two aspects that capture key departures from assumptions traditionally made in spoken dialog systems (Bohus and Horvitz, 2009). The first one is the *dynamic, multiparty* nature of the interaction, *i.e.*, the world typically contains not just one, but multiple agents that are relevant

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 225–234,
Queen Mary University of London, September 2009.

to the interactive system. Engagements in open worlds are often dynamic and asynchronous, *i.e.* relevant agents may enter and leave the observable world at any time, may interact with the system and with each other, and their goals, needs, and intentions may change over time. Managing the engagement process in this context requires that a system explicitly represents, models, and reasons about multiple agents and interaction contexts, and maintains and leverages long-term memory of the interactions to provide support and assistance.

A second important aspect that distinguishes open-world from closed-world dialog is the *situated* nature of the interaction, *i.e.,* the fact that the surrounding physical environment provides rich, streaming context that is relevant for conducting and organizing the interactions. Situated interactions among people often hinge on shared information about physical details and relationships, including structures, geometric relationships and pathways, objects, topologies, and communication affordances. The often implicit, yet powerful physicality of situated interaction, provides opportunities for making inferences in open-world dialog systems, and challenges system designers to innovate across a spectrum of complexity and sophistication. Physicality and embodiment also provide important affordances that can be used by a system to support the engagement process. For instance, the use of a rendered or physically embodied avatar in a spoken dialog system provides a natural point of visual engagement between the system and people, and allows the system to employ natural signaling about attention and engagement with head pose, gaze, facial expressions, pointing and gesturing.

We present in this paper methods that move beyond the realm of closed-world dialog with a *situated multiparty engagement model* that can enable a computational system to fluidly engage, disengage and re-engage one or multiple people, and support natural interactions in an open-world context.

3 Related Work

The process of engagement between people, and between people and computational systems has received a fair amount of attention. Observational studies in the sociolinguistics and conversational analysis communities have revealed that engagement is a complex, mixed-initiative, highly-coordinated process that often involves a variety of non-verbal cues and signals, (Goffman, 1963; Kendon, 1990), spatial trajectory and proximity (Hall, 1966; Kendon, 1990b), gaze and mutual attention (Argyle and Cook, 1976), head and hand gestures (Kendon, 1990), as well as verbal greetings.

A number of researchers have also investigated issues of engagement in human-computer and human-robot interaction contexts. Sidner and colleagues (2004) define engagement as "the process by which two (or more) participants establish, maintain and end their perceived connection during interactions they jointly undertake", and focus on the process of maintaining engagement. They show in a user study (Sidner et al., 2004; 2005) that people directed their attention to a robot more often when the robot made engagement gestures throughout the interaction (*i.e.* tracked the user's face, and pointed to relevant objects at appropriate times in the conversation.) Peters (2005; 2005b) uses an alternative definition of engagement as "the value that a participant in an interaction attributes to the goal of being together with the other participant(s) and of continuing the interaction," and present the high-level schematics for an algorithm for establishing and maintaining engagement. The algorithm highlights the importance of mutual attention and eye gaze and relies on a heuristically computed "interest level" to decide when to start a conversation. Michalowski and colleagues (2006) propose and conduct experiments with a model of engagement grounded in proxemics (Hall, 1966) which classifies relevant agents in the scene in four different categories (*present, attending, engaged* and *interacting*) based on their distance to the robot. The robot's behaviors are in turn conditioned on the four categories above.

In our work, we follow Sidner's definition of engagement as a process (Sidner et al., 2004) and describe a computational model for *situated multiparty engagement*. The proposed model draws on several ideas from the existing body of work, but moves beyond it and provides a more comprehensive framework for managing the engagement process in a dynamic, open-world context, where multiple people with different and changing goals may enter and leave, and communicate and coordinate with each other and with the system.

4 Models for Multiparty Engagement

The proposed framework for managing engagement is centered on a reified notion of *interaction*, defined here as a basic unit of sustained, interactive problem-solving. Each interaction involves two or more participants, and this number may vary in time; new participants may join an existing interaction, or current participants may leave an interaction at any point in time. The system is actively engaged in at most one interaction at a time (with one or multiple participants), but it can simultaneously keep track of additional, suspended interactions. In this context, engagement is viewed as the process subsuming the joint, coordinated activities by which participants *initiate, maintain, join, abandon, suspend, resume,* or *terminate* an interaction. Appendix A shows by means of an example the various stages of an interaction and the role played by the engagement process.

Successfully modeling the engagement process in a situated, multi-participant context requires that the system (1) senses and reasons about the engagement state,

actions and intentions of multiple agents in the scene, (2) makes high-level engagement control decisions (*i.e.* about whom to engage or disengage with, and when) and (3) executes and signals these decisions to the other participants in an appropriate and expected manner (*e.g.* renders them in a set of coordinated behaviors such as gestures, greetings, etc.). The proposed model subsumes these three components, which we discuss in more detail in the following subsections.

4.1 Engagement State, Actions, Intentions

As a prerequisite for making informed engagement decisions, a system must be able to recognize various engagement cues, and to reason about the engagement actions and intentions of relevant agents in the scene. To accomplish this, the sensing subcomponent of the proposed engagement model tracks over time three related engagement variables for each agent a and interaction i: the engagement state $ES_a^i(t)$, the engagement action $EA_a^i(t)$ and the engagement intention $EI_a^i(t)$.

The engagement state, $ES_a^i(t)$, captures whether an agent a is engaged in interaction i and is modeled as a deterministic variable with two possible values: *engaged* and *not-engaged*. The state is updated based on the joint actions of the agent and the system (see Figures 3 and 4). Since engagement is a collaborative process, the transitions to the *engaged* state require that both the agent and the system take either an *engage* action (if the agent was previously not engaged) or a *maintain* action (if the agent was already engaged); we discuss these actions in more detail shortly. On the other hand, disengagement can be a unilateral act: an agent transitions to the *not-engaged* state if either the agent or the system take a *disengage* action or a *no-action*.

The second engagement variable, $EA_a^i(t)$, models the actions that an agent takes to initiate, maintain or terminate engagement. There are four engagement actions: *engage*, *no-action*, *maintain*, *disengage*. The first two are possible only from the *not-engaged* state, while the last two are possible only from the *engaged* state. The engagement actions are estimated based on a conditional probabilistic model of the form:

$$P(EA_a^i(t)|ES_a^i(t), EA_a^i(t-1), SEA_a^i(t-1), \Psi(t))$$

The inference is conditioned on the current engagement state, on the previous agent and system actions, and on additional sensory evidence $\Psi(t)$. $\Psi(t)$ includes the detection of explicit engagement cues such as: salutations (*e.g.* "Hi!", "Bye bye"); calling behaviors (*e.g.* "Laura!"); the establishment or the breaking of an F-formation (Kendon, 1990b), *i.e.* the agent approaches and positions himself in front of the system and attends to the system; an expected, opening dialog move (*e.g.* "Come here!"). Note that each of these cues is explicit, and marks a committed engagement action.

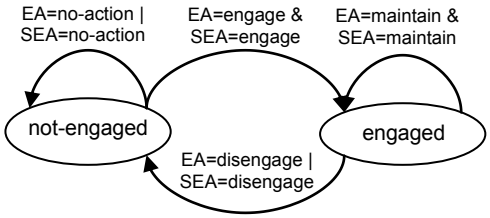


Figure 2. Engagement state transition diagram. EA is the agent's engagement action; SEA is the system's action.

A third variable in the proposed model, $EI_a^i(t)$, tracks the engagement intention of an agent with respect to a conversation. Like the engagement state, the intention can either be *engaged* or *not-engaged*. Intentions are tracked separately from actions since an agent might intend to engage or disengage the system, but not yet take an explicit engagement action. For instance, let us consider the case in which the system is already engaged in an interaction and another agent is waiting in line to interact with the system. Although the waiting agent does not take an explicit, committed engagement action, she might still intend to engage in a new conversation with the system once the opportunity arises. She might also signal this engagement intention via various cues (*e.g.* pacing around, glances that make brief but clear eye contact with the system, etc.) More generally, the engagement intention variable captures whether or not an agent would respond positively should the system initiate engagement. In that sense, it roughly corresponds to Peters' (2005; 2005b) "interest level", *i.e.* to the value the agent attaches to being engaged in a conversation with the system.

Like engagement actions, engagement intentions are inferred based on a direct conditional model:

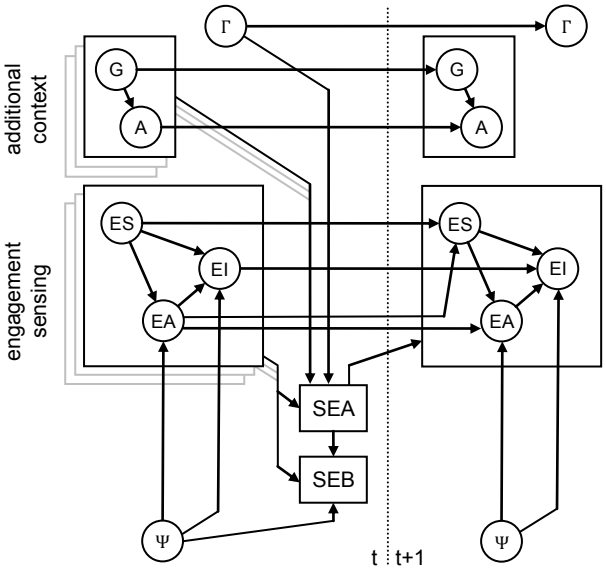


Figure 3. Graphical model showing key variables and dependencies in managing engagement.

$$P(EI_a^i(t)|ES_a^i(t), EA_a^i(t), SEA_a^i(t-1), EI_a^i(\mathrm{t}-1), \Psi(t))$$

This model leverages information about the current engagement state, the previous agent and system actions, the previous engagement intention, as well as additional evidence $\Psi(t)$ capturing implicit engagement cues. Such cues include the spatiotemporal trajectory of the participant and the level of sustained mutual attention. The models for inferring engagement actions and intentions are generally independent of the application. They capture the typical behaviors and cues by which people signal engagement, and, as such, should be reusable across different domains. In other work (Bohus and Horvitz, 2009b), we describe these models in more detail and show how they can be learned automatically from interaction data.

4.2 Engagement Control Policy

Based on the inferred state, actions and intentions of the agents in the scene, as well as other additional evidence to be discussed shortly, the proposed model outputs high-level engagement actions, denoted by SEA decision node in Figure 3. The action-space on the system side contains the same four actions previously discussed: *engage*, *disengage*, *maintain* and *no-action*. Each action is parameterized with a set of agents $\{a_k\}$ and an interaction i. Additional parameters that control the lower level execution of these actions, such as specific greetings, waiting times, urgency, etc. may also be specified. The actual execution mechanisms are discussed in more detail in the following subsection.

In making engagement decisions in an open-world setting, a conversational system must balance the goals and needs of multiple agents in the scene and resolve various tradeoffs (for instance between continuing the current interaction or interrupting it temporarily to address another agent), all the while observing rules of social etiquette in interaction. Apart from the detected engagement state, actions and intentions of an agent $\boldsymbol{E}_a^i = \langle ES_a^i, EA_a^i, EI_a^i \rangle$, the control policy can be enhanced through leveraging additional observational evidence, including high-level information $\boldsymbol{H}_a$ about the various agents in the scene, such as their long-term goals and activities, as well as other global context ($\boldsymbol{\Gamma}$), including the multiple tasks at hand, the history of the interactions, relationships between various agents in the scene (*e.g.* which agents are in a group together), etc. For instance, a system might decide to temporarily refuse engagement even though an agent takes an *engage* action, because it is currently involved in a higher priority interaction. Or, a system might try to take the initiative and engage an agent based on the current context (*e.g.* the system has a message to deliver) and activity of the agent (*e.g.* the agent is passing by), even though the agent has no intention to engage.

Engagement control policies have therefore the form,

$$\pi_{SEA}(\{\boldsymbol{E}_a^i\}_{a,i}, \{\boldsymbol{H}_a\}_a, \boldsymbol{\Gamma})$$

where we have omitted the time index for simplicity. In contrast to the models for inferring engagement intentions and action, the engagement control policy can often be application specific. Such policies can be authored manually to capture the desired system behavior. We will discuss a concrete example of this in Section 5.2. In certain contexts, a more principled solution can be developed by casting the control of engagement as an optimization problem for scheduling collaborations with multiple parties under uncertainties about the estimated goals and needs, the duration of the interactions, time and frustration costs, social etiquette, etc. We are currently exploring such models, where the system also uses information-gathering actions (*e.g.* "Are the two of you together?" "Are you here for X?," etc.), based on value-of-information computations to optimize in the nature and flow of attention and collaboration in multiparty interactions.

4.3 Behavioral Control Policy

At the lower level, the engagement decisions taken by the system have to be executed and rendered in an appropriate manner. With the use of a rendered or physical embodied agent, these actions are translated into a set of coordinated lower-level behaviors, such as head gestures, making and breaking eye contact, facial expressions, salutations, interjections, etc. The coordination of these behaviors is governed by a behavioral control policy, conditioned on the estimated engagement state, actions and intentions of the considered agents, as well as other information extracted from the scene:

$$\pi_{SEB}(SEA, \{\boldsymbol{E}_a^i\}_{a,i}, \Psi)$$

For example, in the current implementation, the *engage* system action subsumes three sub-behaviors performed in a sequence: *EstablishAttention*, *Greeting*, and *Monitor*. First, the system attempts to establish sustained mutual attention with the agent(s) to be engaged. This is accomplished by directing the gaze towards the agents, and if the agent's focus of attention is not on the system, triggering an interjection like "Excuse me!" Once mutual attention is established, on optional *Greeting* behavior is performed; a greeting can be specified as an execution parameter of the *engage* action. Finally, the system enters a *Monitor* behavior, in which it monitors for the completion of engagement. The action completes successfully once the agent(s) are in an engaged state. Alternatively if a certain period of time elapses and the agent(s) have not yet transitioned to the engaged state, the *engage* system action completes with failure (which is signaled to the engagement control layer).

Like the high-level engagement control policies, the behavioral control policies can either be authored manually, or learned from data, either in a supervised (*e.g.*

from a human-human interaction corpus) or unsupervised learning setting. Also, like the engagement sensing component, the behavioral control component is decoupled from the task at hand, and should be largely reusable across multiple application domains.

5 Observational Study

As an initial step towards evaluating the proposed situated multiparty engagement models, we conducted a preliminary observational study with a spoken dialog system that implements these models. The goals of this study were (1) to investigate whether a system can use the proposed engagement models to effectively create and conduct multiparty interactions in an open-world setting, (2) to study user behavior and responses in this setting, and (3) to identify some of the key technical challenges in supporting multiparty engagement and dialog in open-world context. In this section, we describe this study and report on the lessons learned.

5.1 Experimental platform

Studying multiparty engagement and more generally open-world interaction poses significant challenges. Controlled laboratory studies are by their very nature closed-world. Furthermore, providing participants with instructions, such as "Go interact with this system", or "Go join the existing interaction" can significantly prime and alter the engagement behaviors they would otherwise display upon encountering the system in an unconstrained setting. This can in turn cast serious doubts on the validity of the results. Open-world interaction is best observed in the open-world.

To provide an ecologically valid basis for studying situated, multiparty engagement we therefore developed a conversational agent that implements the proposed model, and deployed it in the real-world. The system, illustrated in Figure 4, takes the form of an interactive multi-modal kiosk that displays a realistically rendered avatar head which can interact via natural language. The avatar can engage with one or more participants and plays a simple game, in which the users have to respond to multiple-choice trivia questions.

The system's hardware and software architecture is illustrated in Figure 4. Data gathered from a wide-angle camera, a 4-element linear microphone array, and a 19" touch-screen is forwarded to a scene analysis module that fuses the incoming streams and constructs in real-time a coherent picture of the dynamics in the surrounding environment. The system detects and tracks the location of multiple agents in the scene, tracks the head pose for engaged agents, tracks the current speaker, and infers the focus of attention, activities, and goals of each agent, as well as the group relationships among different agents. An in-depth description of the hardware and scene analysis components falls beyond the scope of this paper, but details are available in (Bohus and Horvitz, 2009). The scene analysis results are forwarded to the control level, which is structured in a two-layer reactive-deliberative architecture. The reactive layer implements and coordinates various low-level behaviors, including engagement, conversational floor management and turn-taking, and coordinating spoken and gestural outputs. The deliberative layer plans the system's dialog moves and high-level engagement actions.

Overall, the game task was purposefully designed to minimize challenges in terms of speech recognition or dialog management, and allow us to focus our attention on the engagement processes. The avatar begins the interactions by asking the engaged user if they would like to play a trivia game. If the user agrees, the avatar goes through four multiple-choice questions, one at a time. After each question, the possible answers are displayed on the screen (Figure 4) and users can respond by either speaking an answer or by touching it. When the answer provided by the user is incorrect, the system provides a short explanation regarding the correct answer before moving on to the next question.

The system also supports multi-participant interactions. The engagement policy used to attract and engage

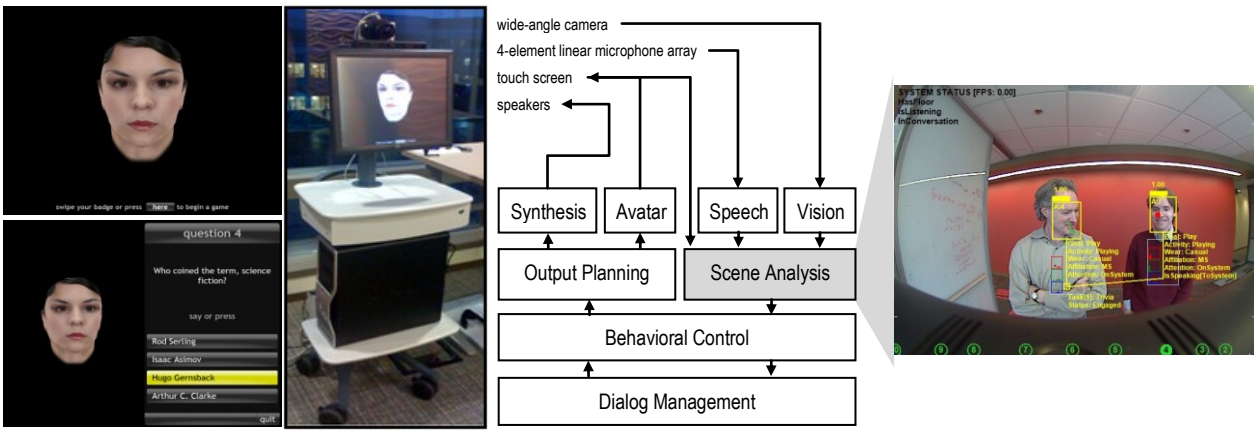


Figure 4. Trivia game dialog system: prototype, architectural overview, and runtime scene analysis

multiple users in a game is the focus of this observational study, and is discussed in more detail in the next subsection. Once the system is engaged with multiple users, it uses a multi-participant turn taking model which allows it to continuously track who the current speaker is, and who has the conversational floor (Bohus and Horvitz, 2009). At the behavioral level, the avatar orients its head pose and gaze towards the current speaker, or towards the addressee(s) of its own utterances. During multiplayer games, the avatar alternates between the users when asking questions. Also, after a response is received from one of the users, the avatar confirms the answer with the other user(s), *e.g.* "Do you agree with that?" A full sample interaction with the system is described in Appendix A, and the corresponding video is available online (Situated Interaction, 2009).

5.2 Multiparty Engagement Policy

The trivia game system implements the situated, multiparty engagement model described in Section 4. The sensing and behavioral control components are application independent and were previously described. We now describe the system's engagement policy, which is application specific.

As previously discussed, apart from using the inferred engagement state, actions and intentions for the agents in the scene, the proposed model also uses information about the high-level goals and activities of these agents when making engagement decisions. Specifically, the system tracks the goal of each agent in the scene, which can be *play*, *watch*, or *other*, and their current activity, which can be *passing-by, interacting, playing, watching,* or *departing*. The goal and activity recognition models are application specific, and in this case are inferred based on probabilistic conditional models that leverage information about the spatiotemporal trajectory of each agent and their spoken utterances, as well as global scene information (*e.g.* is the system engaged in an active interaction, etc.).

Initially, when the system is idle, it uses a conservative engagement policy and waits for the user to initiate engagement via an explicit action. Such actions include the user approaching and entering in an F-formation (Kendon, 1990b) with the system, *i.e.* standing right in front of it, swiping their badge, or pushing the start button (in the idle state the GUI displays "swipe your badge or press here to begin" below the avatar head).

While engaged in an interaction, the system attempts to engage bystanders in an effort to create a collaborative, multi-participant game. In this case, the engagement policy is conditioned on the inferred activities of the agents in the scene. Specifically, if a *watching* bystander is detected, the system temporarily disengages the current participant, and engages and attempts to "convince" the watching bystander to join the existing game. The prompts in this side interaction depend on the current game context, as shown in Table 1. If the watching bystander agrees to join in, the system adds him to the existing interaction, and continues a multi-participant game (see Table 1.A.) Conversely, if the bystander refuses, the system re-engages the previous participant and resumes the single-user game (see Table 1.B.) Additional examples are available in Appendix A.

If a *watching* bystander B is detected during the first 3 questions		
	and the *engaged* participant E has not made any mistakes	
A		[S to E]: Hold on a second. [S to B]: Hi. Excuse me, would you like to join in? [B]: Okay. [S to E&B]: Great. So let's see how well you can do together. Next question […]
	and the engaged participant E has just made a mistake, and the system did not previously engage the bystander B.	
B		[S to E]: Perhaps you can use some help. Hold on a second. [S to B]: Hi. Do you think you could help out? [B]: No thanks [S to B]: Okay. Well, I guess you can just watch for now and if you wait for a bit we'll be done and you can start a new game yourself [S to E]: Guess you're still on your own. Next question […]
	and the engaged participant E has just made a mistake, and the system did previously attempt to engage the bystander B.	
C		[S to E]: I think you could really use some help. Hold on a second. [S to B]: Are you sure you don't want to help out? Come on, this is fun. [B]: Sure [S to E&B]: Great. So let's see how well you can do together. Next question […]
If a *watching* bystander B is detected during the last question		
D		[S to E]: Excuse me for one moment. [S to B]: We're almost done here. If you wait for a bit we can start a new game right after [S to E]: Sorry about that […]

Table 1. Multiparty engagement policy

Finally, if the system is already engaged and a *watching* bystander is detected but only during the last question, the system engages them temporarily to let them know that the current game will end shortly and, if they wait, they can also start a new game (see Table 1.D).

5.3 Results and Lessons Learned

We deployed the system described above for 20 days near one of the kitchenettes in our building. The system attracted attention of passer-bys with the tracking motion of its virtual face that followed people as they passed by. Most people that interacted with the system did so for the first time; only a small number of people interacted several times. No instructions were provided for interacting with the system. We shall now review results from analysis of the collected data.

Throughout the 20 days of deployment, the system engaged in a total of 121 interactive events. Of these, in 54 cases (44%), a participant engaged the system but did not play the game. Typically, the participant would approach and enter in an F-formation with the system,

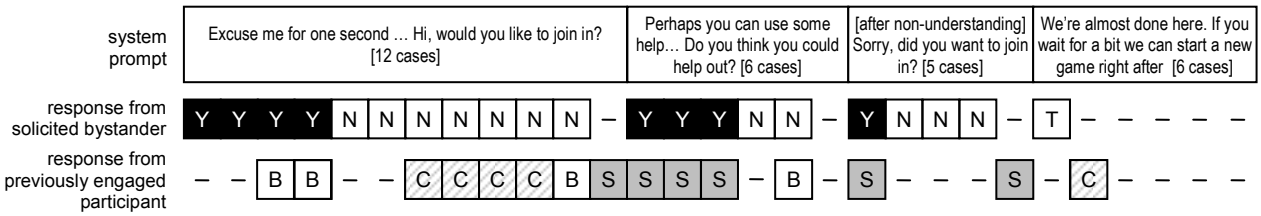


Figure 5. System multiparty engagement actions and responses from bystanders and already engaged participants. For bystander responses, Y denotes a positive response; N denotes a negative response; – denotes no response. For responses from previously engaged participant, B denotes utterances addressed to the bystander, C denotes side comments, S denotes responses directed to the system

but, once the system engaged and asked if they would like to play the trivia game, they responded negatively or left without responding. In 49 cases (40%), a single participant engaged and played the game, but no bystanders were observed during these interactions. In one case, two participants approached and engaged simultaneously; the system played a multi-participant game, but no other bystanders were observed. Finally, in the remaining 17 cases (14% of all engagements, 25% of actual interactions), at least one bystander was observed and the system engaged in multiparty interaction. These multiparty interactions are the focus of our observational analysis, and we will discuss them in more detail.

In 2 of these 17 cases, bystanders appeared only late in the interaction, after the system had already asked the last question. In these cases, according to its engagement policy, the system notified the bystander that they would be attended to momentarily (see Table 1.D), and then proceeded to finish the initial game. In 8 of the remaining 15 cases (53%), the system successfully persuaded bystanders to join the current interaction and carried on a multi-participant game. In the remaining 7 cases (47%), bystanders turned down the offer to join the existing game. Although this corpus is still relatively small, these statistics indicate that the system can successfully engage bystanders and create and manage multi-participant interactions in the open world.

Next, we analyzed more closely the responses and reactions from bystanders and already engaged participants to the system's multiparty engagement actions. Throughout the 17 multiparty interactions, the system planned and executed a total of 23 engagement actions soliciting a bystander to enter the game, and 6 engagement actions letting a bystander know that they will be engaged momentarily. The system actions and responses from bystanders and engaged participants are visually summarized in Figure 5, and are presented in full in Appendix B. Overall, bystanders successfully recognize that they are being engaged and solicited by the system and respond (either positively or negatively) in the large majority of cases (20 out of 23). In 2 of the remaining 3 cases, the previously engaged participant responded instead of the bystander; finally, in one case the bystander did not respond and left the area.

While bystanders generally respond when engaged by the system, the system's engagement actions towards bystanders also frequently elicits spoken responses from the already engaged participants; this happened in 14 out of 23 cases (61%). The responses are sometimes addressed to the system *e.g.* "Yes he does," or towards the bystander, *e.g.* "Say yes!", or they reflect general comments, *e.g.* "That's crazy!" These results show that, when creating the side interaction to solicit a bystander to join the game, the system should engage both the bystander and the existing user in this side interaction, or at least allow the previous user to join this side interaction (currently the system engages only the bystander in this interaction; see example from Appendix A.)

Furthermore, we noticed that, in several cases, bystanders provided responses to the system's questions even prior to the point the system engaged them in interaction (sometimes directed toward the system, sometimes toward the engaged participant.) We employed a system-initiative engagement policy towards bystanders in the current experiment. The initiative being taken by participants highlights the potential value of implementing a mixed-initiative policy for engagement. If a relevant response is detected from a bystander, this can be interpreted as an engagement action (recall from subsection 4.1 that engagement actions subsume expected opening dialog moves), and a mixed-initiative policy can respond by engaging the bystander, *e.g.* "Did you want to join in?" or "Please hang on, let's let him finish. We can play a new game right after that." This policy could be easily implemented under the proposed model.

We also noted side comments by both bystander and the existing participant around the time of multiparty engagement. These remarks typically indicate surprise and excitement at the system's multiparty capabilities. Quotes include: "That's awesome!", "Isn't that great!", "That's funny!", "Dude!", "Oh my god that's creepy!", "That's cool!", "It multitasks!", "That is amazing!", "That's pretty funny", plus an abundance of laughter and smiles. Although such surprise might be expected today with a first-time exposure to an interactive system that is aware of and can engage with multiple parties, we believe that expectations will change in the future, as these technologies become more commonplace.

Overall, this preliminary study confirmed that the system can effectively initiate engagement in multiparty settings, and also highlighted several core challenges for managing engagement and supporting multiparty interactions in the open world. A first important challenge we have identified is developing robust models for tracking the conversational dynamics in multiparty situations, *i.e.* identifying who is talking to whom at any given point. Secondly, the experiment has highlighted the opportunity for using more flexible, mixed-initiative engagement policies. Such policies will rely heavily on the ability to recognize engagement intentions; in (Bohus and Horvitz, 2009b), we describe the automated learning of engagement intentions from interaction data. Finally, another lesson we learned from these initial experiments is the importance of accurate face tracking for supporting multiparty interaction. Out of the 17 multiparty interactions, 7 were affected by vision problems (e.g. the system momentarily lost a face, or swapped the identity of two faces); 4 of these were fatal errors that eventually led to interaction breakdowns.

6 Summary and Future Work

We have described a computational model for managing engagement decisions in open-world dialog. The model harnesses components for sensing and reasoning about the engagement state, actions, and intentions of multiple participants in the scene, for making high-level engagement control decisions about who and when to engage, and for executing and rendering these actions in an embodied agent. We reviewed an observational study that showed that, when weaved together, these components can provide support for effectively managing engagement, and for creating and conducting multiparty interactions in an open-world context.

We believe that the components and policies we have presented provide a skeleton for engagement and interaction in open-world settings. However, there are important challenges and opportunities ahead. Future research includes developing methods for fine tuning and optimizing each of these subcomponents and their interactions. Along these lines, there are opportunities to employ machine learning to tune and adapt multiple aspects of the operation of the system. In (Bohus and Horvitz, 2009b) we introduce and evaluate an approach to learning models for inferring engagement actions and intentions online, through interaction. On another direction, we are investigating the use of decision-theoretic approaches for optimizing mixed-initiative engagement policies by taking into account the underlying uncertainties, the costs and benefits of interruption versus continuing collaboration, queue etiquette associated with expectations of fairness, etc. Another difficult challenge is the creation of accurate low-level behavioral models, including the fine-grained control of pose, gesture, and facial expressions. Developing such methods will likely have subtle, yet powerful influences on the effectiveness of signaling and overall grounding in multiparty settings. We believe that research on these and other problems of open-world dialog will provide essential and necessary steps towards developing computational systems that can embed interaction deeply into the natural flow of everyday tasks, activities, and collaborations.

Acknowledgments

We thank George Chrysanthakopoulos, Zicheng Liu, Tim Paek, Cha Zhang, and Qiang Wang for discussions and feedback in the development of this work.

References

M. Argyle and M. Cook, 1976, *Gaze and Mutual Gaze*, Cambridge University Press, New York

D. Bohus and E. Horvitz, 2009a, *Open-World Dialog: Challenges, Directions and Prototype*, to appear in KRPD'09, Pasadena, CA

D. Bohus and E. Horvitz, 2009b, *An Implicit-Learning Based Model for Detecting Engagement Intentions*, submitted to SIGdial'09, London, UK

E. Goffman, 1963, *Behaviour in public places: notes on the social order of gatherings*, The Free Press, New York

E.T. Hall, 1966, *The Hidden Dimension: man's use of space in public and private*, New York: Doubleday.

A. Kendon, 1990, *A description of some human greetings*, Conducting Interaction: Patterns of behavior in focused encounters, Studies in International Sociolinguistics, Cambridge University Press

A. Kendon, 1990b, *Spatial organization in social encounters: the F-formation system*, Conducting Interaction: Patterns of behavior in focused encounters, Studies in International Sociolinguistics, Cambridge University Press

M.P. Michalowski, S. Sabanovic, and R. Simmons, *A spatial model of engagement for a social robot*, in 9th IEEE Workshop on Advanced Motion Control, pp. 762-767

C. Peters, C. Pelachaud, E. Bevacqua, and M. Mancini, 2005, *A model of attention and interest using gaze behavior*, Lecture Notes in Computer Science, pp. 229-240.

C. Peters, 2005b, *Direction of Attention Perception for Conversation Initiation in Virtual Environments*, in Intelligent Virtual Agents, 2005, pp. 215-228.

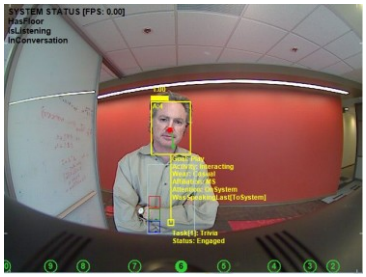
C.L. Sidner, C.D. Kidd, C. Lee, and N. Lesh, 2004, *Where to Look: A Study of Human-Robot Engagement*, IUI'2004, pp. 78-84, Madeira, Portugal

C.L. Sidner, C. Lee, C.D. Kidd, N. Lesh, and C. Rich, 2005, *Explorations in engagement for humans and robots*, Artificial Intelligence, 166 (1-2), pp. 140-164

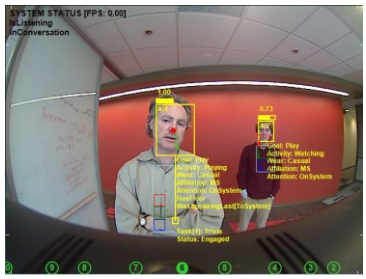
Situated Interaction, 2009, Project page: http://research.microsoft.com/~dbohus/research_situated_interaction.html

R. Vertegaal, R. Slagter, G.C.v.d.Veer, and A. Nijholt, 2001, *Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes*, CHI'01

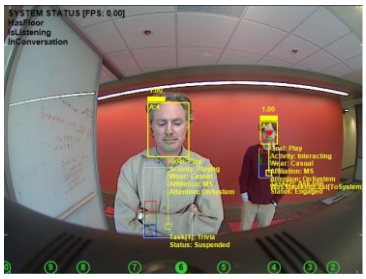
Appendix A. Sample multiparty interaction with trivia game dialog system (not part of the experiment)



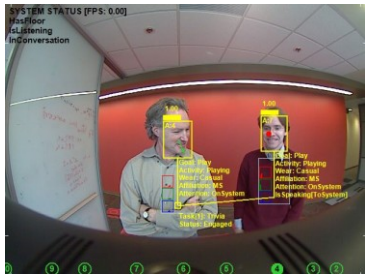
first person engages - around time t2



bystander appears – prior to t_3

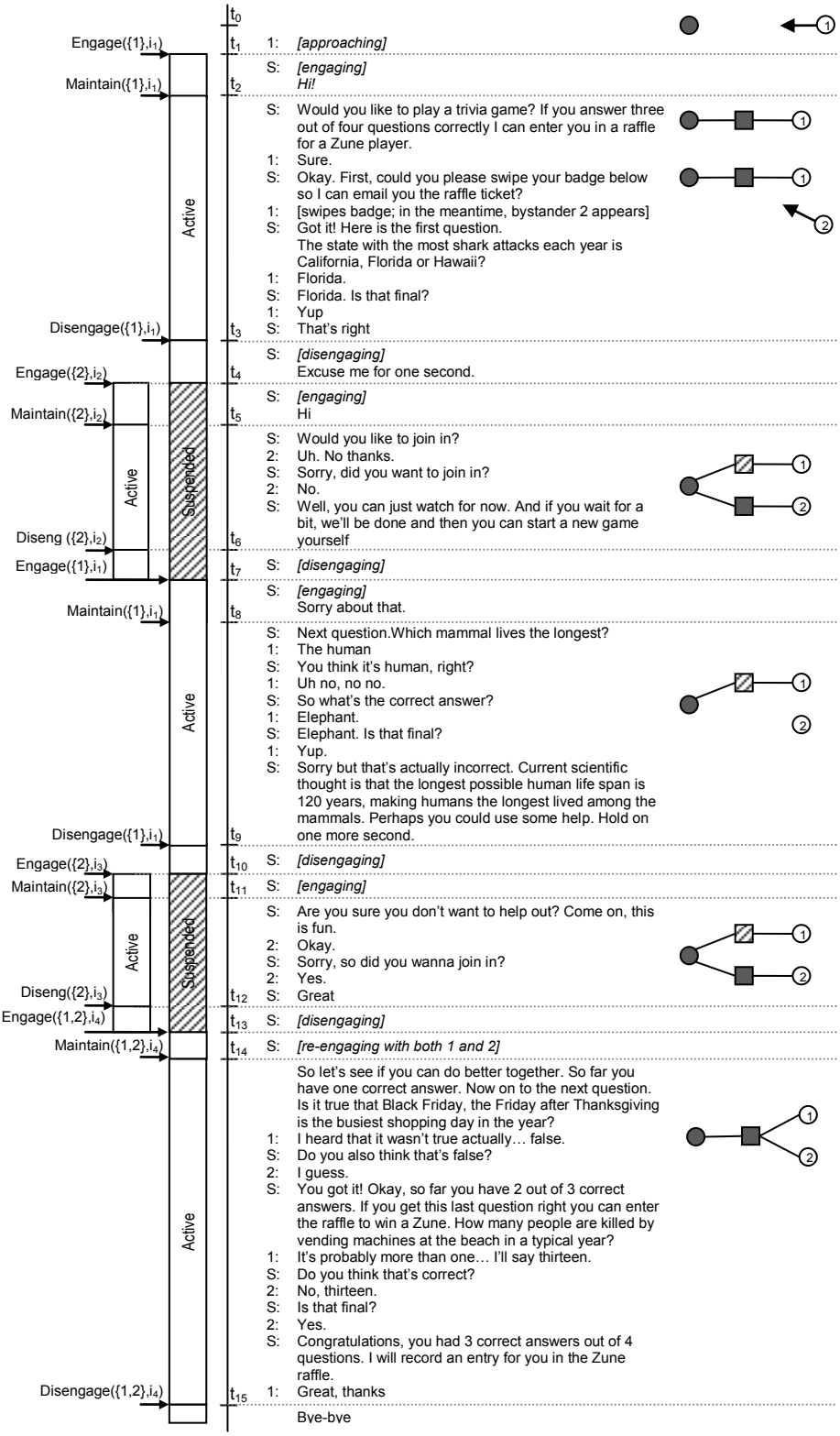


system engages bystander ~ t_5



participants play together ~ t_{14}

system
agent
active conversation
suspended conversation
other conversation



Appendix B. User responses to multiparty engagement actions.

S denotes the system, E denotes the already engaged participant, B denotes a *watching* bystander.

Actions and Resposes	Response from B	Response from E	Timing
[S to E]: *Hold on one second.* [S to B]: *Excuse me, would you like to join in?* **4 positive answers from B** **7 negative answers from B** **1 no answer from B (E answers)**	*Yes*		B only
	Yes	*Say yes*	Overlap
	Sure		B only
	Yes	[to B]: *Would you like to join in?*	E first
	No	*That's crazy!*	B first
	Oh, no. No + [moves away]	*That's funny!*	B first
	No thank you		B only
	No	*No?*	B first
	Woah, no.	*That's cool!*	B first
	No + [moves away] + *That's pretty funny.*	[laughs looking at B]	B first
	[laughs]	[laughs] *Yes. Oh yes.*	E only
[S to E]: *Perhaps you could use some help. Excuse me for one second.* [S to B]: *Hi, do you think you could help out?* **3 positive answers from B** **2 negative answers from B** **1 no-answer from B (moves away)**	*Yes.*	*Yes.*	B first
	Yes	*Yes he does.*	Overlap
	[laughs] + *No.*		B only
	[to E]: *Isn't that weird?* [to S]: *No.* [to E]: *Isn't that great?*	[to B]: *That is amazing!*	B first
	[laughs] + [moves out]	*Quit*	E only
	[laughs] + *Sure*	*Sure*	B first
If the initial response from B was not understood by the system, system asks one more time [S to B]: *Sorry, did you want to join in?* **1 positive answer from B** **3 negative answer from B** **1 no-answer from B (E answers)**	*No.* *Please.*	*Yes, I don't know, help me!*	B first
	No.		B only
	No.		B only
	No.		B only
		No.	E only
[S to B]: *We're almost done here. If you wait for a bit we can start a new game right after.* **1 answer from B** **1 answer from E** **4 no-answer from either B or E**	*Great, thanks.*		B only
		That's awesome	E only

Extracting decisions from multi-party dialogue using directed graphical models and semantic similarity

Trung H. Bui[1], **Matthew Frampton**[1], **John Dowding**[2], and **Stanley Peters**[1]

[1]Center for the Study of Language and Information, Stanford University
{thbui|frampton|peters}@stanford.edu

[2]University of California/Santa Cruz
jdowding@ucsc.edu

Abstract

We use directed graphical models (DGMs) to automatically detect decision discussions in multi-party dialogue. Our approach distinguishes between different dialogue act (DA) types based on their role in the formulation of a decision. DGMs enable us to model dependencies, including sequential ones. We summarize decisions by extracting suitable phrases from DAs that concern the issue under discussion and its resolution. Here we use a semantic-similarity metric to improve results on both manual and ASR transcripts.

1 Introduction

In work environments, people share information and make decisions in multi-party conversations known as meetings. The demand for systems that can automatically process, understand and summarize information contained in audio and video recordings of meetings is growing rapidly. Our own research, and that of other contemporary projects (Janin et al., 2004), aim at meeting this demand.

At present, we are focusing on the automatic detection and summarization of decision discussions. Our approach for detecting decision discussions involves distinguishing between different dialogue act (DA) types based on their role in the decision-making process. Two of these types are DAs which describe the *Issue* under discussion, and DAs which describe its *Resolution*. To summarize a decision discussion, we identify words and phrases in the Issue and Resolution DAs, which can be used to produce a concise, descriptive summary.

This paper describes new experiments in both detecting and summarizing decision discussions. In the detection stage, we investigate the use of Directed Graphical Models (DGMs). DGMs are attractive because they can be used to model sequence and dependencies between predictor variables. In the summarization stage, we attempt to improve phrase selection with a new feature that measures the level of semantic similarity between candidate Issue phrases and Resolution utterances, and vice-versa. The feature is generated by a semantic-similarity metric which uses WordNet as a knowledge source. The motivation is that ordinarily, the Issue and Resolution components in a decision summary should be semantically similar.

The paper proceeds as follows. Firstly, Section 2 describes related work, and Section 3, our data-set and annotation scheme for decision discussions. Section 4 then reports our decision detection experiments using DGMs, and Section 5, the summarization experiments. Finally, Section 6 draws conclusions and proposes ideas for future work.

2 Related Work

User studies (Banerjee et al., 2005) have confirmed that meeting participants consider decisions to be one of the most important meeting outputs, and (Whittaker et al., 2006) found that the development of an automatic decision detection component is critical to the re-use of meeting archives. With the new availability of substantial meeting corpora such as the AMI corpus (McCowan et al., 2005), recent years have therefore seen an increasing amount of research on decision-making dialog. This research has tackled issues such as the automatic detection of agreement and disagreement (Galley et al., 2004), and of the

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 235–243,
Queen Mary University of London, September 2009.

level of involvement of conversational participants (Gatica-Perez et al., 2005). In addition, (Verbree et al., 2006) created an argumentation scheme intended to support automatic production of argument structure diagrams from decision-oriented meeting transcripts. As yet, there has been relatively little work which specifically addresses the automatic detection and summarization of decisions.

Decision discussion detection: (Hsueh and Moore, 2007) used the AMI Meeting Corpus, and attempted to automatically identify DAs in meeting transcripts which are "decision-related". For each meeting, two manually created summaries were used to judge which DAs were decision-related: an extractive summary of the whole meeting, and an abstractive summary of its decisions. Those DAs in the extractive summary which support any of the decisions in the abstractive summary were manually tagged as decision-related. (Hsueh and Moore, 2007) then trained a Maximum Entropy classifier to recognize this single DA class, using a variety of lexical, prosodic, DA and conversational topic features. They achieved an F-score of 0.35.

Unlike (Hsueh and Moore, 2007), (Fernández et al., 2008b) made an attempt at modelling the structure of decision-making dialogue. The authors designed an annotation scheme that takes account of the different roles which utterances can play in the decision-making process—for example it distinguishes between DDAs (decision DAs) which initiate a discussion by raising an issue, those which propose a resolution, and those which express agreement for a proposed resolution. The authors annotated a portion of the AMI corpus, and then applied what they refer to as "hierarchical classification". Here, one *sub-classifier* per DDA class hypothesizes occurrences of that DDA class, and then based on these hypotheses, a *super-classifier* determines which regions of dialogue are decision discussions. All of the classifiers, (sub and super), were linear kernel binary Support Vector Machines (SVMs). Results were better than those obtained with (Hsueh and Moore, 2007)'s approach—the F1-score for detecting decision discussions in manual transcripts was .58 *vs.* .50. Note that (Purver et al., 2007) had previously pursued the same basic approach as (Fernández et al., 2008b) in order to detect action items.

In this paper, we build on the promising results of (Fernández et al., 2008b), by using Directed Graphical Models (DGMs) in place of SVMs. DGMs are attractive because they provide a natural framework for modelling sequence and dependencies between variables including the DDAs. We are especially interested in whether DGMs better exploit non-lexical features. (Fernández et al., 2008b) obtained much more value from lexical than non-lexical features (and indeed no value at all from prosodic features), but lexical features have disadvantages. In particular, they can be domain specific, increase the size of the feature space dramatically, and deteriorate more than other features in quality when ASR is poor.

Decision summarization: Recent years have seen research on spoken dialogue summarization (e.g. (Zechner, 2002)). Most has attempted to generate summaries of full dialogues, but some very recent research has focused on specific dialogue events, namely action items (Purver et al., 2007), and decisions (Fernández et al., 2008a).

(Fernández et al., 2008a) used the DDA annotation scheme mentioned above, and began by extracting the DDAs which raise issues or provide accepted resolutions. Only manual transcripts were used and the DDAs were extracted by hand rather than automatically. The next step was to parse each DDA with a general rule-based parser (Dowding et al., 1993), producing multiple short fragments rather than one full utterance parse. Then, for each DDA, an SVM regression model used various features (including parse, semantic and lexical features) to select the fragment which was most likely to appear in a gold-standard extractive decision summary. The entire manual utterance transcriptions were used as the baseline, and although the SVM's precision was high, it was not enough to offset the baseline's perfect recall, and so its F-score was lower. The "Oracle", which always chooses the fragment with the highest F1-score produced very good results. This motivates deeper investigation into how to improve the fragment/parse selection phase, and so we assess the usefulness of a semantic-similarity feature for the SVM. We conduct experiments with ASR as well as manual transcripts.

3 Data

For the experiments reported in this study, we used 17 meetings from the AMI Meeting Corpus (McCowan et al., 2005), a freely available corpus of

multi-party meetings with both audio and video recordings, and a wide range of annotated information including DAs and topic segmentation. Conversations are in English, but some participants are non-native English speakers. The meetings last around 30 minutes each, and are scenario-driven, wherein four participants play different roles in a company's design team: *project manager*, *marketing expert*, *interface designer* and *industrial designer*.

3.1 Modelling Decision Discussions

We use the same annotation scheme as (Fernández et al., 2008b) to model decision-making dialogue. As stated in Section 2, this scheme distinguishes between a small number of DA types based on the role which they perform in the formulation of a decision. Apart from improving the initial detection of decision discussions (Fernández et al., 2008b), such a scheme also aids their subsequent summarization, because it indicates which utterances contain particular types of information.

The annotation scheme is based on the observation that a decision discussion contains the following main structural components: (a) a topic or issue requiring resolution is raised, (b) one or more possible resolutions are considered, (c) a particular resolution is agreed upon and so becomes the decision. Hence the scheme distinguishes between three main decision dialogue act (DDA) classes: *issue* (*I*), *resolution* (*R*), and *agreement* (*A*). Class *R* is further subdivided into *resolution proposal* (*RP*) and *resolution restatement* (*RR*). *I* utterances introduce the topic of the decision discussion, examples being *"Are we going to have a backup?"* and *"But would a backup really be necessary?"* in Dialogue 1. On the other hand, *R* utterances specify the resolution which is ultimately adopted as the decision. *RP* utterances propose this resolution (e.g. *"I think maybe we could just go for the kinetic energy..."*), while *RR* utterances close the discussion by confirming/summarizing the decision (e.g. *"Okay, fully kinetic energy"*) . Finally, *A* utterances agree with the proposed resolution, signalling that it is adopted as the decision, (e.g. *"Yeah"*, *"Good"* and *"Okay"*). Note that an utterance can be assigned to more than one DDA class, and within a decision discussion, more than one utterance can be assigned to the same DDA class.

We use both manual and ASR one-best transcripts[1] in the experiments described here. DDA annotations were first made on the manual transcripts, and then transferred onto the ASR transcripts. Inter-annotator agreement was satisfactory, with kappa values ranging from .63 to .73 for the four DDA classes. Due to different segmentation, the manual and ASR transcripts contain a total of 15,680 and 8,357 utterances respectively, and on average, 40 and 33 DDAs per meeting. Hence DDAs are slightly less sparse in the ASR transcripts: for all DDAs, 6.7% *vs.* 4.3% of the total number of utterances, for *I*, 1.6% *vs.* 0.9%, for *RP*, 2% *vs.* 1%, for *RR*, 0.5% *vs.* 0.4%, and for *A*, 2.6% *vs.* 2%.

(1) A: Are we going to have a backup? Or we do just–
B: But would a backup really be necessary?
A: I think maybe we could just go for the kinetic energy and be bold and innovative.
C: Yeah.
B: I think– yeah.
A: It could even be one of our selling points.
C: Yeah –*laugh*–.
D: Environmentally conscious or something.
A: Yeah.
B: Okay, fully kinetic energy.
D: Good.[2]

4 Decision Discussion Detection using Directed Graphical Models

A directed graphical model (DGM) *M*, (see Murphy (2002)), is a directed acyclic graph consisting of nodes which represent random variables, arcs which represent dependencies among these variables, and a probability distribution P over the variables. Let $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ be a set of random variables that are associated with nodes in a DGM and $Pa(X_i)$ be parents of X_i. The probability distribution of the model M satisfies:

$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} (P(X_i)|Pa(X_i))$$

When a DGM is used as a classifier, the goal is to correctly infer the value of the class node $X_c \in \mathbf{X}$ given a vector of values for the observed node(s)

[1] We used SRI's Decipher for which (Stolcke et al., 2008) reports a word error rate of 26.9% on AMI meetings.

[2] This example was extracted from the AMI dialogue ES2015c and has been modified slightly for presentation purposes.

$X_o \subseteq \mathbf{X} \setminus X_c$. This is done by using M to find the value of X_c which gives the highest conditional probability $P(X_c|X_o)$.

To detect each individual DDA class, we examined the four simple DGMs in Figure 1 (see Appendix). The DDA node is binary where value 1 indicates the presence of a DDA and 0 its absence. The evidence node (E) is a multi-dimensional vector of observed values of non-lexical features. These include utterance features (UTT) such as length in words, duration in milliseconds, position within the meeting (as percentage of elapsed time), manually annotated dialogue act (DA) features[3] such as *inform*, *assess*, *suggest*, and prosodic features (PROS) such as energy and pitch. These features are the same as the non-lexical features used by Fernández et al. (2008b). The hidden component node (C) represents the distribution of observable evidence E as a single Gaussian in the *-sim* models, and a mixture in the *-mix* models. For the -mix models, the number of Gaussian components is hand-tuned during the training phase.

More complex models are constructed from the four simple models in Figure 1 to allow for dependencies between different DDAs. For example, the model in Figure 2 (see Appendix) generalizes Figure 1c with arcs connecting the DDA classes based on analysis of the annotated AMI data.

4.1 Experiments

The DGM classifiers in Figures 1 and 2 were implemented in Matlab using the BNT software[4]. Since the current BNT version does not support multiple time series training for fully observable Dynamic Bayesian Networks (DBNs), we extended the software for training models using this structure (e.g., Figure 1c and Figure 2).

A DGM classifier is considered to have hypothesized a DDA if the marginal probability of its DDA node is above a hand-tuned threshold. We tested the DGMs on manual and ASR transcripts in a 17-fold cross-validation, and evaluated their performance on both a per-utterance basis, and also with the same lenient-match metric as Fernández et al. (2008b). This allows a margin of 20 seconds preceding and following a hypothesized DDA, and so we refer to it as the 40 second metric. In addition, we hypothesized decision discussion regions using the DGM output and the following two simple rules:

- A decision discussion region begins with an *Issue* DDA.
- A decision discussion region contains at least one *Issue* DDA and one *Resolution* DDA.

To evaluate the accuracy of these hypothesized regions, like Fernández et al. (2008b), we divided the dialogue into 30-second windows and evaluated on a per window basis.

4.2 Results

Tables 1 and 2 show the F1-scores for each DGM when using the best feature sets (I: UTT+DA+PROS, RP: UTT+DA, RR: UTT, A: UTT+DA). The BN-mix model gives the highest F1-score for *A* on both evaluation metrics, and the DBN-mix model, the highest for *I*, *RP*, and *RR*, but there are no statistically significant differences between any of the alternative DGMs.

Classifier	I	RP	RR	A
BN-mix	.09	.09	.04	.19
DBN-mix	.16	.14	.05	.17
BN-sim	.12	.09	.04	.17
DBN-sim	.15	.11	.04	.16

Table 1: F1-score (per utterance) of the DGMs using the best combination of non-lexical features.

Classifier	I	RP	RR	A
BN-mix	.19	.24	.07	.38
DBN-mix	.27	.24	.07	.32
BN-sim	.23	.22	.06	.36
DBN-sim	.25	.22	.06	.31

Table 2: F1-score (40 seconds) of the DGMs using the best combination of non-lexical features.

To determine whether modeling dependencies between DDAs improves performance, we experimented with the DGMs that are generalized from the DBN-sim (Figure 2) and DBN-mix models. The F1-scores did not improve for *I*, *RP*, and *RR*, while for *A*, the DGM generalized from DBN-sim gave a .03 improvement according to the 40 seconds metric, but this was not statistically significant.

For each DDA, Table 3 compares the results of the best DGM and the hierarchical SVM classification method of Fernández et al. (2008b) (see

[3] We use the AMI DA annotations. These are only available for manual transcripts.

[4] http://www.cs.ubc.ca/~murphyk/Software/BNT/bnt.html

Section 2). The DGM performs better for all DDAs on both evaluation metrics ($p < 0.005$). Note that while prosodic features proved useless to SVM classifiers (Fernández et al. (2008b)), with DGMs, they have some predictive power.

		Per utterance			40 seconds		
Classifier	DDA	Pr	Re	F1	Pr	Re	F1
SVM	I	.03	.62	.05	.04	.89	.08
DGM		.11	.28	.16	.20	.44	.27
SVM	RP	.03	.60	.07	.05	.90	.10
DGM		.09	.35	.14	.16	.57	.24
SVM	RR	.01	.49	.02	.01	.80	.03
DGM		.02	.42	.05	.04	.58	.07
SVM	A	.05	.70	.10	.07	.90	.13
DGM		.13	.31	.19	.29	.55	.38

Table 3: Performance of the DGM classifier vs. the SVM classifier. Both use the best combination of non-lexical features.

We also generated results without DA features. Here, the best F1-scores for *I*, *RP*, and *A* degrade between .07 and .09 ($p < 0.05$), but they are still higher than the equivalent SVM results with DA features. Since (Fernández et al., 2008b) report that lexical features are the most useful for the SVM classifiers, it will be interesting to see how well the DGMs perform when they use lexical as well as non-lexical features.

Detecting DDAs in ASR transcripts: Table 4 compares the DGM F1-scores when using ASR one-best and manual transcripts. The DGMs perform well on ASR output. For *I* and *RP*, the results on ASR are actually higher, perhaps because the DDAs are less sparse. In the absence of DA features, prosodic features improve the performance for *A* in both sources.

	UTT				UTT+PROS			
	I	RP	RR	A	I	RP	RR	A
ASR	.20	.21	.06	.24	.16	.24	.07	.28
Man	.18	.17	.07	.27	.16	.15	.05	.30

Table 4: F1-scores (40 seconds) computed using ASR one-best vs. manual transcriptions.

Detecting decision discussion regions: Table 5 shows that according to the 30-second window metric, rule-based classification with DGM output compares well with hierarchical SVM classification (Fernández et al., 2008b). In fact, even when the latter uses lexical as well as non-lexical features, its F1-score is still about the same as the DGM-based classifier. Our future work will involve dispensing with the rule-based approach and designing a DGM which can detect decision discussion regions.

Classifier	Pr	Re	F1
SVM	.35	.88	.50
DGM	.39	.93	.55

Table 5: Results in detecting decision discussion regions for the SVM super-classifier and rule-based DGM classifier, both using the best combination of non-lexical features.

5 Decision Summarization

We now turn to the task of extracting useful phrases for summarization. Since a summary of a decision discussion should minimally contain the issue under discussion, and its resolution, we leave *Agreement (A)* utterances aside, and concentrate on extracting phrases from *Issues (I)* and *Resolutions (R)*.

Our basic approach is the same taken in (Fernández et al., 2008a): The WCN[5] of each *I* and *R* utterance is parsed by the Gemini parser (Dowding et al., 1993) to produce multiple short fragments, and then an SVM regression model uses certain features in order to select the parse that is most likely to match a gold-standard extractive summary. Our work is new in two respects: summarizing from ASR output in addition to manual transcriptions, and using a semantic-similarity feature in the SVM. This new feature is generated using Ted Pedersen's semantic-similarity package (Pedersen, 2002), and is motivated by the fact that ordinarily the *Issue* summary should be semantically similar to the *Resolution* and vice versa.

The next section describes the lexical resources used by Gemini, and Section 5.2, the metric for calculating semantic similarity.

5.1 Open-Domain Semantic Parser

Since human-human spoken dialogue, especially after being processed by an imperfect recognizer, is likely to be highly ungrammatical, we have developed a semantic parser that only attempts to find basic predicate-argument structures of the major phrase types (S, VP, NP, and PP) and has access to a broad-coverage lexicon. To build a broad-coverage lexicon, we used publicly available lexical resources for English, including COMLEX,

[5]When using manual transcripts, we create "dummy WCNs": WCNs with a single path.

VerbNet, WordNet, and NOMLEX.

COMLEX provides detailed syntactic information for the 40k most common words of English, and VerbNet, detailed semantic information for verbs, including verb class, verb frames, thematic roles, mappings of syntactic position to thematic roles, and selection restrictions on thematic role fillers. From WordNet we extracted another 15K nouns and the semantic class information for all nouns. These semantic classes were hand-aligned to the selectional classes used in VerbNet, based on the upper ontology of EuroWordNet. NOMLEX provides syntactic information for event nominalizations, and information for mapping the noun arguments to the corresponding verb syntactic positions.

These resources were combined and converted to the Prolog-based format used in the Gemini framework, which includes a fast bottom-up robust parser in which syntactic and semantic information is applied interleaved. Gemini can compute parse probabilities on the context-free skeleton of the grammar. In the experiments described here these parse probabilities are trained on Switchboard tree-bank data.

5.2 Semantic Similarity Metric: Normalized Path Length

Ted Pedersen's semantic similarity package (Pedersen, 2002) can be used to apply a number of different metrics that use WordNet as a knowledge base. The metric used here, *Normalized Path Length* (Leacock and Chodorow, 1998), defines the semantic similarity *sim* between words w_1 and w_2 as:

$$sim_{c_1,c_2} = -\log \frac{len(c_1, c_2)}{2 \times D} \qquad (1)$$

where c_1 and c_2 are concepts corresponding to w_1 and w_2, $len(c_1, c_2)$ is the length of the shortest path between them, and D is the maximum depth of the taxonomy.

5.3 Experiments

Data: For the manual transcripts in our subcorpus, the average length in words of *I* and *R* utterances is 12.2 and 11.9 respectively, and for the ASR, 22.4 and 18.1. To provide a gold-standard, phrases from *I* and *R* utterances in the manual transcriptions were annotated as summary-worthy. The aim was to select those phrases which should appear in an extractive summary, or could be the basis of a generated abstractive summary. As a general guideline, we tried to select the phrase(s) which describe the issue/resolution as succinctly as possible. This does not include phrases which express the speaker's attitude towards the issue/resolution. Dialogue 2 is an example where square brackets indicate which phrases were selected as summary-worthy.

(2) A:(*I*) So we we're looking at [*sliders for both volume and channel change*]
B:(*R*) I was thinking kind of [*just for the volume*]

Regression models: We use *SVMlight* (Joachims, 1999) to learn separate SVM regression models for *Issues* and *Resolutions*. These rank the Gemini parses for each utterance according to their likelihood of matching the gold-standard summary. The top-ranked parse is then entered into the automatically-generated decision summary.

Features: We train the regression models with various types of feature (see Table 6), including properties of the WCN paths, parse, semantic and lexical features. As lexical features are likely to be more domain-specific, and they dramatically increase size of the feature space, we prefer to avoid them if possible.

To generate the semantic-similarity feature for an I/R parse, we compute its semantic similarity with the full transcripts of each of the R/I utterances within the same decision discussion. The feature's value is then equal to the greatest of the resulting semantic-similarity scores. Since Ted Pedersen's package operates on the noun portion of WordNet, we must first extract all of the nouns in the parse/utterance transcription. Next, we form all of the possible pairs containing one noun from the parse, and one from the utterance transcription. Then we compute the semantic similarity for each pair, and take their sum to be the level of semantic similarity between the parse and the utterance transcription. We experimented with averaging rather than summing these scores, but the resulting semantic-similarity feature was less predictive.

Evaluation: The models are evaluated in 10-fold cross-validations using the same metric as (Fernández et al., 2008a): Recall is the total proportion of the gold-standard extractive summary

WCN	phrase length (WCN arcs) start/end point (absolute & percentage)
Parse	parse probability phrase type (S/VP/NP/PP)
Semantic	main verb VerbNet class head noun WordNet synset
Sem-sim	Normalized Path Length
Lexical	main verb, head noun

Table 6: Features for parse fragment ranking

	Issue			*Resolution*		
	Re	Pr	F1	Re	Pr	F1
Baseline	1.0	.50	.67	1.0	.60	.75
Oracle	.77	.96	.85	.74	.99	.84
WCN,parse,sem	.63	.69	.66	.61	.66	.64
+ sem-sim	.65	.71	.68	.64	.69	.67
+ lexical	.65	.67	.66	.65	.70	.67

Table 7: Parse ranking results for *I* & *R* Utterances using manual transcriptions.

covered by the selected parse; precision is the total proportion of the chosen parse which overlaps with the gold-standard summary. The baseline is the entire transcription, and we also compare to an "oracle" that always chooses a parse with the highest F1-score. Note that we use the extractive summaries from the manual transcriptions as the gold-standard for the evaluation of the results obtained with ASR.

Results and analysis: Results with manual transcriptions are shown in Table 7, and those with ASR, in Table 8. In all cases, when starting with a feature set containing WCN, parse and semantic features, the F1-score is improved by adding the semantic-similarity feature. For *Issues*, the F1-score improves from .66 to .68 with manual transcripts, and from .30 to .32 with ASR. The improvements for *Resolutions* are highly significant: with manual transcripts, the F1 score increases from .64 to .67 ($p < 0.005$), and with ASR, from .33 to .37 ($p < 0.005$). Note that the further addition of lexical features only produces a significant improvement in the case of *I* summarization with ASR.

Compared to the full transcript baseline, we achieve higher F1-scores for *Issues*—.68 *vs.* .67 with manual transcriptions, and .35 *vs.* .31 with ASR—but slightly lower for *Resolutions*. There remains a fairly large gap between our best scores and their corresponding oracles (especially with ASR), and so there may still be potential for substantial improvement.

	Issue			*Resolution*		
	Re	Pr	F1	Re	Pr	F1
Baseline	.77	.20	.31	.80	.27	.40
Oracle	.61	.87	.72	.59	.91	.72
WCN,parse,sem	.28	.33	.30	.31	.35	.33
+ sem-sim	.30	.34	.32	.35	.38	.36
+ lexical	.35	.35	.35	.34	.39	.37

Table 8: Parse ranking results for *I* & *R* Utterances using ASR.

6 Conclusions and Future Work

This paper has presented work on the detection and summarization of decision discussions in multi-party dialogue. In the detection experiments, we investigated the use of directed graphical models (DGMs), and found that when using non-lexical features, the DGMs outperform the hierarchical SVM classification method of Fernández et al. (2008b). The F1-score for the four DDA classes increased between .04 and .19 ($p < .005$), and for identifying decision discussion regions, by .05. This is encouraging because lexical features have disadvantages—for example they can be domain specific and greatly increase the feature space. In addition, modelling the dependencies between the DDA classes increased performance for *Agreement* utterances, and the DGMs were robust to ASR.

In the summarization experiments, we summarized decision discussions by extracting key words/phrases from their *Issue (I)* and *Resolution (R)* utterances. Each utterance's Word Confusion Network was parsed with an open-domain semantic parser, thus producing multiple candidate phrases, and then an SVM regression model selected one of these phrases to enter into the summary. The experiments here investigated the usefulness of a new SVM feature which measures the level of semantic similarity between candidate *I* parses and *R* utterances, and vice-versa. This feature was generated with a semantic-similarity metric which uses WordNet as a knowledge source. It was found to improve performance with both manual transcripts and ASR, and for *R* summarization, the improvements were highly significant ($p < .005$).

In future work, we plan to integrate lexical features into our DGMs by using a switching Dynamic Bayesian Network similar to that reported in (Ji and Bilmes, 2005). We also plan to extend the decision discussion annotation scheme so that we can try to automatically extract supporting ar-

guments for decisions.

Acknowledgements This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185/0004, and by the Department of the Navy Office of Naval Research (ONR) under Grants No. N00014-05-1-0187 and N00014-09-1-0106. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or ONR.

References

Satanjeev Banerjee, Carolyn Rosé, and Alex 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*.

John Dowding, Jean Mark Gawron, Doug Appelt, John Bear, Lynn Cherny, Robert Moore, and Douglas Moran. 1993. GEMINI: a natural language system for spoken-language understanding. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Raquel Fernández, Matthew Frampton, John Dowding, Anish Adukuzhiyil, Patrick Ehlen, and Stanley Peters. 2008a. Identifying relevant phrases to summarize decisions in spoken meetings. In *Proceedings of Interspeech*.

Raquel Fernández, Matthew Frampton, Patrick Ehlen, Matthew Purver, and Stanley Peters. 2008b. Modelling and detecting decisions in multi-party dialogue. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*.

Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. 2004. Identifying agreement and disagreement in conversational speech: Use of Bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Daniel Gatica-Perez, Ian McCowan, Dong Zhang, and Samy Bengio. 2005. Detecting group interest level in meetings. In *Proceedings of ICASSP*.

Pey-Yun Hsueh and Johanna Moore. 2007. Automatic decision detection in meeting speech. In *Proceedings of MLMI 2007*, Lecture Notes in Computer Science. Springer-Verlag.

Adam Janin, Jeremy Ang, Sonali Bhagat, Rajdip Dhillon, Jane Edwards, Javier Marcías-Guarasa, Nelson Morgan, Barbara Peskin, Elizabeth Shriberg, Andreas Stolcke, Chuck Wooters, and Britta Wrede. 2004. The ICSI meeting project: Resources and research. In *Proceedings of the 2004 ICASSP NIST Meeting Recognition Workshop*.

Gang Ji and Jeff Bilmes. 2005. Dialog act tagging using graphical models. In *Proceedings of ICASSP*.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. MIT Press.

Claudia Leacock and Martin Chodorow, 1998. *WordNet: An Electronic Lexical Database*, chapter Combining local context and WordNet similarity for word sense identification. University of Chicago Press.

Iain McCowan, Jean Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. The AMI Meeting Corpus. In *Proceedings of Measuring Behavior, the 5th International Conference on Methods and Techniques in Behavioral Research*, Wageningen, Netherlands.

Kevin Murphy. 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, University of California Berkeley.

Ted Pedersen. 2002. Semantic similarity package. http:/www.d.umn.edu/ tpederse/similarity.

Matthew Purver, John Dowding, John Niekrasz, Patrick Ehlen, Sharareh Noorbaloochi, and Stanley Peters. 2007. Detecting and summarizing action items in multi-party dialogue. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium.

Andreas Stolcke, Xavier Anguera, Kofi Boakye, Özgür Çetin, Adam Janin, Matthew Magimai-Doss, Chuck Wooters, and Jing Zheng. 2008. The ICSI-SRI spring 2007 meeting and lecture recognition system. In *Proceedings of CLEAR 2007 and RT2007*.

Daan Verbree, Rutger Rienks, and Dirk Heylen. 2006. First steps towards the automatic construction of argument-diagrams from real discussions. In *Proceedings of the 1st International Conference on Computational Models of Argument*, volume 144, pages 183–194. IOS press.

Steve Whittaker, Rachel Laban, and Simon Tucker. 2006. Analysing meeting records: An ethnographic study and technological implications. In S. Renals and S. Bengio, editors, *Machine Learning for Multimodal Interaction: Second International Workshop, MLMI 2005, Revised Selected Papers*, volume 3869 of *Lecture Notes in Computer Science*, pages 101–113. Springer.

Klaus Zechner. 2002. Automatic summarization of open-domain multiparty dialogues in diverse genres. *Computational Linguistics*, 28(4):447–485.

Appendix

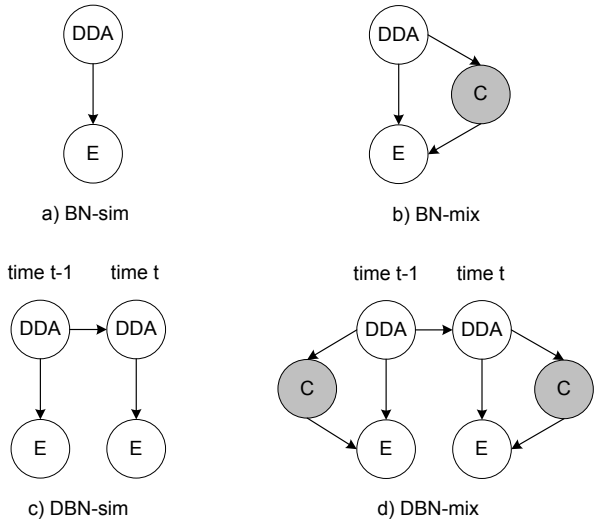


Figure 1: Simple DGMs for individual decision detection. During training, the shaded nodes are hidden, and the clear nodes are observable.

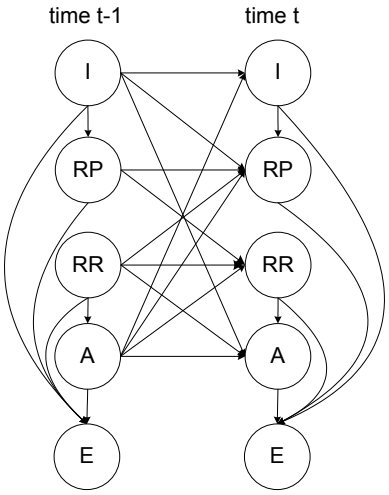


Figure 2: A DGM that takes the dependencies between decisions into account.

Learning to Predict Engagement with a Spoken Dialog System in Open-World Settings

Dan Bohus
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
dbohus@microsoft.com

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA, 98052
horvitz@microsoft.com

Abstract

We describe a machine learning approach that allows an open-world spoken dialog system to learn to predict engagement intentions in situ, from interaction. The proposed approach does not require any developer supervision, and leverages spatiotemporal and attentional features automatically extracted from a visual analysis of people coming into the proximity of the system to produce models that are attuned to the characteristics of the environment the system is placed in. Experimental results indicate that a system using the proposed approach can learn to recognize engagement intentions at low false positive rates (*e.g.* 2-4%) up to 3-4 seconds prior to the actual moment of engagement.

1 Introduction

We address the challenge of predicting the forthcoming engagement of people with *open-world* conversational systems (Bohus and Horvitz, 2009a), *i.e.* systems that operate in relatively unconstrained environments, where multiple participants might come and go, establish, maintain and break the communication frame, and simultaneously interact with a system and with others. Examples of such systems include interactive billboards in a mall, robots in a home environment, intelligent home control systems, interactive systems that provide assistance and support during procedural tasks, etc.

In traditional *closed-world* dialog systems the engagement problem is generally resolved via simple, unambiguous signals. For example, engagement is generally assumed once a phone call is answered by a telephony dialog system. Similarly, a push-to-talk button can provide a clear engagement signal for a speech enabled mobile application. These solutions are however inappropriate for systems that must operate continuously in open, dynamic environments, and engage with multiple people and groups over time. Such systems should ideally be ready to initiate dialog in a fluid, natural manner. They should manage engagement with participants who are close by, and with those who are at a distance, with participants who have a standing plan to interact with a system, and with those whom opportunistically decide to engage, in-stream with their other ongoing activities. In recognizing engagement intentions, such systems need to minimize false positives, while also minimizing the unnatural delays and discontinuities that come with false negatives about engagement intentions.

The work described in this paper is set in the larger context of a computational model for supporting fluid engagement in open-world dialog systems that we have previously described in (Bohus and Horvitz, 2009b). The above mentioned model harnesses components for sensing the engagement state, actions, and intentions of multiple participants in the scene, for making engagement control decisions, and for rendering these decisions into coordinated low-level behaviors, such as the changing pose and expressions of the face of an embodied agent. In this paper, we focus on the sensing subcomponent of this larger model and describe an approach for automatically learning to detect engagement intentions from interaction.

2 Related Work

The challenges of engagement between people, and between people and computational systems, have already received some attention in the conversational analysis, sociolinguistics, and human-computer interaction communities. For instance, in an early treatise Goffman (1963) discusses how people use cues to detect engagement in an effort to avoid the social costs of engaging in interaction with an unwilling participant. In later work, Kendon (1990a) presents a detailed investigation of video sequences of greetings in human-human interaction, and identifies several stages of complex coordinated action (*pre-sighting, sighting, distance salutation,*

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 244–252,
Queen Mary University of London, September 2009.

approach, close salutation), together with the head and body gestures that they typically involve. In (1990b), Kendon also introduces the notion of an *F-formation*, a pattern said to arise when "two or more people sustain a spatial and orientational relationship in which they have equal, direct, and exclusive access," and discusses the role of F-formations in establishing and maintaining social interactions. Argyle and Cook (1976) as well as others (*e.g.,* Duncan, 1972; Vertegaal et al., 2001) have identified and discussed the various functions of eye gaze in maintaining social and communicative engagement. Overall, this body of work suggests that engagement is a rich, mixed-initiative, and well-coordinated process that involves non-verbal cues and signals, such as spatial trajectory and proximity, gaze and mutual attention, head and hand gestures, and verbal greetings.

More recently, several researchers have investigated issues of engagement in human-computer and human-robot interaction contexts. Sidner et al. (2004; 2005) define engagement as "the process by which two (or more) participants establish, maintain and end their perceived connection during interactions they jointly undertake," and conduct a user study that explores the process of maintaining engagement. They show that people direct their attention to a robot more often when the robot makes engagement gestures throughout an interaction, *i.e.* tracks the user's face, and points to relevant objects at appropriate times in the conversation.

Peters et al (2005a; 2005b) use an alternative definition of engagement as "the value that a participant in an interaction attributes to the goal of being together with the other participant(s) and of continuing the interaction," and present the high-level schematics for an algorithm for establishing and maintaining engagement. The proposed algorithm highlights the importance of eye gaze and mutual attention in this process and relies on a heuristically computed *interest level* to decide when to begin a conversation.

Michalowski et al (2006) propose and conduct experiments with a spatial model of engagement, grounded in proxemics (Hall, 1966). Their model classifies relevant agents in the scene in four different categories based on their distance to the robot: *present* (standing far), *attending* (standing closer), *engaged* (next to the robot), and *interacting* (standing right in front of the robot). The robot's behaviors are in turn conditioned on these categories: the robot turns towards attending people, greets engaged people and verbally prompts interacting people for input. The authors discuss several lessons learned from an observational study conducted with this robot in a building lobby. They find that the fast-paced movements of people in the environment pose a number of challenges: often the robot greeted people too late (earlier anticipation was needed), or greeted people that did not intend to engage (more accurate anticipation was needed). The authors recognize that these limitations stem partly from their reliance on static models, and hypothesize that temporal information such as speed and trajectory may provide additional cues regarding a person's engagement with the robot.

In this paper, we expand on our previous work on a situated multiparty engagement model (Bohus and Horvitz, 2009b). Specifically, we focus on a key subcomponent in this model: detecting whether or not a user intends to engage in an interaction with a system. We introduce an approach that improves upon the existing work (Peters 2005a, 2005b; Michalowski et. al, 2006) in several significant ways. First, the approach is data-driven: the use of machine learning techniques allows the system to adapt to the specific characteristics of its physical location and to the behaviors of the surrounding population of potential participants. Second, we leverage a wide array of observations, including temporal features. Finally, no developer supervision is required for training the model: the supervision signal is extracted automatically, in-stream with the interactions, allowing for online learning and adaptation.

3 Situated Multiparty Engagement Model

To set the broader context for the work described in this paper, we now briefly review the overall model for managing engagement in an open-world setting introduced in (Bohus and Horvitz, 2009b). The model is centered on a reified notion of *interaction*, defined as a basic unit of sustained, interactive problem-solving. Each interaction can involve two or more participants, and this number may vary in time; new participants may join an existing interaction and current participants may leave an interaction at any point in time. The system is actively engaged in at most one interaction at a time (with one or multiple participants), but it can simultaneously keep track of additional, suspended interactions. In this context, engagement is viewed as the process subsuming the joint, coordinated activities by which participants initiate, maintain, join, abandon, suspend, resume, or terminate an interaction.

Successfully managing this process requires that the system (1) senses and reasons about the engagement state, actions and intentions of multiple agents in the scene, (2) makes high-level engagement control decisions (*i.e.* about whom to engage or disengage with, and when) and (3) executes and signals these decisions to the other participants in an appropriate manner (*e.g.* via a set of coordinated behaviors such as gestures, greetings, etc.) The proposed model, illustrated in Figure 1, subsumes these three components.

The sensing subcomponent in the model tracks the engagement state, engagement actions, and engagement intention for each agent in the visual scene. The engagement state, $ES_a^i(t)$, denotes whether an agent a is

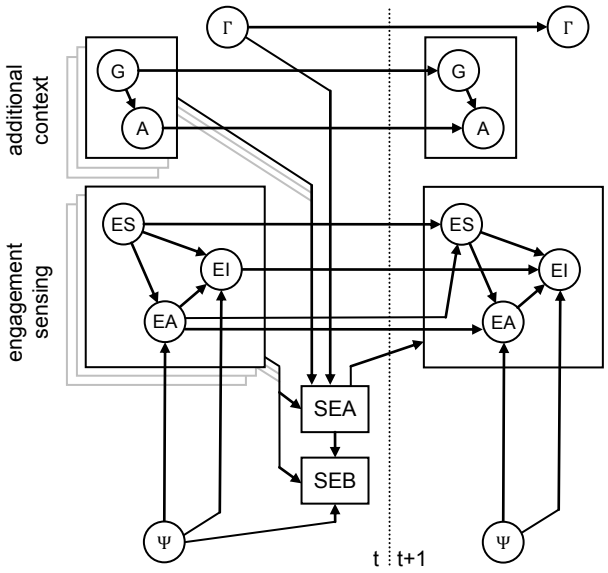


Figure 1. Graphical model showing key variables and dependencies in managing engagement.

engaged in interaction i and is modeled as a deterministic variable with two possible values: *engaged* and *not-engaged*. The state is updated based on the joint actions of the system and the agent.

A second engagement variable, $EA_a^i(t)$, models the actions that an agent takes to initiate, maintain or terminate engagement. There are four possible engagement actions: *engage*, *no-action*, *maintain*, *disengage*. These actions are tracked by means of a conditional probabilistic model that takes into account the engagement state $ES_a^i(t)$, the previous agent and system actions, as well as additional sensory evidence Ψ capturing committed engagement actions, such as: salutations (*e.g.* "Hi!"); calling behaviors (*e.g.* "Laura!"); the establishment or the breaking of an F-formation (Kendon, 1990b); expected opening dialog moves (*e.g.* "Come here!") *etc.*

A third variable in the proposed model, $EI_a^i(t)$, tracks whether or not each agent intends to be engaged in a conversation with the system. Like the engagement state, the intention can either be *engaged* or *not-engaged*. Intentions are tracked separately from actions since an agent might intend to engage or disengage the system, but not yet take an explicit engagement action. For instance, let us consider the case in which the system is already engaged in an interaction and another use is waiting in line to interact with the system: although the waiting user does not take an explicit, committed engagement action, she might signal (*e.g.* via a glance that makes brief but clear eye contact with the interactive system) that her intention is to engage in a new conversation once the opportunity arises. More generally, the engagement intention captures whether or not an agent would respond positively should the system initiate engagement. In that sense, it roughly corresponds to Peters' (2005; 2005b) "interest level", *i.e.* to the value the agent attaches to being engaged in a conversation with the system. Like engagement actions, engagement intentions are inferred based on probabilistic models that take into account the current engagement state, the previous agent and system actions, the previous engagement intention, as well as additional evidence that captures implicit engagement cues, *e.g.* the spatiotemporal trajectory of the participant, the level of sustained mutual attention, etc.

Based on the inferred engagement state, actions, and intentions of the agents in the scene, as well as other additional high-level evidence such as the agents' inferred goals (G), activities (A) and relationships (Γ), the proposed model outputs engagement actions – denoted by the SEA decision node in Figure 1. The action-space consists of the same four actions previously discussed: *engage*, *disengage*, *maintain* and *no-action*. At the lower level, the engagement decisions taken by the system are translated into a set of coordinated lower-level behaviors (SEB) such as head gestures, making eye contact, facial expressions, salutations, interjections, etc.

In related work (Bohus and Horvitz, 2009a; 2009b), we have demonstrated how this model can be used to effectively create and support multiparty interactions in an open-world context. Here, we focus on one specific subcomponent of this framework: the model for detecting engagement intentions.

4 Approach

To illustrate the problem of detecting engagement intentions, consider for instance a situated conversational system that examines through its sensors the scenes from Figure 3. How can such a system detect whether the person in the image intends to engage in a conversation or is just passing-by? Studies of human-human conversational engagement (Goffman, 1963; Argyle and Cook, 1976; Duncan, 1972; Kendon, 1990, 1990b) indicate that people signal and detect engagement intentions by producing and monitoring for a variety of cues, including gaze and sustained attention, trajectory and proximity, head and hand gestures, body pose, etc.

In the proposed approach, we use machine learning techniques, and leverage a wide array of observations from the sensors to create a model that allows an open-world interactive system to detect the specific patterns characterizing an engagement intention. Existing work on detecting engagement intentions has focused on static heuristic models that leverage proximity and attention features (Peters, 2005, 2005b; Michalowski, 2006). As previously discussed, psychologists have shown the important role played by geometric relationships, trajectories, and sustained attention in signaling and detecting engagement. The use of machine learning allows us to consider a wide array of such features, including trajectory, speed, and the attention of agents over time.

In general, as discussed in the previous section, the engagement intentions of an agent may evolve temporally under the proposed model, as a function of the various system actions and behaviors (*e.g.* an embodied system that makes eye contact, or smiles, or moves toward a participant might alter the engagement intention of that participant). In this work we concentrate on a simplified problem, in which the system's behavior is fixed (*e.g.* system always tracks people that pass by), and the engagement intention can be assumed constant within a limited time window.

The central idea of the proposed approach is to start by using a very conservative (*i.e.,* low false-positives) detector for engagement intentions, such as a push-to-engage button, and automatically gather sensor data surrounding the moments of engagement, together with labels that indicate whether someone actually engaged or not. Note that the system eventually finds out if a person becomes engaged with it. If we assume that an intention to engage existed for a limited window of time prior to the moment of engagement, the collected data can be used to learn a model for predicting this intention ahead of the actual moment of engagement. The proposed approach therefore enables a system to learn in-situ models for predicting forthcoming engagement, and the models are attuned to the specifics of the environment the system is in. No explicit developer supervision is required, as the training labels are extracted automatically from interaction.

5 Experimental Setup

To provide an ecologically valid basis for data collection and for evaluating the proposed approach, we developed a situated conversational agent and deployed it in the real-world. The system, illustrated in Figure 2, is an interactive multimodal kiosk that displays a realistically rendered avatar head. The avatar can engage and interact via natural language with one or more participants, and plays a simple game in which the users have to respond to multiple-choice trivia questions. The system, and sample interactions are described in more detail in (Bohus and Horvitz, 2009.)

The hardware and software architecture is also illustrated in Figure 2. Data gathered from a wide-angle camera, a 4-element linear microphone array, and a 19" touch-screen is forwarded to a scene analysis module that fuses the incoming streams and constructs in real-time a coherent picture of the dynamics in the surrounding environment. The system detects and tracks the location of multiple agents in the scene, tracks the head pose for engaged agents, and infers the focus of attention, activities, goals and (group) relationships among different agents in the scene. An in-depth description of these scene analysis components falls beyond the scope of this paper, but more details are available in (Bohus and Horvitz, 2009). The scene analysis results are forwarded to the control level, which is structured in a two-layer reactive-deliberative architecture. The reactive layer implements and coordinates low-level behaviors, including engagement, conversational floor management and turn-taking, and coordinating spoken and gestural outputs. The deliberative layer plans the system's dialog moves and high-level engagement actions.

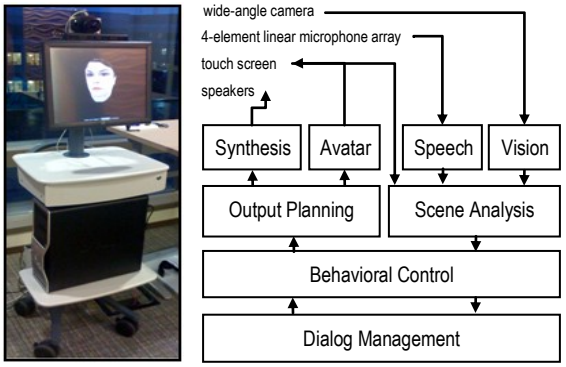


Figure 2. System prototype and architectural overview.

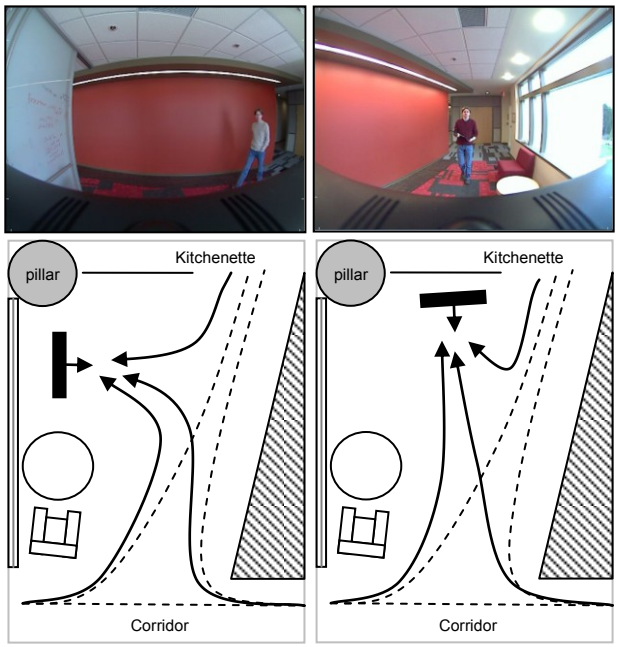


Figure 3. Placement and visual fields of view for *side* (right) and *front* (left) orientations.

We deployed the system described above in an open-space near the kitchenette area in our building. As we were interested in exploring the influence of the spatial setup on the engagement models, we deployed the system in two different spatial orientations, illustrated together with the resulting visual fields of view in Figure 3. Even though the location is similar, the two orientations create considerable differences in the relative trajectories of people that go by (dashed lines) and people that engage with the system (continuous lines). In the *side* orientation, people typically enter the system's field

	Side	Front	Total
Size (hours:minutes)	83:16	75:15	158:32
# face traces	2025	1249	3274
# engaged	72	74	146
% engaged	3.55%	5.92%	4.46%
# false-positive engaged	1	5	6
% false-positive engaged	0.04%	0.40%	0.18%
# not-engaged	1953	1175	3128
% not-engaged	96.45%	94.08%	95.54%

Table 1. Corpus statistics.

of view and approach it from the sides. In the *front* orientation, people enter the field of view and approach either frontally, or from the immediate right side.

6 Data and Modeling

The system was deployed during regular business hours for 10 days in each of the two orientations described above, for a total of 158 hours and 32 minutes. No instructions were provided and most people that interacted with the system did so for the first time.

6.1 Corpus and Implicit Labels

Throughout the data collection, the system used a conservative heuristic to detect engagement intentions: it considered that a user wanted to engage when they approached the system and entered in an F-formation (Kendon, 1990b) with it. Specifically, if a sufficiently large (close by) frontal face was detected in front of it, the system triggered an engaging action and started the interaction. We found this F-formation heuristic to be fairly robust, having a false-positive rate of 0.18% (6 false engagements out of 3274 total faces tracked). In 2 of these cases the face tracker committed an error and falsely identified a large nearby face, and in 4 cases a person passed by very close to the system but without any visible intention to engage.

Although details on false-negative statistics have not yet been calculated (this would require a careful examination of all 158 hours of data), our experience with the face detector suggests this number is near 0. In months of usage, we never observed a case where the system failed to detect a close by, frontal face. At the same time, we note that there is an important distinction between people who *actually engage* with the system, and people who *intend to engage*, but perhaps not come in close-enough proximity for the system to detect this intention (according to the heuristic described above). In this sense, while our heuristic can detect people who engage at a 0 false-negative rate, the false-negative rate with respect to engagement intentions is non-zero. Despite these false-negatives, we found that the proposed heuristic still represents a good starting point for learning to detect engagement intentions. As we shall see later, empirical results indicate that, by learning to detect who actually engages, the system can learn to also detect people who might intend to engage, but who ultimately do not engage with the dialog system.

In the experiments described here, we focus on detecting engagement intentions for people that approached while the system was idle. We therefore automatically eliminated all faces that were temporally overlapping with the periods when the system was already engaged in an interaction. For the remaining face traces, we automatically generate labels as follows:

- if a person entered in an F-formation and became engaged in interaction with the system at time t_e, the corresponding face trace was labeled with a positive engagement intention label from t_e-20sec; until t_e; the initial portion of the trace, from the moment it was detected until t_e-20sec was marked with a negative engagement intention label. Finally, the remainder of the trace (from t_e until the face disappeared) was discarded, as the user was actively engaged with the system during this time.
- if the face was never engaged in interaction (*i.e.* a person was just passing by), the entire trace was labeled with a negative engagement intention.

Note that in training the models described below we used these automatic labels, which are not entirely accurate: they include a small number of false-positives, as discussed above. However, for evaluation purposes, we used the corrected labels (no false-positives).

6.2 Models

To review, the task at hand is to learn a model for predicting engagement intentions, based on information that can be extracted at runtime from face traces, including spatiotemporal trajectory and cues about attention. We cast this problem as a frame-by-frame binary classification task: at each frame, the model must classify each visible face as either intending to engage or not. We used a maximum entropy model to make this prediction:

$$P(EI|X) = \frac{1}{Z(X)} exp\left(\sum_i \lambda_i \cdot f_i(X)\right)$$

The key role in the proposed maximum entropy model is played by the set of features $f_i(X)$, which must capture cues that are relevant for detecting an engagement intention. We designed several subsets of features, summarized in Table 2. The location subset, loc, includes the x and y location of the detected face in the visual scene, and the width and height of the face region, which indirectly reflect the proximity of the agent. The second feature subset, loc+ff, also includes a probability score (and a binarized version of it) produced by the face detector which reflects the confidence that the face is frontal and thus provides an automatic measure of the focus-of-attention of the agent. Apart from these auto-

Feature sets	Description **[total # of features in set]**
Loc	location features: x, y, width and height **[4]**
loc+ff	location features plus a confidence score indicating whether the face is frontal (ff), as well as a binary version of this score (ff=1) **[6]**
traj(loc)	location features plus trajectory of location features over windows of 5, 10, 20, 30 frames **[118]**
traj(loc+ff)	location and face frontal features, as well as trajectory of location and of face-frontal features over windows of 5, 10, 20, 30 frames **[172]**
traj(loc+attn)	location and manually labeled attention features, as well as trajectory of location and of attention over windows of 5, 10, 20, 30 frames **[133]**

Table 2. Feature sets for detecting engagement intention.

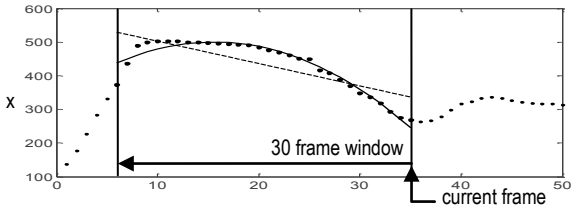


Figure 4. Trajectory features extracted by fitting linear and quadratic functions.

matically generated attention features, we also experimented with a manually annotated binary attention score, attn. The attention of each detected face was manually tagged throughout the entire dataset. This information is not available to the system at runtime; we use it only to identify an upper performance baseline.

The maximum entropy model is not temporally structured. The temporal structure of the spatial and attentional trajectory is captured via a set of additional features, derived as follows. Given an existing feature f, we compute a set of trajectory features traj.w(f) by accumulating aggregate statistics for the feature f over a past window of size w frames. We explored windows of size 5, 10, 20, 30. For continuous features, the trajectory statistics include the min, max, mean, and variance of the features in the specified window. In addition, we performed a linear and a quadratic fit of f in this window, and used the resulting coefficients (2 for the linear fit and 3 for the quadratic fit) as features (see the example in Figure 4). For the binary features, the trajectory statistics include the number and proportion of times the feature had a value of 1 in the given window, and the number of frames since the feature last had a value of 1.

7 Experimental Results

We trained and evaluated (using a 10-fold cross-validation process) a set of models for each of the two system orientations shown in Figure 3 and for each of the 5 feature subsets shown in Table 2. The results on the per-frame classification task, including the ROC

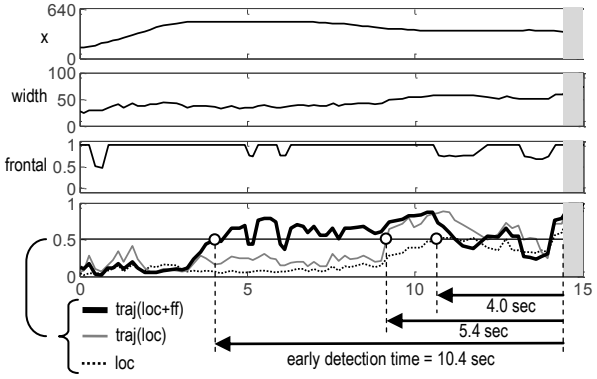


Figure 5. Example predictions for three different models.

curves for the different models are presented and discussed in more detail in Appendix A.

At runtime, the system uses these frame-based models to predict across time the likelihood that a given agent intends to engage (see Figure 5). In this context, an evaluation that counts the errors per person (*i.e.,* per trace), rather than errors per frame is more informative. Furthermore, since early detection is important for supporting a natural engagement process, an informative evaluation should also capture how soon a model can detect a positive engagement intention (see Figure 5).

Making decisions about an agent's engagement intentions typically involves comparing the probability of engagement against a preset threshold. Given a threshold, we can compute for each model the number of false-positives at the trace level: if the prediction exceeds the threshold at any point in the trace, we consider that a positive detection. We note that, if we aim to detect people who will actually engage, there are no false negatives at the trace level. The system can use the machine learned models in conjunction with the previous heuristic (a user is detected standing in front of the system), to eventually detect when people engage. Also, given a threshold, we can identify how early a model can correctly detect the intention to engage (compared to the existing F-formation heuristic that defined the moment of engagement in the training data). These durations are illustrated for a threshold of 0.5 in Figure 5, and are referred to in the sequel as *early detection time*. By varying the threshold between 0 and 1, we can obtain a profile that links the false-positive rate at the trace level to how early the system can detect engagement, *i.e.* to the mean early detection time.

Figure 6 shows the false-positive rate as a function of the mean early detection time for models trained using each of the five feature subsets shown in Table 2, in the *side* orientation. The model that uses only location information (including the size of the face and proximity) performs worst. Adding automatically extracted information about attention leads only to a marginal improvement. However, adding information about the tra-

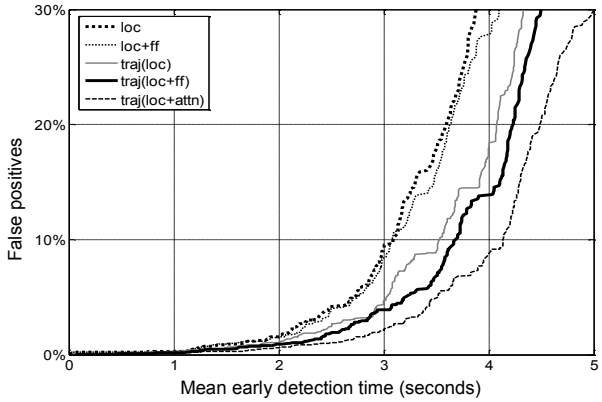


Figure 6. False-positives vs. early detection time (side).

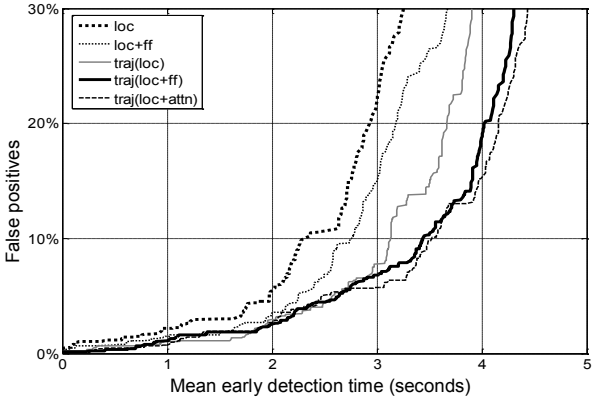


Figure 7. False-positives vs. early detection time (front).

	False positive rate					
Model	EDT=1	EDT=2	EDT=2.5	EDT=3	EDT=3.5	EDT=4
loc	0.31%	1.6%	4.3%	9.4%	18.4%	32.6%
loc+ff	0.31%	1.5%	4.1%	8.7%	18.3%	28.6%
traj(loc)	0.31%	1.1%	2.6%	4.8%	9.3%	18.6%
traj(loc+ff)	0.15%	0.9%	2.0%	4.0%	7.1%	14.3%
traj(loc+attn)	0.26%	0.6%	1.1%	2.2%	5.1%	8.9%

Table 3. *False-positive rate at different EDT (side)

	False positive rate					
Model	EDT=1	EDT=2	EDT=2.5	EDT=3	EDT=3.5	EDT=4
loc	2.3%	5.8%	11.3%	23.0%	35.2%	44.5%
loc+ff	1.6%	3.7%	7.3%	15.8%	28.5%	41.7%
traj(loc)	1.1%	3.1%	4.7%	8.2%	15.6%	36.8%
traj(loc+ff)	1.2%	2.7%	4.7%	7.2%	10.9%	19.8%
traj(loc+attn)	0.8%	2.9%	5.4%	5.4%	10.3%	16.1%

Table 5. *False-positive rate at different EDT (front)

	Early detection time			
Model	FP=2.5%	FP=5%	FP=10%	FP=20%
loc	2.18	2.72	3.09	3.59
loc+ff	2.25	2.74	3.08	3.63
traj(loc)	2.51	3.03	3.53	4.07
traj(loc+ff)	2.68	3.20	3.68	4.22
traj(loc+attn)	3.08	3.52	4.13	4.49

Table 4.*Early detection times at different FP rates (side).

	Early detection time			
Model	FP=2.5%	FP=5%	FP=10%	FP=20%
loc	1.14	1.97	2.29	2.92
loc+ff	1.70	2.25	2.74	3.18
traj(loc)	1.93	2.57	3.13	3.66
traj(loc+ff)	1.99	2.64	3.44	4.02
traj(loc+attn)	1.97	2.47	3.52	4.15

Table 6 * Early detection times at different FP rates (front).

***s**haded cells in Tables 3-6 show statistically significant improvements in performance ($p<0.05$) over the corresponding model that uses the immediately previous feature set (*e.g.* the cell right above). The traj(loc), traj(loc+ff), traj(loc+attn) always statistically significantly ($p<0.05$) improve upon the loc models

jectory of location and of attention, leads to larger cumulative gains. Adding the more accurate (manually tagged) information about attention yields the best model. The relative performance of these models (which can be observed at the frame-level in Appendix A) confirms our expectations and the importance of trajectory features (both spatial and attentional) in detecting engagement intentions. The results also indicate that the differences, and hence the importance of these features, are larger when trying to detect engagement early on, *i.e.* at larger early detection times. Tables 3 and 4 further highlight these differences. For instance, when detecting engagement intentions at a mean early detection above 3 seconds, the model that uses trajectory information, traj(loc+ff), decreases the false positive rate by a factor of 3 compared to the location-only model.

Figure 7 and Tables 5 and 6 show the results for the *front* orientation. The relative trends are similar to those observed in the *side* orientation, highlighting again the importance of trajectory features. At the same time, the models are performing slightly worse in absolute terms, which is consistent with the increased difficulty of the task. Several contributing factors can be identified in Figure 3: people may simply pass by in closer proximity to the system; people who come from the corridor are generally frontally oriented towards the system, making frontal face cues less informative; and finally, people who will engage need to deviate less from the regular trajectory of people who are just passing by.

Next, we review how well the models trained generalize across the two different setups, by evaluating the trajectory models traj(loc+ff) across the two datasets. The results indicate that the models are attuned to the dataset they are trained on (see Figure 7). As we discussed earlier, we expect this result given the different geometry of the relative trajectories of engagement in the two orientations. These results highlight the importance of learning in situ, and show that the proposed approach can be used to learn the specific patterns of engagement in a given environment automatically, without explicit developer supervision.

Finally, we performed an error analysis. We focused on the *side* orientation and visually inspected the 79 (4%) false-positive errors committed by the traj(loc+ff)

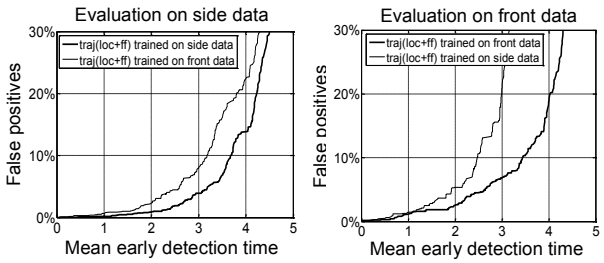


Figure 7. Model evaluation across orientations.

model when using a threshold corresponding to a mean early detection time of 3 seconds. This analysis indicates that in 22 out of these 79 errors (28%) the person did actually exhibit behaviors consistent with an intention to engage the system, such as stopping by or turning around after passing the system, and approaching and maintaining sustained attention for a significant amount of time. These cases represent false-negatives committed by our conservative F-formation heuristic with respect to engagement intention; the user did not approach close enough for the system to trigger engagement. The actual false-positive rate of the trained model is therefore 2.9% rather than 4%. The system was able to correctly identify these cases because the behavioral patterns are similar to the ones exhibited by people who did approach close enough for the heuristic detector to fire. We plan to assess the false-negative rate of the current heuristic more closely and explore how many false negatives are actually recovered by the trained model. This analysis will require that multiple judges assess engagement intentions on all 3274 traces.

8 Summary and Future Work

We described an approach to learning engagement intentions in a situated conversational system. The proposed models fit into a larger framework for supporting multiparty, situated engagement and open-world dialog (Bohus and Horvitz, 2009a; 2009b). Experimental results indicate that a system using the proposed approach can learn to detect engagement intentions at low false positive rates up to 3-4 seconds prior to the actual moment of engagement. The models leverage features that capture spatiotemporal and attentional cues that are tuned to the specifics of the physical environment in which the system operates. Furthermore, the models can be trained in previously unseen environments, without any explicit developer supervision.

We believe the methods and results described represent a first step towards supporting fluid, natural engagement in open-world interaction. Numerous challenges remain. While we confirmed the importance of spatiotemporal and attentional features in detecting engagement intentions, we believe that leveraging additional and more accurate sensory information (*e.g.* body pose, eye gaze, more accurate depth information, agent identity coupled with longer term memory features) may improve performance. Secondly, while the current models where trained in a batch fashion, the proposed method naturally lends itself to an online approach, where the system starts with a prior model for detecting engagement intentions, and refines this model online. More importantly, rather than just learning to detect engagement intentions, we plan to focus on the more general problem of controlling the engagement process: how should the system time its actions (*i.e.* gaze and sustained attention, smiles, greeting, etc.) to create natural, fluid engagements in the open world. Introducing mobility to dialog systems brings yet another interesting dimension to this problem: how can a mobile system, such as a robot, detect engagement intentions and respond to support a natural engagement process? We believe that there is great opportunity to address these challenges by learning predictive models from data.

References

M. Argyle and M. Cook, 1976, *Gaze and Mutual Gaze*, Cambridge University Press, New York

D. Bohus and E. Horvitz, 2009a, *Open-World Dialog: Challenges, Directions and Prototype*, to appear in KRPD'09, Pasadena, CA

D. Bohus and E. Horvitz, 2009b, *Computational Models for Multiparty Engagement in Open-World Dialog*, submitted to SIGdial'09, London, UK.

E. Goffman, 1963, *Behaviour in public places: notes on the social order of gatherings*, The Free Press, New York

E.T. Hall, 1966, *The Hidden Dimension: man's use of space in public and private*, New York: Doubleday.

A. Kendon, 1990a, *A description of some human greetings*, Conducting Interaction: Patterns of behavior in focused encounters, Studies in International Sociolinguistics, Cambridge University Press

A. Kendon, 1990b, *Spatial organization in social encounters: the F-formation system*, Conducting Interaction: Patterns of behavior in focused encounters, Studies in International Sociolinguistics, Cambridge University Press

M.P. Michalowski, S. Sabanovic, and R. Simmons, *A spatial model of engagement for a social robot*, in 9th IEEE Workshop on Advanced Motion Control, pp. 762-767

C. Peters, C. Pelachaud, E. Bevacqua, and M. Mancini, 2005a, *A model of attention and interest using gaze behavior*, Lecture Notes in Computer Science, pp. 229-240.

C. Peters, 2005b, *Direction of Attention Perception for Conversation Initiation in Virtual Environments*, in Intelligent Virtual Agents, 2005, pp. 215-228.

C.L. Sidner, C.D. Kidd, C. Lee, and N. Lesh, 2004, *Where to Look: A Study of Human-Robot Engagement*, IUI'2004, pp. 78-84, Madeira, Portugal

C.L. Sidner, C. Lee, C.D. Kidd, N. Lesh, and C. Rich, 2005, *Explorations in engagement for humans and robots*, Artificial Intelligence, 166 (1-2), pp. 140-164

R. Vertegaal, R. Slagter, G.C.v.d.Veer, and A. Nijholt, 2001, *Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes*, CHI'01

Appendix A. Per-frame evaluation of maximum entropy models for detecting engagement intentions

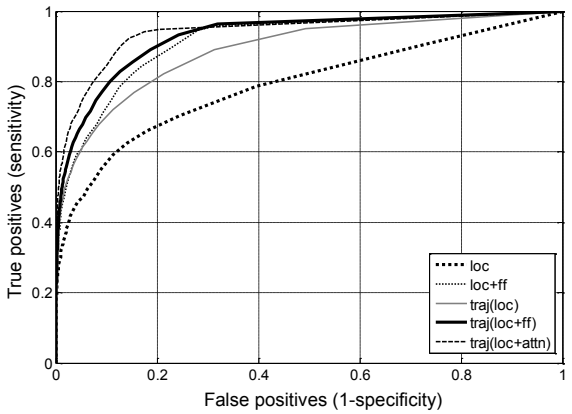


Figure 1. Per-frame ROC for side orientation models

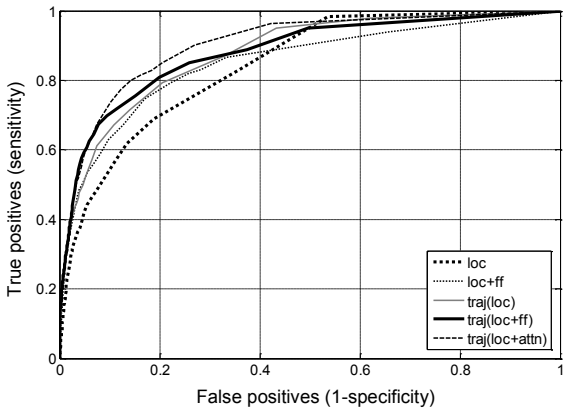


Figure 2. Per-frame ROC for front orientation models

Model	Avg. log-likelihood			Hard error		
	Base	Train	CV	Base	Train	CV
loc	-0.1651	-0.1222	-0.1259	3.91%	3.22%	3.25%
loc+ff	-0.1651	-0.0962	-0.0984	3.91%	3.01%	3.07%
traj(loc)	-0.1651	-0.0947	-0.1073	3.91%	2.88%	3.06%
traj(loc+ff)	-0.1651	-0.0836	-0.0904	3.91%	2.69%	2.85%
traj(loc+attn)	-0.1651	-0.0765	-0.0810	3.91%	2.47%	2.56%

Table 1. Baseline, training-set and cross-validation performance (data average log-likelihood and classification error) for side orientation models

Model	Avg. log-likelihood			Hard error		
	Base	Train	CV	Base	Train	CV
loc	-0.1875	-0.1451	-0.1498	4.63%	4.58%	4.72%
loc+ff	-0.1875	-0.1326	-0.1392	4.63%	4.22%	4.39%
traj(loc)	-0.1875	-0.1262	-0.1338	4.63%	3.99%	4.24%
traj(loc+ff)	-0.1875	-0.1159	-0.1298	4.63%	3.91%	4.38%
traj(loc+attn)	-0.1875	-0.1150	-0.1267	4.63%	4.04%	4.47%

Table 2. Baseline, training-set and cross-validation performance (data average log-likelihood and classification error) for front orientation models

Turn-Yielding Cues in Task-Oriented Dialogue

Agustín Gravano
Department of Computer Science
Columbia University
New York, NY, USA
agus@cs.columbia.edu

Julia Hirschberg
Department of Computer Science
Columbia University
New York, NY, USA
julia@cs.columbia.edu

Abstract

We examine a number of objective, automatically computable TURN-YIELDING CUES — distinct prosodic, acoustic and syntactic events in a speaker's speech that tend to precede a smooth turn exchange — in the Columbia Games Corpus, a large corpus of task-oriented dialogues. We show that the likelihood of occurrence of a turn-taking attempt from the interlocutor increases linearly with the number of cues conjointly displayed by the speaker. Our results are important for improving the coordination of speaking turns in interactive voice-response systems, so that systems can correctly estimate when the user is willing to yield the conversational floor, and so that they can produce their own turn-yielding cues appropriately.

1 Introduction and Previous Research

Users of state-of-the-art interactive voice response (IVR) systems often find interactions with these systems to be unsatisfactory. Part of this reaction is due to deficiencies in speech recognition and synthesis technologies, but some can also be traced to coordination problems in the exchange of speaking turns between system and user (Ward et al., 2005; Raux et al., 2006). Users are not sure when the system is ready to end its turn, and systems are not sure when users are ready to relinquish theirs. Currently, the standard method for determining when a user is willing to yield the conversational floor is to wait for a silence longer than a prespecified threshold, typically ranging from 0.5 to 1 second (Ferrer et al., 2003). However, this strategy is rarely used by humans, who rely instead on cues from sources such as syntax, acoustics and prosody to anticipate turn transitions (Yngve, 1970). If such TURN-YIELDING CUES could be modeled and incorporated in IVR systems, it should be possible to make faster, more accurate turn-taking decisions, thus leading to a more fluent interaction. Additionally, a better understanding of the mechanics of turn-taking could be used to vary the speech output of IVR systems to (i) produce turn-yielding cues when the system is finished speaking and the user is expected to speak next, and (ii) avoid producing such cues when the system has more things to say. In this paper we examine the existence of turn-yielding cues in a large corpus of task-oriented dialogues in Standard American English (SAE).

The question of what types of cues humans exploit for engaging in synchronized conversation has been addressed by several studies. Duncan (1972, inter alia) conjectures that speakers display complex signals at turn endings, composed of one or more discrete turn-yielding cues, such as the completion of a grammatical clause, or any phrase-final intonation other than a plateau. Duncan also hypothesizes that the likelihood of a turn-taking attempt by the listener increases linearly with the number of such cues conjointly displayed by the speaker. Subsequent studies have investigated some of these hypotheses (Ford and Thompson, 1996; Wennerstrom and Siegel, 2003). More recent studies have investigated how to improve IVR system's the turn-taking decisions by incorporating some of the features found to correlate with turn endings (Ferrer et al., 2003; Atterer et al., 2008; Raux and Eskenazi, 2008). All of these models are shown to improve over silence-based techniques for predicting turn endings, motivating further research. In this paper we present results

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 253–261,
Queen Mary University of London, September 2009.

of a large, corpus-based study of turn-yielding cues in the Columbia Games Corpus which verifies some of Duncan's hypotheses and adds additional cues to turn-taking behavior.

2 Materials and Method

The materials for our study are taken from the Columbia Games Corpus (Gravano, 2009), a collection of 12 spontaneous task-oriented dyadic conversations elicited from 13 native speakers of SAE. In each session, two subjects were paid to play a series of computer games requiring verbal communication to achieve joint goals of identifying and moving images on the screen, while seated in a soundproof booth divided by a curtain to ensure that all communication was verbal. The subjects' speech was not restricted in any way, and the games were not timed. The corpus contains 9 hours of dialogue, which were orthographically transcribed; words were time-aligned to the source by hand. Around 5.4 hours have also been intonationally transcribed using the ToBI framework (Beckman and Hirschberg, 1994).

We automatically extracted a number of acoustic features from the corpus using the Praat toolkit (Boersma and Weenink, 2001), including pitch, intensity and voice quality features. Pitch slopes were computed by fitting least-squares linear regression models to the F_0 track extracted from given portions of the signal. Part-of-speech (POS) tags were labeled automatically using Ratnaparkhi's maxent tagger trained on a subset of the Switchboard corpus in lower-case with all punctuation removed, to simulate spoken language transcripts. All speaker normalizations were calculated using z-scores: $z = (x - \mu)/\sigma$, where x is a raw measurement, and μ and σ are the mean and standard deviation for a speaker.

For our turn-taking studies, we define an INTER-PAUSAL UNIT (IPU) as a maximal sequence of words surrounded by silence longer than 50 ms.[1] A TURN then is defined as a maximal sequence of IPUs from one speaker, such that between any two adjacent IPUs there is no speech from the interlocutor. Boundaries of IPUs and turns are computed automatically from the time-aligned transcriptions. Two trained annotators classified each turn transition in the corpus using a labeling scheme adapted from Beattie (1982) that identifies, inter alia, SMOOTH SWITCHES — transitions from speaker A to speaker B such that (i) A manages to complete her utterance, and (ii) no overlapping speech occurs between the two conversational turns. Additionally, all continuations from one IPU to the next within the same turn were labeled automatically as HOLD transitions. The complete labeling scheme is shown in the Appendix.

Our general approach consists in contrasting IPUs immediately preceding smooth switches (**S**) with IPUs immediately preceding holds (**H**). (Note that in this paper we consider only non-overlapping exchanges.) We hypothesize that turn-yielding cues are more likely to occur before **S** than before **H**. It is important to emphasize the optionality of all turn-taking phenomena and decisions: For **H**, turn-yielding cues — whatever their nature — may still be present; and for **S**, they may sometimes be absent. However, we hypothesize that their likelihood of occurrence should be much higher before **S**. Finally, note that we do **not** make claims regarding whether speakers consciously produce turn-yielding cues, or whether listeners consciously perceive and/or use them to aid their turn-taking decisions.

3 Individual Turn-Yielding Cues

Figures 1 and 2 show the speaker-normalized mean of a number of objective, automatically computed variables for IPUs preceding **S** and **H**. In all cases, one-way ANOVA and Kruskal-Wallis tests reveal significant differences (at $p < 0.001$) between the two groups. We we discuss these results in detail below.

3.1 Intonation

The literature contains frequent mention of the propensity of speaking turns to end in any intonation contour **other than** a plateau (a sustained pitch level, neither rising nor falling). We first analyze the categorical prosodic labels in the portion of the Columbia Games Corpus annotated using the ToBI annotations. We tabulate the phrase

	S		**H**	
H-H%	**484**	**22.1%**	513	9.1%
[!]H-L%	289	13.2%	**1680**	**29.9%**
L-H%	309	14.1%	646	11.5%
L-L%	**1032**	**47.2%**	1387	24.7%
No boundary tone	16	0.7%	**1261**	**22.4%**
Other	56	2.6%	136	2.4%
Total	2186	100%	5623	100%

Table 1: ToBI phrase accents and boundary tones.

[1] 50 ms was identified empirically to avoid stopgaps.

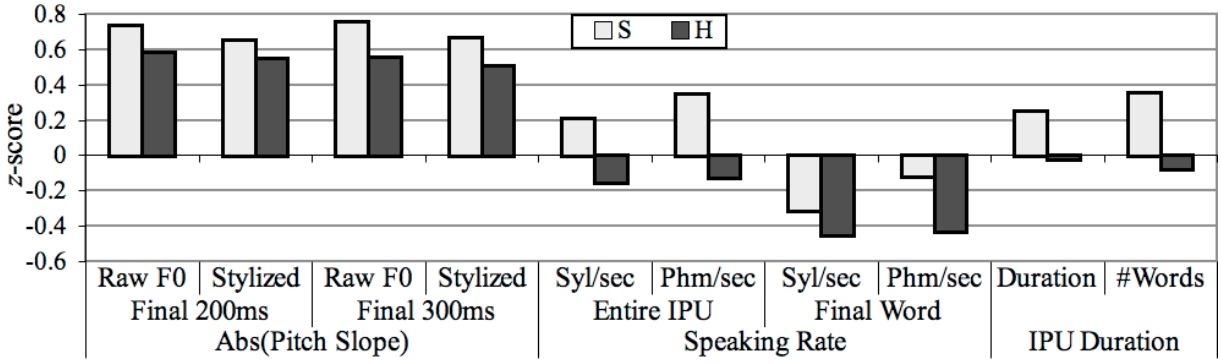


Figure 1: Individual turn-yielding cues: intonation, speaking rate and IPU duration.

accent and boundary tone labels assigned to the end of each IPU, and compare their distribution for the **S** and **H** turn exchange types, as shown in Table 1. A chi-square test indicates that there is a significant departure from a random distribution ($\chi^2 = 1102.5$, $df = 5$, $p \approx 0$). Only 13.2% of all IPUs immediately preceding a smooth switch (**S**) — where turn-yielding cues are most likely present — end in a plateau ([!]H-L%); most of the remaining ones end in either a falling pitch (L-L%) or a high rise (H-H%). For IPUs preceding a hold (**H**) the counts approximate a uniform distribution, with the plateau contours being the most common, supporting the hypothesis that this contour functions as a TURN-HOLDING CUE (that is, a cue that typically prevents turn-taking attempts from the listener). The high counts for the falling contour preceding a hold (24.7%) may be explained by the fact that, as discussed above, taking the turn is optional for the listener, who may choose not to act despite hearing some turn-yielding cues. It is not entirely clear what the role is of the low-rising contour (L-H%), as it occurs in similar proportions before **S** and before **H**. Finally, we note that the absence of a boundary tone works as a strong indication that the speaker has not finished speaking, since nearly all (98%) IPUs without a boundary tone precede a hold transition.

Next, we examine four objective acoustic approximations of this perceptual feature: the absolute value of the speaker-normalized F_0 slope, both raw and stylized, computed over the final 200 and 300 ms of each IPU. The case of a plateau corresponds to a value of F_0 slope close to zero; the other case, of either a rising or a falling pitch, corresponds to a high absolute value of F_0 slope. As shown in Figure 1, we find that the final slope before **S** is significantly higher than before **H** in all four cases. These findings provide additional support to the hypothesis that turns tend to end in falling and high-rising final intonations, and provide automatically identifiable indicators of this turn-yielding cue.

3.2 Speaking rate

Duncan (1972) hypothesizes a "drawl on the final syllable or on the stressed syllable of a terminal clause" [p. 287] as a turn-yielding cue, which would probably correspond to a noticeable decrease in speaking rate. We examine this hypothesis in our corpus using two common definitions of speaking rate: syllables per second and phonemes per second. Syllable and phoneme counts were estimated from dictionary lookup, and word durations were extracted from the manual orthographic alignments. Figure 1 shows that both measures, computed over either the whole IPU or its final word, are significantly higher before **S** than before **H**, which indicates an **increase** in speaking rate before turn boundaries rather than Duncan's hypothesized drawl.

Furthermore, the speaking rate is, in both cases (before **S** and before **H**), significantly slower on the final word than over the whole IPU, a finding that is in line with phonological theories that predict a segmental lengthening near prosodic phrase boundaries (Wightman et al., 1992). This finding may indeed correspond to the drawl or lengthening described by Duncan before turn boundaries. However, it seems to be the case — at least for our corpus — that the final lengthening tends to occur at all phrase final positions, not just at turn endings. In fact, our results indicate that the final lengthening is more prominent in turn-medial IPUs than in turn-final ones.

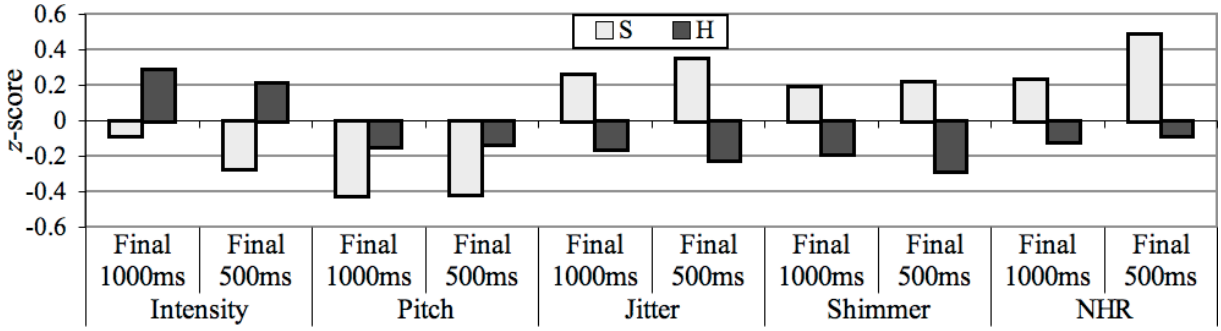


Figure 2: Individual turn-yielding cues: intensity, pitch and voice quality.

3.3 IPU duration and acoustic cues

In the Columbia Games Corpus, we find that turn-final IPUs tend to be significantly longer than turn-medial ones, both when measured in seconds and in number of words (Figure 1). This suggests that IPU duration could function as a turn-yielding cue, supporting similar findings in perceptual experiments by Cutler and Pearson (1986).

We also find that IPUs followed by **S** have a mean intensity significantly lower than those followed by **H** (computed over the IPU-final 500 and 1000 ms, see Figure 2). Also, the differences increase when moving towards the end of the IPU. This suggests that speakers tend to lower their voices when approaching potential turn boundaries, whereas they reach turn-internal pauses with a higher intensity.

Phonological theories conjecture a declination in the pitch level, which tends to decrease gradually within utterances, and across utterances within the same discourse segment, as a consequence of a gradual compression of the pitch range (Pierrehumbert and Hirschberg, 1990). For conversational turns, then, we would expect to find that speakers tend to lower their pitch level as they reach potential turn boundaries. This hypothesis is verified by the dialogues in our corpus, where we find that IPUs preceding **S** have a significantly lower mean pitch than those preceding **H** (Figure 2). In consequence, pitch level may also work as a turn-yielding cue.

Next we examine three acoustic features associated with the perception of voice quality: jitter, shimmer and noise-to-harmonics ratio (NHR) (Bhuta et al., 2004), computed over the IPU-final 500 and 1000 ms (Figure 2). We compute jitter and shimmer only over voiced frames for improved robustness. For all three features, the mean value for IPUs preceding **S** is significantly higher than for IPUs preceding **H**, with the difference increasing towards the end of the IPU. Therefore, voice quality seems to play a clear role as a turn-yielding cue.

3.4 Lexical cues

Stereotyped expressions such as *you know* or *I think* have been proposed in the literature as lexical turn-yielding cues. However, in the Games Corpus we find that none of the most frequent IPU-final unigrams and bigrams, both preceding **S** and **H**, correspond to such expressions (see Table A.1 in the Appendix). Instead, such unigrams and bigrams are specific to the computer games in which the subjects participated. For example, the game objects tended to be spontaneously described by subjects from top to bottom and from left to right, as shown in the following excerpt (pauses are indicated with #):

A: *I have a blue lion on top # with a lemon in the bottom left # and a yellow crescent moon in- # i- # in the bottom right*
B: *oh okay* [...]

In consequence, bigrams such as *lower right* and *bottom right* are common before **S**, while *on top* or *bottom left* are common before **H**. These are all task-specific lexical constructions and do not constitute stereotyped expressions in the traditional sense.

Also very common among the most frequent IPU-final expressions are AFFIRMATIVE CUE WORDS — heavily overloaded words, such as *okay* or *yeah*, that are used both to initiate and to end discourse segments, among other functions (Gravano et al., 2007). The occurrence of these words does not constitute a turn-yielding or turn-holding cue *per se*; rather, additional contextual, acoustic and prosodic information is needed to disambiguate their meaning.

While we do not find clear examples of lexical turn-yielding cues in our task-oriented corpus, we do find two lexical turn-holding cues: word fragments (e.g., *incompl-*) and filled pauses (e.g., *uh*, *um*). Of the 8123 IPUs preceding **H**, 6.7% end in a word fragment, and 9.4% in a filled pause. By constrast, only 0.3% of the 3246 IPUs preceding **S** end in a word fragment, and 1% in a filled pause. These differences suggest that, after either a word fragment or a filled pause, the speaker is much more likely to intend to continue holding the floor. This notion of disfluencies functioning as a turn-taking cue has been studied by Goodwin (1981), who shows that they may be used to secure the listener's attention at turn beginnings.

3.5 Textual completion

Several authors (Duncan, 1972; Ford and Thompson, 1996; Wennerstrom and Siegel, 2003) claim that some form of syntactic or semantic completion, independent of intonation and interactional import, functions as a turn-yielding cue. Although some call this *syntactic completion*, since all authors acknowledge the need for semantic and discourse information in judging it, we choose the more neutral term TEXTUAL COMPLETION for this phenomenon. We annotated a portion of our corpus with respect to textual completion and trained a machine learning (ML) classifier to automatically label the whole corpus. From these annotations we then examined how textual completion labels relate to turn-taking categories in the corpus.

3.5.1. Manual labeling: In conversation, listeners judge textual completion incrementally and without access to later material. To simulate these conditions in the labeling task, annotators were asked to judge the textual completion of a turn up to a target pause from the written transcript alone, without listening to the speech. They were allowed to read the transcript of the full previous turn by the other speaker (if any), but they were not given access to anything after the target pause. These are two sample tokens:

A: *the lion's left paw our front*
B: *yeah and it's th- right so the*

A: *and then a tea kettle and then the wine*
B: *okay well I have the big shoe and the wine*

We selected 400 tokens at random from the Games Corpus; the target pauses were also chosen at random. Three annotators labeled each token independently as either complete or incomplete according to these guidelines: *Determine whether you believe what speaker B has said up to this point could constitute a complete response to what speaker A has said in the previous turn/segment. Note: If there are no words by A, then B is beginning a new task, such as describing a card or the location of an object.* To avoid biasing the results, annotators were not given the turn-taking labels of the tokens. Inter-annotator reliability is measured by Fleiss' κ at 0.814, which corresponds to the 'almost perfect' agreement category. The mean pairwise agreement between the three subjects is 90.8%. For the cases in which there is disagreement between the three annotators, we adopt the MAJORITY LABEL as our gold standard; that is, the label chosen by two annotators.

3.5.2. Automatic classification: Next, we trained a ML model using the 400 manually annotated tokens as training data to automatically classify all IPUs in the corpus as either complete or incomplete. For each IPU we extracted a number of lexical and syntactic features from the current turn up to the IPU itself: lexical identity of the IPU-final word (*w*); POS tags and simplified POS tags (N, V, Adj, Adv, Other) of *w* and of the IPU-final bigram; number of words in the IPU; a binary flag indicating if *w* is a word fragment; size and type of the biggest (*bp*) and smallest (*sp*) phrase that end in *w*; binary flags indicating if each of *bp* and *sp* is a major phrase (NP, VP, PP, ADJP, ADVP); binary flags indicating if *w* is the head of each of *bp* and *sp*. We chose these features in order to capture as much lexical and syntactic information as possible from the transcripts. The syntactic features were computed using two different parsers: the Collins statistical parser (Collins, 2003) and CASS, a partial parser especially designed for use with noisy text (Abney, 1996). We experimented with the learners listed in Table 2, using the implementations provided in the WEKA ML toolkit (Witten and Frank, 2000). Table 2 shows the accuracy of the majority-class baseline and of each classifier, using 10-fold cross validation on the 400 training data points, and the mean pairwise agreement by the three human labelers. The linear-kernel support-vector-machine (SVM) classifier achieves the highest accuracy, significantly outperforming the baseline, and approaching the mean agreement of human labelers.

Classifier	Accuracy
Majority-class ('complete')	55.2%
C4.5 (decision trees)	55.2%
Ripper (propositional rules)	68.2%
Bayesian networks	75.7%
SVM, RBF kernel ($c = 1, \varepsilon = 10^{-12}$)	78.2%
SVM, linear kernel ($c = 1, \varepsilon = 10^{-12}$)	80.0%
Human labelers (mean agreement)	90.8%

Table 2: Textual completion: ML results.

3.5.3. Results: First we examine the tokens that were manually labeled by the human annotators. Of the 100 tokens followed by **S**, 91 were labeled textually complete, a significantly higher proportion than the 42% followed by **H** that were labeled complete (χ^2=51.7, df=1, $p\approx$0). Next, we used our highest performing classifier, the linear-kernel SVM, to automatically label all IPUs in the corpus. Of the 3246 IPUs preceding **S**, 2649 (81.6%) were labeled textually complete, and about half of all IPUs preceding **H** (4272/8123, or 52.6%) were labeled complete. The difference is also significant ($\chi^2 = 818.7$, $df = 1$, $p \approx 0$). These results suggest that textual completion as defined above constitutes a necessary, but not sufficient, turn-yielding cue.

4 Combining Turn-Yielding Cues

So far, we have shown strong evidence supporting the existence of individual acoustic, prosodic and textual turn-yielding cues. Now we shift our attention to the manner in which they combine together to form more complex turn-yielding signals. For each individual cue type, we choose two or three features shown to correlate strongly with smooth switches, as shown in Table 3 (e.g., the speaking rate cue is represented by two automatic features: syllables and phonemes per second over the whole IPU). We consider a cue c to be PRESENT on IPU u if, for any feature f modeling c, the value of f on u is closer to f_S than to f_H, where f_S and f_H are the mean values of f across all IPUs preceding **S** and **H**, respectively. Otherwise, we say c is ABSENT on u. Also, we automatically annotate all IPUs in the corpus for textual completion using the linear-kernel SVM classifier described in Section 3.5. IPUs classified as complete are considered to bear the textual completion turn-yielding cue.

We first analyze the frequency of occurrence of conjoined individual turn-yielding cues. Table 4 shows the top frequencies of complex turn-yielding cues for IPUs immediately before smooth switches (**S**) and holds (**H**). The most frequent cases before **S** correspond to all, or almost all, cues present at once. For IPUs preceding a hold (**H**), the opposite is true: those with no cues, or with just one or two, represent the most frequent cases.

Individual cues	**Automatic features**
Intonation	Abs(F_0 slope) over IPU-final 200 ms Abs(F_0 slope) over IPU-final 300 ms
Speaking rate	Syllables per second over whole IPU Phonemes per second over whole IPU
Intensity level	Mean intensity over IPU-final 500 ms Mean intensity over IPU-final 1000 ms
Pitch level	Mean pitch over IPU-final 500 ms Mean pitch over IPU-final 1000 ms
IPU duration	IPU duration in ms Number of words in IPU
Voice quality	Jitter over IPU-final 500 ms Shimmer over IPU-final 500 ms NHR over IPU-final 500 ms

Table 3: Features used to estimate the presence of individual turn-yielding cues.

S		**H**	
Cues	Count	Cues	Count
1234567	267	...4...	392
.234567	226	7	247
1234.67	138	.....	223
.234.67	109	...4..7	218
.23..67	98	...45..	178
..34567	94	.2....7	166
123..67	93	1234.67	163
.2.4567	73	.2..5.7	157
	...		...
Total	3246	Total	8123

Table 4: Top frequencies of complex turn-yielding cues for IPUs preceding **S** and **H**. A digit indicates the presence of a specific cue; a dot, its absence. 1: Intonation; 2: Speaking rate; 3: Intensity level; 4: Pitch level; 5: IPU duration; 6: Voice quality; 7: Textual completion.

Table 5 shows the same results, now grouping together all IPUs with the same **number** of cues, independently of the cue types. Again, we observe that larger proportions of IPUs preceding **S** present more conjoined cues than IPUs preceding **H**.

Next we look at how the likelihood of a turn-taking attempt varies with respect to the number of individual cues displayed by the speaker, a relation hypothesized to be linear by Duncan (1972). Figure 3 shows the proportion of IPUs with 0-7 cues present that are followed by a turn-taking attempt from the interlocutor.[2] The dashed line cor-

[2] The proportion of turn-taking attempts is computed for each cue count as the number of **S** and **PI** divided by the number of **S**, **PI**, **H** and **BC**, according to our labeling scheme.

Cue count	S		H	
0	4	0.1%	223	2.7%
1	52	1.6%	970	11.9%
2	241	7.4%	1552	19.1%
3	518	16.0%	1829	22.5%
4	740	22.8%	1666	20.5%
5	830	25.6%	1142	14.1%
6	594	18.3%	611	7.5%
7	267	8.2%	130	1.6%
Total	3246	100%	8123	100%

Table 5: Distribution of the number of turn-yielding cues displayed in IPUs preceding smooth switches (**S**) and hold transitions (**H**).

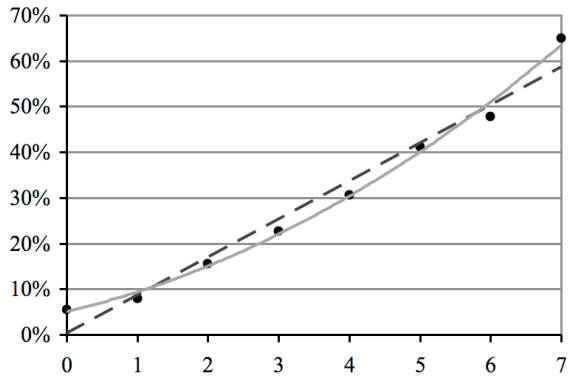


Figure 3: Percentage of turn-taking attempts from the listener (either **S** or **PI**) following IPUs containing 0-7 turn-yielding cues.

responds to a linear model fitted to the data (Pearson's correlation test: $r^2 = 0.969$), and the continuous line, to a quadratic model ($r^2 = 0.995$). The high correlation coefficient of the linear model supports Duncan's hypothesis, that the likelihood of a turn-taking attempt by the interlocutor increases linearly with the number of individual cues displayed by the speaker. However, an ANOVA test reveals that the quadratic model fits the data significantly better than the linear model ($F(1, 5) = 23.01$; $p = 0.005$), even though the curvature of the quadratic model is only moderate, as can be observed in the figure.

5 Speaker Variation

To investigate possible speaker dependence in our turn-yielding cues, we examine evidence for each cue for each of our thirteen speakers. Table 6 summarizes this data. For each speaker, a check ($\surd$) indicates that there is significant evidence of the speaker producing the corresponding individual turn-yielding cue (at $p < 0.05$, using the same statistical tests described in the previous sections). Five speakers show evidence of all seven cues,

Speaker	101	102	103	104	105	106	107	108	109	110	111	112	113
Intonation	√	√	√	√	√	√	√	√	√		√	√	
Spk. rate	√	√	√	√	√	√	√	√	√	√	√	√	√
Intensity		√	√	√	√	√	√	√	√	√	√	√	√
Pitch	√	√			√		√		√	√		√	
Completion	√	√	√	√	√	√	√	√	√	√	√	√	√
Voice quality	√	√	√	√	√	√	√	√	√	√	√	√	√
IPU duration	√	√	√	√	√	√	√	√	√	√	√	√	√
LM r^2	**.92**	.93	**.82**	.88	.97	**.96**	.95	.95	**.97**	.91	**.95**	**.97**	.89
QM r^2	**.98**	.95	**.95**	.92	.98	**.98**	.96	.95	**.99**	.94	**.98**	**.99**	.90

Table 6: Summary of results for each individual speaker.

while the remaining eight speakers show either five or six cues. Pitch level is the least reliable cue, present only for seven subjects. Notably, the cues related to speaking rate, textual completion, voice quality, and IPU duration are present for all thirteen speakers.

The two bottom rows in Table 6 show the correlation coefficients (r^2) of linear and quadratic regressions performed on the data from each speaker. In all cases, the coefficients are very high. The fit of the quadratic model is significantly better for six speakers (shown in bold typeface); for the remaining seven speakers, both models provide statistically indistinguishable explanations of the data.

6 Discussion

We have examined seven turn-yielding cues — i.e., seven measurable events that take place with a significantly higher frequency on IPUs preceding smooth turn switches than on IPUs preceding hold transitions. These events may be summarized as follows: (i) a falling or high-rising intonation at the end of the IPU; (ii) an increased speaking rate; (iii) a lower intensity level; (iv) a lower pitch level; (v) a longer IPU duration; (vi) a higher value of three voice quality features: jitter, shimmer, and NHR; and (vii) a point of textual completion. We have also shown that, when several turn-yielding cues occur simultaneously, the likelihood of a subsequent turn-taking attempt by the interlocutor increases in an almost linear fashion.

We propose that these findings can be used to improve some turn-taking decisions of state-of-the-art IVR systems. For example, if a system wishes to yield the floor to a user, it should include in its output as many of the described cues as possible. Conversely, when the user is speaking, the system may detect appropriate moments to take the turn by estimating the presence of turn-

yielding cues at every silence. If the number of detected cues is high enough, then the system should take the turn; otherwise, it should remain silent.

Two assumptions of our study are that turn-yielding cues are binary and all contribute equally to the overall "count". In future research we will explore alternative methods of combining and weighting the different features — by means of multiple linear regression, for example — in order to experiment with more sophisticated models of turn-yielding behavior. We also plan to examine new turn-yielding cues, paying special attention to additional voice quality features, given the promising results obtained for jitter, shimmer and noise-to-harmonics ratio.

7 Acknowledgements

This work was funded in part by NSF IIS-0307905. We thank Stefan Benus, Enrique Henestroza, Elisa Sneed and Gregory Ward, for valuable discussion and for their help in collecting and labeling the data, and the anonymous reviewers for helpful comments and suggestions.

References

S. Abney. 1996. Partial parsing via finite-state cascades. *Journal of Natural Language Engineering*, 2(4):337–344.

M. Atterer, T. Baumann, and D. Schlangen. 2008. Towards incremental end-of-utterance detection in dialogue systems. In *Proceedings of Coling*, Manchester, UK.

G. W. Beattie. 1982. Turn-taking and interruption in political interviews: Margaret Thatcher and Jim Callaghan compared and contrasted. *Semiotica*, 39(1/2):93–114.

M. E. Beckman and J. Hirschberg. 1994. The ToBI annotation conventions. *Ohio State University*.

T. Bhuta, L. Patrick, and J. D. Garnett. 2004. Perceptual evaluation of voice quality and its correlation with acoustic measurements. *Journal of Voice*, 18(3):299–304.

P. Boersma and D. Weenink. 2001. Praat: Doing phonetics by computer. *http://www.praat.org*.

M. J. Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

E. A. Cutler and M. Pearson. 1986. On the analysis of prosodic turn-taking cues. In C. Johns-Lewis, Ed., *Intonation in Discourse*, pp. 139–156. College-Hill.

S. Duncan. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.

L. Ferrer, E. Shriberg, and A. Stolcke. 2003. A prosody-based approach to end-of-utterance detection that does not require speech recognition. In *Proceedings of ICASSP*.

C. E. Ford and S. A. Thompson. 1996. Interactional units in conversation: Syntactic, intonational and pragmatic resources for the management of turns. In E. Ochs, E. A. Schegloff, and S. A. Thompson, Eds., *Interaction and Grammar*, pp. 134–184. Cambridge University Press.

C. Goodwin. 1981. *Conversational Organization: Interaction between Speakers and Hearers*. Academic Press.

A. Gravano, S. Benus, J. Hirschberg, S. Mitchell, and I. Vovsha. 2007. Classification of discourse functions of affirmative words in spoken dialogue. In *Proceedings of Interspeech*.

A. Gravano. 2009. *Turn-Taking and Affirmative Cue Words in Task-Oriented Dialogue*. Ph.D. thesis, Columbia University, New York.

J. Pierrehumbert and J. Hirschberg. 1990. The meaning of intonational contours in the interpretation of discourse. In P. Cohen, J. Morgan, and M. Pollack, Eds., *Intentions in Communication*, pp. 271–311. MIT Pr.

A. Raux and M. Eskenazi. 2008. Optimizing endpointing thresholds using dialogue features in a spoken dialogue system. In *Proceedings of SIGdial*.

A. Raux, D. Bohus, B. Langner, A. W. Black, and M. Eskenazi. 2006. Doing research on a deployed spoken dialogue system: One year of *Let's Go!* experience. In *Proceedings of Interspeech*.

N. G. Ward, A. G. Rivera, K. Ward, and D. G. Novick. 2005. Root causes of lost time and user stress in a simple dialog system. In *Proceedings of Interspeech*.

A. Wennerstrom and A. F. Siegel. 2003. Keeping the floor in multiparty conversations: Intonation, syntax, and pause. *Discourse Processes*, 36(2):77–107.

C. W. Wightman, S. Shattuck-Hufnagel, M. Ostendorf, and P. J. Price. 1992. Segmental durations in the vicinity of prosodic phrase boundaries. *The Journal of the Acoustical Society of America*, 91:1707.

I. H. Witten and E. Frank. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

V. H. Yngve. 1970. On getting a word in edgewise. *Sixth Regional Meeting of the Chicago Linguistic Society*, 6:657–677.

For each turn by speaker S2, where S1 is the other speaker, label S2's turn as follows:

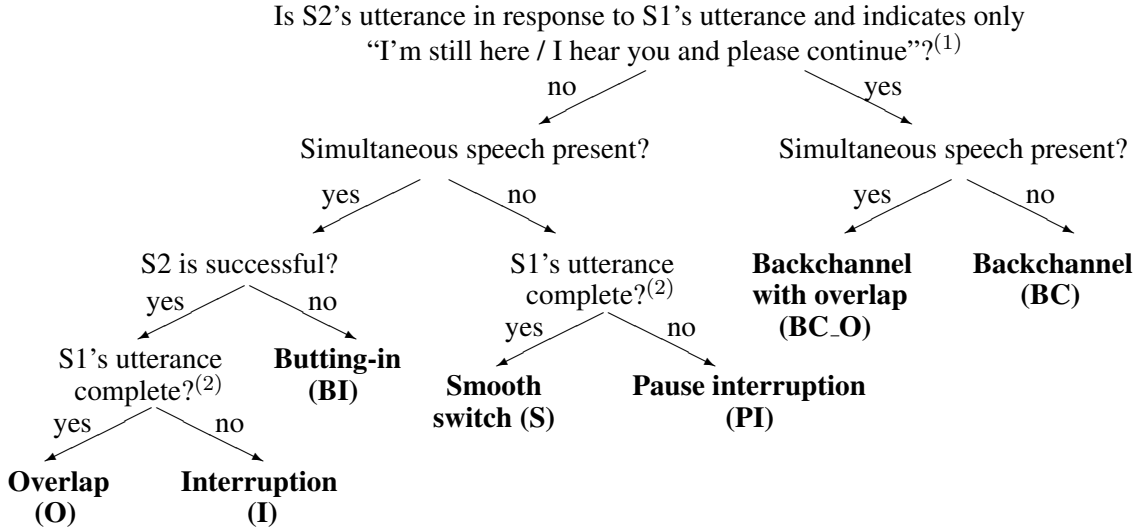


Figure A.1: Turn-taking labeling scheme.

Appendix: Turn-Taking Labeling Scheme

We adopt a slightly modified version of Beattie's (1982) labeling scheme, depicted in Figure A.1. We incorporate backchannels (excluded from Beattie's study) by adding the decision marked (1) at the root of the decision tree, for which we use the annotations described in Gravano et al. (2007). For the decision marked (2), we use Beattie's informal definition of utterance completeness: "Completeness [is] judged intuitively, taking into account the intonation, syntax, and meaning of the utterance" [p. 100]. All continuations from one IPU to the next within the same turn are labeled automatically **H**, for 'hold'. Also, we identify three special cases that do not correspond to actual turn exchanges:

Task beginnings: Turns beginning a new game task are labeled **X1**.

Continuations after BC or BC_O: If a turn t is a continuation after a backchannel b from the other speaker, it is labeled **X2_O** if t and b overlap, or **X2** if not.

Simultaneous starts: Fry (1975) reports that humans require at least 210 ms to react verbally to a verbal stimulus.[3] Thus, if two turns begin within 210 ms of each other, they are most probably connected to preceding events than to one another. In Figure A.2, A_1, A_2 and B_1 represent turns from speakers A and B. Most likely, A_2 is simply a continuation from A_1, and B_1 occurs in response to A_1. Thus, B_1 is labeled with respect to A_1 (not A_2), and A_2 is labeled **X3**.

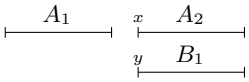


Figure A.2: Simultaneous start ($|y-x| < 210$ms).

S	Count	H	Count
okay	241	*okay*	402
yeah	167	*on top*	172
lower right	85	*um*	136
bottom right	74	*the top*	117
the right	59	*of the*	67
hand corner	52	*blue lion*	57
lower left	43	*bottom left*	56
the iron	37	*with the*	54
the onion	33	*the um*	54
bottom left	31	*yeah*	53
the ruler	30	*the left*	48
mm-hm	30	*and*	48
right	28	*lower left*	46
right corner	27	*uh*	45
the bottom	26	*oh*	45
the left	24	*and a*	45
crescent moon	23	*alright*	44
the lemon	22	*okay um*	43
the moon	20	*the uh*	42
tennis racket	20	*the right*	41
blue lion	19	*the bottom*	39
the whale	18	*I have*	39
the crescent	18	*yellow lion*	37
the middle	17	*the middle*	37
of it	17	*I've got*	34
	...		...
Total	3246	Total	8123

Table A.1: 25 most frequent final bigrams preceding smooth turn switches (**S**) and hold transitions (**H**). (See Section 3.4.)

[3] D. B. Fry. 1975. Simple reaction-times to speech and non-speech stimuli. *Cortex*, 11(4):355-60.

Split Utterances in Dialogue: a Corpus Study

Matthew Purver, Christine Howes,
and **Patrick G. T. Healey**
Department of Computer Science
Queen Mary University of London
Mile End Road, London E1 4NS, UK
{mpurver,chrizba,ph}@dcs.qmul.ac.uk

Eleni Gregoromichelaki
Department of Philosophy
King's College London
Strand, London WC2R 2LS, UK
eleni.gregor@kcl.ac.uk

Abstract

This paper presents a preliminary English corpus study of *split utterances* (*SUs*), single utterances split between two or more dialogue turns or speakers. It has been suggested that SUs are a key phenomenon of dialogue, which this study confirms: almost 20% of utterances were found to fit this general definition, with nearly 3% being the between-speaker case most often studied. Other claims/assumptions in the literature about SUs' form and distribution are investigated, with preliminary results showing: splits can occur within syntactic constituents, apparently at any point in the string; it is unusual for the separate parts to be complete units in their own right; explicit repair of the antecedent does not occur very often. The theoretical consequences of these results for claims in the literature are pointed out. The practical implications for dialogue systems are mentioned too.

1 Introduction

Split utterances (*SUs*) – single utterances split between two or more dialogue turns/speakers – have been claimed to occur regularly in dialogue, especially according to the observations reported in the Conversational Analysis (CA) literature, which is based on the analysis of naturally occurring dialogues. SUs are of interest to dialogue theorists as they are a clear sign of how turns cohere with each other at all levels – syntactic, semantic and pragmatic. They also indicate the radical context-dependency of conversational contributions. Turns can, in general, be highly elliptical and nevertheless not disrupt the flow of the dialogue. SUs are the most dramatic illustration of this: contributions spread across turns/speakers rely crucially on the dynamics of the unfolding context, linguistic and extra-linguistic, in order to guarantee successful processing and production.

Utterances that are split across speakers also present a canonical example of participant coordination in dialogue. The ability of one participant to continue another interlocutor's utterance coherently, both at the syntactic and the semantic level, suggests that both speaker and hearer are highly coordinated in terms of processing and production. The initial speaker must be able to switch to the role of hearer, processing and integrating the continuation of their utterance, whereas the initial hearer must be closely monitoring the grammar and content of what they are being offered so that they can take over and continue in a way that respects the constraints set up by the first part of the utterance. In fact there is (anecdotal) evidence that such constraints are fully respected across speaker and hearer in such utterances (see e.g. Gregoromichelaki et al. (2009)). A large proportion of the CA literature on SUs tries to identify the conditions under which SUs usually occur (see section 2). However, this emphasis seems to miss the important generalisation, confirmed by the present study, that, syntactically, a speaker switch may be able to occur anywhere in a string.

From a theoretical point of view, the implications of the above are that, if such observations have an empirical foundation, the grammar employed by the interlocutors must be able to license and the semantics interpret chunks much smaller than the usual sentence/proposition units. Moreover, these observations have implications for the nature of the grammar itself: dynamic, incremental formalisms seem more amenable to the mod-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 262–271,
Queen Mary University of London, September 2009.

elling of this phenomenon as the switch of roles while syntactic/semantic dependencies are pending can be taken as evidence for direct involvement of the grammar in the successful processing/production of such utterances. Indeed, Poesio and Rieser (to appear) claim that "[c]ollaborative completions ... are among the strongest evidence yet for the argument that dialogue requires *coordination* even at the sub-sentential level" (italics original).

From a psycholinguistic point of view, the phenomenon of SUs is compatible with mechanistic approaches as exemplified by the Interactive Alignment model of Pickering and Garrod (2004) where it is claimed that it should be as easy to complete someone else's sentence as one's own (Pickering and Garrod, 2004, p186). According to this model, speaker and listener ought to be interchangeable at any point. This is also the stance taken by the grammatical framework of Dynamic Syntax (DS) (Kempson et al., 2001; Cann et al., 2005). In DS, parsing and production are taken to employ the same mechanisms, leading to a prediction that split utterances ought to be strikingly natural (Purver et al., 2006). However, from a pragmatic point of view, utterance continuation by another speaker might involve some kind of guessing[1] or preempting the other interlocutor's intended content. It has therefore been claimed that a full account of this phenomenon requires a complete model of pragmatics that can handle intention recognition and formation. Indeed, Poesio and Rieser (to appear) claim that "the study of sentence completions ... may be used to compare competing claims about coordination – i.e. whether it is best explained with an intentional model like Clark (1996)'s ... or with a model based on simpler alignment models like Pickering and Garrod (2004)'s." They conclude that a model which includes modelling of intentions better captures the data.

For computational models of dialogue, however, SUs pose a challenge. While Poesio and Rieser (to appear) and Purver et al. (2006) provide general foundational models for various parts of the phenomenon, there are many questions that remain if we are to begin automatic processing. A computational dialogue system must be able to identify SUs, match up their two (or more) parts (which may not necessarily be adjacent), integrate them into some suitable syntactic and/or semantic representation, and determine the overall pragmatic contribution to the dialogue context. SUs also have implications for the organisation of *turn-taking* in such models (see e.g. Sacks et al. (1974)), as regards what conditions (if any) allow or prevent successful turn transfer. Additionally, from a socio-linguistic point of view, turn-taking operates (according to Schegloff (1995)) not on individual conversational participants, but on 'parties'. Lerner (1991) suggests that split utterances can clarify the formation of such parties in that they reveal evidence of how syntax can be employed to organise participants into 'groups'.

Analysis of SUs, when they can or cannot occur, and what effects they have on the coordination of agents in dialogue, is therefore an area of interest not only for conversational analysts wishing to characterise systematic interactions in dialogue, but also for linguists trying to formulate grammars of dialogue, psychologists and sociolinguists interested in alignment mechanisms and social interaction, and those interested in building automatic dialogue processing systems. In this paper we present and examine empirical corpus data in order to shed light on some of the questions and controversies around this phenomenon.

2 Related Work

Most previous work on what we call SUs has examined specific sub-cases, generally of the cross-speaker type, and have referred to these variously as *collaborative turn sequences* (Lerner, 1996; Lerner, 2004), *collaborative completions* (Clark, 1996; Poesio and Rieser, to appear), *co-constructions* (Sacks, 1992), *joint productions* (Helasvuo, 2004), *co-participant completions* (Hayashi, 1999; Lerner and Takagi, 1999), *collaborative productions* (Szczepek, 2000) and *anticipatory completions* (Fox and others, 2007) (amongst others). Here we discuss some of these views.

Conversation Analysis Lerner (1991) identifies various structures typical of SUs which contain characteristic split points. Firstly he gives a number of 'compound' *turn-constructional units* (TCUs), i.e., structures that include an initial constituent that hearers can identify as introducing some later final component. Examples include the IF X-THEN Y, WHEN X-THEN Y and INSTEAD

[1] Note that this says nothing about whether such a continuation is the same as the initial speaker's intended continuation.

OF X-Y constructions:

(1) A: Before that then if they were ill
G: They get nothing. *[BNC H5H 110-111]*

Other cues for potential *anticipatory completions* include quotation markers (e.g. SHE SAID), parenthetical inserts and lists, as well as non-syntactic cues such as contrast stress or prefaced disagreements. Rühlemann (2007) uses corpus analysis to examine *sentence relatives* as typical expansions of another interlocutor's turn (see also (16)):

(2) A: profit for the group is a hundred and ninety thousand pounds.
B: Which is superb. *[BNC FUK 2460-2461]*

Opportunistic Cases Although Lerner focuses on these projectable turn completions, he also mentions that splits can occur at other points such as "intra-turn silence", hesitations etc. which he terms *opportunistic completions*:

(3) A: Well I do know last week thet=uh Al was certainly very ⟨ pause 0.5⟩
B: pissed off *[(Lerner, 1996, p260)]*

As he makes no claims regarding the frequency of such devices for SUs, it would be interesting to know how common these are (insomuch as they occur at all and can be accordingly classified), especially as studies on SUs in Japanese (Hayashi, 1999) show that although SUs do occur, they do not rely on compound TCUs.

Expansions vs. Completions Other classifications of SUs often distinguish between *expansions* and *completions* (Ono and Thompson, 1993). Expansions are continuations which add, e.g., an adjunct, to an already complete syntactic element:

(4) T: It'll be an E sharp.
G: Which will of course just be played as an F. *[BNC G3V 262-263]*

whilst completions involve the addition of syntactic material which is required to make the whole utterance complete:

(5) A: . . . and then we looked along one deck, we were high up, and down below there were rows of, rows of lifeboats in case you see
B: There was an accident.
A: of an accident *[BNC HDK 63-65]*

In terms of frequency, the only estimate we know of is Szczepek (2000), where there are apparently 200 cross-person SUs in 40 hours of English conversation (there is no mention of the number of sentences or turns this equates to), of which 75% are completions.[2] As briefly outlined above, CA analyses of SUs tend to be broadly descriptive of what they reveal for conversational practices. Because such analyses present real examples they establish that the phenomenon is a genuine one; however, there is no discussion of its scale (with the exception of Szczepek (2000), which offers extremely limited figures). Even though as a genuine phenomenon it is of theoretical interest, the lack of frequency statistics prevents generalisability. Therefore, any claims that SUs are pervasive in dialogue need empirical backing.

Linguistic Models Purver et al. (2006) present a grammatical model for split utterances, using an inherently incremental grammar formalism, Dynamic Syntax (Kempson et al., 2001; Cann et al., 2005). This model shows how syntactic and semantic processing can be accounted for no matter where the split occurs in a sentence; however, as their interest is in grammatical processing, they give no account of any higher-level inferences which may be required. Poesio and Rieser (to appear) present a general model for *collaborative completions* based in the PTT framework, using an incremental LTAG-based grammar and an information-state-based approach to context modelling. While many parts of their model are compatible with a simple alignment-based communication model like Pickering and Garrod (2004)'s, they see intention recognition as crucial to dialogue management. They conclude that an intention-based model, more like Clark (1996)'s, is more suitable. Their primary concern is to show how such a model can account for the hearer's ability to infer a suitable continuation, but their use of an incremental interpretation method also allows an explanation of the low-level utterance processing required. Nevertheless, the use of an essentially head-driven grammar formalism suggests that some syntactic splits that appear in our corpus might be more problematic than others.

Corpus Studies Skuplik (1999), as reported by Poesio and Rieser (to appear), collected data from German two-party task-oriented dialogue, and annotated for split utterance phenomena. She found that *expansions* (cases where the part before the split can be considered already complete) were

[2] However, this could be affected by her decision not to include what she calls *appendor questions* in her data which could also be argued to be expansion SUs.

more common than *completions* (where the first part is incomplete as it stands). Given that this study focuses on task-oriented dialogue, it needs to be shown that its results can be replicated in naturally occurring dialogue. In addition, de Ruiter and van Dienst (in preparation) are also in the process of studying other-initiated completions, in the above sense, and their effect on the progressivity of dialogue turns; however no results are available to us at this point in time.

Dialogue Models We are not aware of any system/model which treats other-person splits, but same-person ones are now being looked at. Skantze and Schlangen (2009) present an incremental system design (for a limited domain) which can react to user feedback, e.g., backchannels, and resume with utterance completion if interrupted. Some related empirical work regarding the issue of turn-switch addressed here is also presented in Schlangen (2006) but the emphasis there centered mostly on prosodic rather than grammar/theory-based factors.

3 Method

3.1 Terminology

In this paper, as our interest is general, we use the term **split utterances** (*SUs*) to cover all instances where an utterance is spread across more than one dialogue contribution – whether the contributions are by the same or different speakers. We therefore use the term **split point** to refer to the point at which the utterance is split (rather than e.g. *transition point* which is associated with a speaker change). Cases where speaker does change across the split will be called **other-person** splits; otherwise **same-person** splits. One of the reasons for including same-person splits is that there are claims in the literature that the initial speaker may strategically continue completing their own utterance, after another person's intervention, as an alternative to acceptance or rejection of this intervention (*delayed completion*, (Lerner, 1996)). In addition, both grammatical formalisms (Purver et al., 2006) and psycholinguistic models (Pickering and Garrod, 2004) predict that SUs should be equally natural in both the same- and other- person conditions.

As not all cases will lead to complete contributions, and not all will be split over exactly two contributions, we also avoid terms like *first-half*, *second-half* and *completion*: instead the contributions on either side of a split point will be referred to as the **antecedent** and the **continuation**. In cases where an utterance has more than one split point, some portions may therefore act as the continuation for one split point, and the antecedent for the next.

3.2 Questions

General Our first interest is in the general statistics regarding SUs: how often do they occur, and what is the balance between same- and other-person splits? Do they usually fall into the specific categories (with specific preferred split points) examined by e.g. Lerner (1991), or can the split point be anywhere?

Completeness For a grammatical treatment of SUs, as well as for implementing parsing/production mechanisms for their processing, we need to know about the likely completeness of antecedent and continuation (if they are always complete in their own right, a standard head-driven grammar may be suitable; if not, something more fundamentally incremental may be required). In addition, CA and other strategic analyses of dialogue phenomena predict that split utterances should occur at turn-transfer points that are foreseeable by the participants. Complete syntactic units serve this purpose from this point of view and lack of such completeness will seem to weaken this general claim. We therefore ask how often antecedents and continuations are themselves complete,[3] and look at the syntactic and lexical categories which occur either side of the split.

Repair and Overlap Thirdly, we look at how often splits involve explicit repair of antecedent material, and how this depends on antecedent completeness. Although, sometimes, repair might be attributed to overlap or speaker uncertainty, it also might indicate issues regarding preemptive tactics on the part of the current speaker who needs to reformulate the original contribution in order to accommodate their novel offering or take into account feedback offered while constructing their utterance. Amount of repair also indicates the degree of attempt the current speaker is making to

[3] For antecedents, we are more interested in whether they *end* in a way that seems complete (they may have started irregularly due to overlap or another split); for continuations, whether they *start* in such a way (they may not get finished for some other reason, but we want to know if they would be complete if they do get finished).

Tag	Value	Explanation
`end-complete`	y/n	For all sentences: does this sentence end in such a way as to yield a complete proposition or speech act?
`continues`	sentence ID	For all sentences: does this sentence continue the proposition or speech act of a previous sentence? If so, which one?
`repairs`	number of words	For continuations: does this continuation explicitly repair words in the antecedent? If so, how many?
`start-complete`	y/n	For continuations: does this continuation start in such a way as to be able to stand alone as a complete proposition or speech act?

Table 1: Annotation Tags

integrate syntactically their contribution with the antecedent. However, we also examine how often continuations involve overlap, which also has implications for turn-taking management, and how this depends on antecedent completeness.

3.3 Corpus

For this exercise we used the portion of the BNC (Burnard, 2000) annotated by Fernández and Ginzburg (2002), chosen to maintain a balance between context-governed dialogue (tutorials, meetings, doctor's appointments etc.) and general conversation. This portion comprises 11,469 sentences taken from 200-turn sections of 53 separate dialogues.

The BNC transcripts are already annotated for overlapping speech, for non-verbal noises (laughter, coughing etc.) and for significant pauses. Punctuation is included, based on the original audio and the transcribers' judgements; as the audio is not available, we allowed annotators to use punctuation where it aided interpretation. The BNC transcription protocol provides a sentence-level annotation as well as an utterance (turn)-level one, where turns may be made of several sentences by the same speaker. We annotated at a sentence-level, to allow self-continuations within a turn to be examined. The BNC also forces turns to be presented in linear order, which is vital if we are to accurately assess whether turns are continuations of one another; however, this has a side-effect of forcing long turns to appear split into several shorter turns when interrupted by intervening backchannels. We will discuss this further below.

Annotation Scheme The initial stage of manual annotation involved 4 tags: `start-complete`, `end-complete`, `continues` and `repairs` – these are explained in Table 1 above. Sentences which somehow *require* continuation (whether they receive it or not) are therefore those marked `end-complete=n`; sentences which act as continuations are those marked with non-empty `continues` tags; and their antecedents are the values of those `continues` tags. Further specific information about the syntactic or lexical nature of antecedent or continuation components could then be extracted (semi-)automatically, using the BNC transcript and part-of-speech annotations.

Inter-Annotator Agreement Three annotators were used, all linguistically knowledgeable. First, all three annotators annotated one dialogue independently, then compared results and discussed differences. They then annotated 3 further dialogues independently to assess inter-annotator agreement; kappa statistics (Carletta, 1996) are shown in Table 2 below.

Tag	KND	KBG	KB0
`end-complete`	.86-.92	.80-1.0	.73-.90
`continues` (y/n)	.89-.81	.76-.85	.77-.89
`continues` (ant)	.90-.82	.74-.85	.76-.86
`repairs`	1.0-1.0	.55-.81	1.0-1.0

Table 2: Inter-Annotator κ statistic (min-max)

With the exception of the `repairs` tag for one annotator pair for one dialogue, all are above 0.7; the low figure results from a few disagreements in a dialogue with only a very small number of `repairs` instances. The remaining dialogues were divided evenly between the three annotators.

4 Results and Discussion

The 11,469 sentences annotated yielded 2,228 SUs, of which 1,902 were same-person and 326 other-person splits; 111 examples involved an explicit repair by the continuation of some part of the antecedent.

person:	same	other
overlapping	0	17
adjacent	840	260
sep. by overlap	320	10
sep. by backchnl	460	17
sep. by 1 sent	239	16
sep. by 2 sents	31	4
sep. by 3 sents	5	1
sep. by 4 sents	4	0
sep. by 5 sents	1	0
sep. by 6 sents	2	1
Total	1902	326

Table 3: Antecedent/continuation separation

person:	same	other
and/but/or	748	116
so/whereas	257	39
because	77	3
(pause)	56	5
which/who/etc	26	4
instead of	4	1
said/thought/etc	14	0
if_then	1	0
when_then	1	1
(other)	783	161

Table 4: Continuation categories

General Same-person splits are much more common than other-person; however, this is partly an artefact of the BNC transcription protocol (which forces contributions to be linearly ordered) and our choice to annotate at the sentence level. Around 44% of same-person cases are splits between sentences within the same-speaker turn; and a further 17% are separated only by other-speaker material which entirely overlaps with the antecedent and therefore does not necessarily actually interrupt the turn. Both of these might be considered as single utterances under some views. However, we believe that splits between same-turn sentences must be investigated in that the transcription into separate sentences does indicate some pause or other separating prosody and, from a processing/psycholinguistic point of view, it should be determined whether other-person splits occur in the same places as same-person split boundaries. Even in cases of overlap, one cannot exclude the fact that the shape of the current speaker's utterance is influenced by receipt of the feedback. Nevertheless, we will examine these issues in further research and hence we exclude within-turn splits of this type from here on.

Many splits are non-adjacent (see Table 3), with the antecedent and continuation separated by at least one intervening sentence. In same-person cases, once we have excluded the within-turn splits described above, this must in fact always be the case; the intervening material is usually a backchannel (62% of remaning cases) or a single other sentence (32%, often e.g. a clarification question), but two intervening sentences are possible (4%) with up to six being seen. In other-person cases, 88% are adjacent or separated only by overlapping material, but again up to six intervening sentences were seen, with a single sentence most common (10%, in half of which the intervening sentence was a backchannel).

Many utterances have more than one split. In same-person cases, a single utterance can be split over as many as thirteen individual sentence contributions; although such extreme cases occur generally within one-sided dialogues such as tutorials, many multi-split cases are also seen in general conversation. Only 63% of cases consisted of only two contributions. Antecedents can also receive more than one competing continuation, although this is rare: two continuations are seen in 2% of cases.

CA Categories We searched for examples which match CA categories (Lerner, 1991; Rühlemann, 2007) by looking for particular lexical items on either side of the split. Matching was done loosely, to allow for the ungrammatical nature of dialogue – for example, an instance was taken to match the IF X-THEN Y pattern if the continuation began with *'then'* (modulo filled pauses and non-verbal material) and the antecedent contained *'if'* at any point) – so the counts may be over-estimates. For Lerner (1996)'s *opportunistic* cases, we looked for filled pauses (*'er/erm'* etc.) or pauses explicitly annotated in the transcript, so counts in this case may be underestimates.[4] We also chose some other broad categories based on our observations of the most common cases. Results are shown in Table 4.[5]

The most common of the CA categories can be

[4] In further research we will examine other features as specialised laugh tokens, repetitions etc. as well as their particular positioning

[5] Note that the categories in Table 4 are not all mutually exclusive (e.g. an example may have both an *'and'*-initial continuation and an antecedent ending in a pause), so column sums will not match Table 3.

seen to be Lerner (1996)'s hesitation-related *opportunistic* cases, which make up at least 2-3% of both same- and other-person splits. Rühlemann (2007)'s *sentence relative* clause cases are next, with over 1%; the others make up only small proportions.

In contrast, by far the most common pattern (for both same- and other-) is the addition of an extending clause, either a conjunction introduced by *'and/but/or/nor'* (35-40%), or other clause types with *'so/whereas/nevertheless/because'*. Other less obviously categorisable cases make up 40-50% of continuations, with the most common first words being *'you'*, *'it'*, *'I'*, *'the'*, *'in'* and *'that'*.

Completeness and repair Examination of the `end-complete` annotations shows that about 8% of sentences in general are incomplete, but that (perhaps surprisingly) only 63% of these get continued. For both same- and other-person continuations, the vast majority (72% and 74%) continue an already complete antecedent, with only 26-28% therefore being *completions* in the sense of e.g. de Ruiter and van Dienst (in preparation). This does, however, mean that continuations are significantly more likely than other sentences to follow an incomplete antecedent ($p < 0.001$ using $\chi^2_{(1)}$). Interestingly, though, continuations are no more likely than other sentences to be complete themselves.

The frequent clausal categories from Table 4 are all more likely to continue complete antecedents than incomplete ones, with the exception of the `(other)` category; this suggests that split points often occur at random points in a sentence, without regard to particular clausal constructions (see also A.1 for more examples and context):

(6) D: you know what the actual variations
U: entails
D: entails. you know what the actual quality of the variations are.
[BNC G4V 114-117]

For the less frequent (e.g. *'if/then'*, *'instead of'*) categories, the counts are too low to be sure.

Excluding all the clausal constructions (i.e. looking only at the general `(other)` category), and looking only at other-person cases, we see that antecedents often end in a complete way (53%) but that continuations do not often start in a complete way (24%). Continuations are more than twice as likely to start in a non-complete as opposed to complete way, even after complete antecedents. Explicit repair of some portion of the antecedent is not common, only occurring in just under 5% of splits. As might be expected, incomplete antecedents are more likely to be repaired (13% vs. 2%, $p < 0.001$ using $\chi^2_{(1)}$). Other-continuations are also significantly more likely to repair their antecedents than same-person cases (10% vs. 4%, $p < 0.001$ using $\chi^2_{(1)}$).

Problematic cases Examination of the data shows that SUs is not necessarily an autonomous well-defined category independent of other fragment classifications in the literature. Besides cases where it is not easy to identify whether a fragment is a continuation or not or what the antecedent is (see A.2), there are also cases where, as has already been pointed out in the literature (Gregoromichelaki et al., 2009; Bunt, 2009), fragments exhibit multifunctionality. This can be illustrated by the following where the continuation could be taken also as request for confirmation/question (7) or a reply to a clarification request (8):

(7) M: It's generated with a handle and
J: Wound round?
M: Yes *[BNC K69 109-112]*

(8) S: Quite a good word processor.
J: A word processor?
S: Which is vag- it's basically a subset of Word. *[BNC H61 37-39]*

In this respect, an interesting category is Lerner's *delayed completion*s where often the continuation also serves as some kind of repair or reformulation (see e.g. (6) and A.3 (26)).

5 Conclusions

Although most of Lerner (1991)'s categories appear, they are not necessarily the most frequent. On the other hand, the general results seem to indicate that splits can occur anywhere in a string, both in the same- or other- conditions. Both these are consistent with models that advocate highly coordinated resources between interlocutors and, moreover, the need for highly incremental means of processing (Purver et al., 2006; Skantze and Schlangen, 2009). From a computational modelling point of view, the results also indicate that start-completeness of continuations is rare, which means that a dialogue system has a chance of spotting continuations from surface characteristics of

the input. This is hampered though by the fact that the split can occur within any type of syntactic constituent, hence no reliable grammatical features can be employed securely. On the other hand, end-incompleteness of antecedents is not as common as would be expected and long distances between antecedent and continuation are possible. In this respect, locating the antecedent is not a straightforward task for automated systems, especially again as this can be any type of constituent.

References

H. Bunt. 2009. Multifunctionality and multidimensional dialogue semantics. In *Proceedings of DiaHolmia, 13th SEMDIAL Workshop*.

L. Burnard. 2000. *Reference Guide for the British National Corpus (World Edition)*. Oxford University Computing Services `http://www.natcorp.ox.ac.uk/docs/userManual/`.

R. Cann, R. Kempson, and L. Marten. 2005. *The Dynamics of Language*. Elsevier, Oxford.

J. Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–255.

H. Clark. 1996. *Using Language*. Cambridge University Press.

J. de Ruiter and M. van Dienst. in preparation. Completing other people's utterances: evidence for forward modeling in conversation. ms.

R. Fernández and J. Ginzburg. 2002. Non-sentential utterances: A corpus-based study. *Traitement Automatique des Langues*, 43(2).

A. Fox et al. 2007. Principles shaping grammatical practices: an exploration. *Discourse Studies*, 9(3):299.

E. Gregoromichelaki, Y. Sato, R. Kempson, A. Gargett, and C. Howes. 2009. Dialogue modelling and the remit of core grammar. In *Proceedings of IWCS*.

M. Hayashi. 1999. Where Grammar and Interaction Meet: A Study of Co-Participant Completion in Japanese Conversation. *Human Studies*, 22(2):475–499.

M. Helasvuo. 2004. Shared syntax: the grammar of co-constructions. *Journal of Pragmatics*, 36(8):1315–1336.

R. Kempson, W. Meyer-Viol, and D. Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

G. Lerner and T. Takagi. 1999. On the place of linguistic resources in the organization of talk-in-interaction: A co-investigation of English and Japanese grammatical practices. *Journal of Pragmatics*, 31(1):49–75.

G. Lerner. 1991. On the syntax of sentences-in-progress. *Language in Society*, pages 441–458.

G. Lerner. 1996. On the semi-permeable character of grammatical units in conversation: Conditional entry into the turn space of another speaker. In E. Ochs, E. A. Schegloff, and S. A. Thompson, editors, *Interaction and grammar*, pages 238–276. Cambridge University Press.

G. Lerner. 2004. Collaborative turn sequences. In *Conversation analysis: Studies from the first generation*, pages 225–256. John Benjamins.

T. Ono and S. Thompson. 1993. What can conversation tell us about syntax. In P. Davis, editor, *Alternative Linguistics: Descriptive and Theoretical Modes*. Benjamin.

M. Pickering and S. Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226.

M. Poesio and H. Rieser. to appear. Completions, coordination, and alignment in dialogue. Ms.

M. Purver, R. Cann, and R. Kempson. 2006. Grammars as parsers: Meeting the dialogue challenge. *Research on Language and Computation*, 4(2-3):289–326.

C. Rühlemann. 2007. *Conversation in context: a corpus-driven approach*. Continuum.

H. Sacks, E. A. Schegloff, and G. Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.

H. Sacks. 1992. *Lectures on Conversation*. Blackwell.

E. Schegloff. 1995. Parties and talking together: Two ways in which numbers are significant for talk-in-interaction. *Situated order: Studies in the social organization of talk and embodied activities*, pages 31–42.

D. Schlangen. 2006. From reaction to prediction: Experiments with computational models of turn-taking. In *Proceedings of the 9th International Conference on Spoken Language Processing (INTERSPEECH - ICSLP)*.

G. Skantze and D. Schlangen. 2009. Incremental dialogue processing in a micro-domain. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*.

K. Skuplik. 1999. Satzkooperationen. definition und empirische untersuchung. SFB 360 1999/03, Bielefeld University.

B. Szczepek. 2000. Formal Aspects of Collaborative Productions in English Conversation. *Interaction and Linguistic Structures (InLiSt)*, `http://www.uni-potsdam.de/u/inlist/issues/17/`.

A Examples

A.1 Split points

(6) D: Yeah I mean if you're looking at quantitative things it's really you know how much actual- How much variation happens whereas qualitative is ⟨pause⟩ you know what the actual variations
U: entails
D: entails. you know what the actual quality of the variations are.
[BNC G4V 114-117]

(9) A: All the machinery was
G: [[All steam.]][6]
A: [[operated]] by steam
[BNC H5G 177-179]

(10) K: I've got a scribble behind it, oh annual report I'd get that from.
S: Right.
K: And the total number of [[sixth form students in a division.]]
S: [[Sixth form students in a division.]] Right.
[BNC H5D 123-127]

(11) M: 292 And another sixteen percent is the other Ne- Nestle coffee ⟨pause⟩ erm Blend Thirty Seven which I used to drink a long time ago and others ⟨laugh⟩ and twenty two percent is er ⟨pause⟩
U: Maxwell.
M: Maxwell House, which has become the other local brand now seeing as how Maxwell House is owned by Kraft, and Kraft now own Terry's.
[BNC G3U 292-294]

(12) A: Erm because as Moira said that Kraft is erm ⟨pause⟩ now what was she saying, what was she saying Kraft is the same as ⟨pause⟩
M: Craft? *[BNC G3U 412-413]*

(13) J: And I couldn't remember whether she said at the end of the three months or
A: End of the month. *[BNC H4P 17-18]*

(14) G: Had their own men
A: unload the boats?
G: unload the boats, yes. *[BNC H5H 91-93]*

(15) G: That's right they had to go on a rota.
A: Run by the Dock Commission?
G: Run by the Dock Commission.
[BNC H5H 100-102]

(16) A: So I thought, oh, I think I'll put lace over it, it'll tone the lilac [[down.]]
B: [[down.]] Yes.
Which it is has done
[BNC KBC 3195-3198]

A.2 Uncertain antecedents

(17) C: Look you're cleaning this ⟨pause⟩ [[with erm]]
G: [[That box.]]
C: [[This.]]
G: [[With]] this. [[And this.]]
C: [[And this.]] [[And this.]]
G: [[And this.]]
Whoops! *[BNC KSR 9-17]*

(18) S: You're trying to be everything ⟨pause⟩ and they're pushing it away cos it's not what they really want ⟨pause⟩ and they, I mean, all, all you can get from him is how marvellous, you're right, how marvellous his brothers are ⟨pause⟩ and yet, what I've heard of the brothers they're not
C: Not much, [[yeah.]]
S: [[they're]] not all that marvellous, they're not really that much to look [[up]]
C: [[Ah]].
S: to.
C: No *[BNC KBG 76-81]*

(19) S: Well this is why I think he'd be better off, hi- his needs ⟨pause⟩ are not met by a class teacher. And I don't think they have been for this last
C: Mm, we need a support teacher [[to go there.]]
S: [[for the last]] year. But yo-, you need somebody who's gonna work with him every day ⟨pause⟩ and ⟨pause⟩ with an individual programme and you just can't offer that ⟨pause⟩ in a class. *[BNC KBG 56-60]*

[6] Overlapping material is shown in double square brackets, aligned with the material with which it co-occurs.

(20) M: I might be a bit biased, I think they still do that but I think erm ⟨pause⟩
J: The television has ⟨pause⟩
M: the television has made a difference. I think not only just at fire stations, I think in the whole of life, hasn't it?
[BNC K69 51-54]

(21) A5: I'll definitely use that
U: ⟨reading⟩:[Get a headache]?
A5: [[in getting to know]]
A2: [[Year seven]]
A5: new [[year seven]]
A2: [[Oh yeah]] for year seven
[BNC J8D 190-195]

(22) G: Well a chain locker is where all the spare chain used to like coil up
A: So it ⟨unclear⟩ came in and it went round
G: round the barrel about three times round the barrel then right down into the chain locker but if you kept, let it ride what we used to call let it ride well ⟨unclear⟩ well now it get so big then you have to run it all off cos you had one lever, that's what you had and the steam valve could have all steamed. *[BNC H5G 174:176]*

A.3 Multifunctionality of fragments

(7) *Completion and confirmation request:*
J: How does it generate?
M: It's generated with a handle and
J: Wound round?
M: Yes, wind them round and this should, should generate a charge which rang bells and sounded bells and then er you lift up a telephone and plug in a jack and, and take a message in that way.
[BNC K69 109-112]

(23) *Completion and confirmation request:*
G: Had their own men
A: unload the boats?
G: unload the boats, yes. *[BNC H5H 91-93]*

(24) *Late completion and (repetitive) confirmation:*
N: Alistair [last or full name] erm he's, he's made himself er he has made himself co-ordinator.
U: And section engineer.
N: And section engineer.
N: I didn't sign it as coordinator.
[BNC H48 141-144]

(25) *Completion and clarification reply:*
John: If you press N
Sarah: N?
John: N for name, it'll let you type in the docu document name. *[BNC G4K 84-86]*

(26) *Expansion and reformulation/repair:*
S: Secondly er
J: We guarantee P five.
S: We we are we're guaranteeing P five plus a noise level.
J: Yeah. *[BNC JP3 167-170]*

(27) *Expansion and question:*
I: I can't remember exactly who lived on the right hand side, I've forgotten but th I know the Chief Clerk lived just a little way down [address], you see, er
A: In one of those little red brick cottages?
[BNC HDK 124-125]

(28) *Answer and expansion:*
A: We could hear it from outside ⟨unclear⟩.
R: Oh you could hear it?
A: Occasionally yeah. *[BNC J8D 13-15]*

(29) *Answer/reformulation and expansion:*
G: [address], that was in the middle, more or less in the middle of the town.
A: And you called that the manual?
G: The manual school, yes.
[BNC H5G 96-98]

k-Nearest Neighbor Monte-Carlo Control Algorithm for POMDP-based Dialogue Systems

F. Lefèvre*, M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu and S. Young
Spoken Dialogue Systems Group
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, UK
{frfl2, mg436, fj228, sk561, farm2, brmt2, ky219, sjy}@eng.cam.ac.uk

Abstract

In real-world applications, modelling dialogue as a POMDP requires the use of a summary space for the dialogue state representation to ensure tractability. Sub-optimal estimation of the value function governing the selection of system responses can then be obtained using a grid-based approach on the belief space. In this work, the Monte-Carlo control technique is extended so as to reduce training over-fitting and to improve robustness to semantic noise in the user input. This technique uses a database of belief vector prototypes to choose the optimal system action. A locally weighted k-nearest neighbor scheme is introduced to smooth the decision process by interpolating the value function, resulting in higher user simulation performance.

1 Introduction

In the last decade dialogue modelling as a Partially Observable Markov Decision Process (POMDP) has been proposed as a convenient way to improve spoken dialogue systems (SDS) trainability, naturalness and robustness to input errors (Young et al., 2009). The POMDP framework models dialogue flow as a sequence of unobserved dialogue states following stochastic moves, and provides a principled way to model uncertainty.

However, to deal with uncertainty, POMDPs maintain distributions over all possible states. But then training an optimal policy is an NP hard problem and thus not tractable for any non-trivial application. In recent works this issue is addressed by mapping the dialog state representation space (the *master* space) into a smaller *summary* space (Williams and Young, 2007). Even though optimal policies remain out of reach, sub-optimal solutions can be found by means of grid-based algorithms.

Within the Hidden Information State (HIS) framework (Young et al., 2009), policies are represented by a set of grid points in the summary belief space. Beliefs in master space are first mapped into summary space and then mapped into a summary action via the dialogue policy. The resulting summary action is then mapped back into master space and output to the user.

Methods which support interpolation between points are generally required to scale well to large state spaces (Pineau et al., 2003). In the current version of the HIS framework, the policy chooses the system action by associating each new belief point with the single, closest, grid point. In the present work, a k-nearest neighbour extension is evaluated in which the policy decision is based on a locally weighted regression over a subset of representative grid points. This method thus lies between a strictly grid-based and a point-based value iteration approach as it interpolates the value function around the queried belief point. It thus reduces the policy's dependency on the belief grid point selection and increases robustness to input noise.

The next section gives an overview of the CUED HIS POMDP dialogue system which we extended for our experiments. In Section 3, the grid-based approach to policy optimisation is introduced followed by a presentation of the k-nn Monte-Carlo policy optimization in Section 4, along with an evaluation on a simulated user.

*Fabrice Lefèvre is currently on leave from the University of Avignon, France.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 272–275,
Queen Mary University of London, September 2009.

2 The CUED Spoken Dialogue System

2.1 System Architecture

The CUED HIS-based dialogue system pipelines five modules: the ATK speech recogniser, an SVM-based semantic tuple classifier, a POMDP dialogue manager, a natural language generator, and an HMM-based speech synthesiser. During an interaction with the system, the user's speech is first decoded by the recogniser and an N-best list of hypotheses is sent to the semantic classifier. In turn the semantic classifier outputs an N-best list of user dialogue acts. A dialogue act is a semantic representation of the user action headed by the user intention (such as `inform`, `request`, etc) followed by a list of items (slot-value pairs such as `type=hotel, area=east` etc). The N-best list of dialogue acts is used by the dialogue manager to update the dialogue state. Based on the state hypotheses and the policy, a machine action is determined, again in the form of a dialogue act. The natural language generator translates the machine action into a sentence, finally converted into speech by the HMM synthesiser. The dialogue system is currently developed for a tourist information domain (Towninfo). It is worth noting that the dialogue manager does not contain any domain-specific knowledge.

2.2 HIS Dialogue Manager

The unobserved dialogue state of the HIS dialogue manager consists of the user goal, the dialogue history and the user action. The user goal is represented by a partition which is a tree structure built according to the domain ontology. The nodes in the partition consist mainly of slots and values. When querying the venue database using the partition, a set of matching entities can be produced. The dialogue history consists of the grounding states of the nodes in the partition, generated using a finite state automaton and the previous user and system action. A hypothesis in the HIS approach is then a triple combining a partition, a user action and the respective set of grounding states. The distribution over all hypotheses is maintained throughout the dialogue (*belief state monitoring*). Considering the ontology size for any real-world problem, the so-defined state space is too large for any POMDP learning algorithm. Hence to obtain a tractable policy, the state/action space needs to be reduced to a smaller scale summary space. The set of possible machine dialogue acts is also reduced in summary space. This is mainly achieved by removing all act items and leaving only a reduced set of dialogue act types. When mapping back into master space, the necessary items (i.e. slot-value pairs) are inferred by inspecting the most likely dialogue state hypotheses.

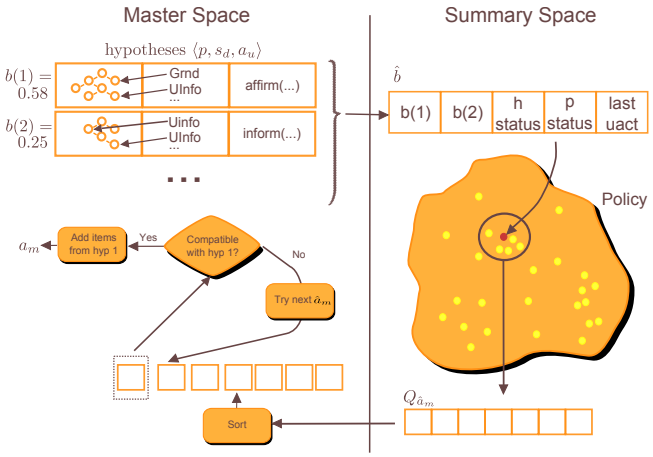


Figure 1: Master-summary Space Mapping.

The optimal policy is obtained using reinforcement learning in interaction with an agenda based simulated user (Schatzmann et al., 2007). At the end of each dialogue a reward is given to the system: +20 for a successful completion and -1 for each turn. A grid-based optimisation is used to obtain the optimal policy (see next section). At each turn the belief is mapped to a summary point from which a summary action can be determined. The summary action is then mapped back to a master action by adding the relevant information.

3 Grid-based Policy Optimisation

In a POMDP, the optimal exact value function can be found iteratively from the terminal state in a process called *value iteration*. At each iteration t, policy vectors are generated for all possible action/observation pairs and their corresponding values are computed in terms of the policy vectors at step $t-1$. However, exact optimisation is not tractable in practice, but approximate solutions can still provide useful policies. Representing a POMDP policy by a grid of representative belief points yields an MDP optimisation problem for which many tractable solutions exist, such as the Monte Carlo Control algorithm (Sutton and Barto, 1998) used here.

In the current HIS system, each summary belief point is a vector consisting of the probabilities of the top two hypotheses in master space, two discrete status variables summarising the state of the

Algorithm 1 Policy training with k-nn Monte Carlo

Let $Q(\hat{b}, \hat{a}_m)$ = expected reward on taking action $\hat{a}_m$ from belief point $\hat{b}$
Let $N(\hat{b}, \hat{a}_m)$ = number of times action $\hat{a}_m$ is taken from belief point $\hat{b}$
Let $\mathcal{B}$ be a set of grid-points in belief space, $\{\hat{b}\}$ any subset of it
Let $\pi_{\text{knn}} : \hat{b} \rightarrow \hat{a}_m; \forall \hat{b} \in \mathcal{B}$ be a policy
repeat
  $t \leftarrow 0$
  $\hat{a}_{m,0} \leftarrow$ initial greet action
  $b = b_0$ [= all states in single partition]
  Generate dialogue using ϵ-greedy policy
  repeat
    $t \leftarrow t + 1$
    Get user turn $a_{u,t}$ and update belief state b
    $\hat{b}_t \leftarrow \text{SummaryState}(b)$
    $\{\hat{b_k}\}_{\text{knn}} \leftarrow k\text{-Nearest}(\hat{b}_t, \mathcal{B})$
    $\hat{a}_{m,t} \leftarrow \begin{cases} \text{RandomAction} & \text{with probability } \epsilon \\ \pi_{\text{knn}}(\hat{b}_t) & \text{otherwise} \end{cases}$
    record $\langle \hat{b}_t, \{\hat{b_k}\}_{\text{knn}}, \hat{a}_{m,t} \rangle, T \leftarrow t$
  until dialogue terminates with reward R from user simulator
  Scan dialogue and update $\mathcal{B}$, Q and N
  for $t = T$ **downto** 1 **do**
    if $\exists \hat{b}_i \in \mathcal{B}$, $|\hat{b}_t - \hat{b}_i| < \delta$ **then** ← update nearest pt in $\mathcal{B}$
      for all $\hat{b}_k$ **in** $\{\hat{b}_k\}_{\text{knn}}$ **do**
        $w \leftarrow \Phi(\hat{b}_t, \hat{b}_k)$ ← Φ weighting function
        $Q(\hat{b}_k, \hat{a}_{m,t}) \leftarrow \frac{Q(\hat{b}_k, \hat{a}_{m,t}) * N(\hat{b}_k, \hat{a}_{m,t}) + R * w}{N(\hat{b}_k, \hat{a}_{m,t}) + w}$
        $N(\hat{b}_k, \hat{a}_{m,t}) \leftarrow N(\hat{b}_k, \hat{a}_{m,t}) + w$
      end for
    else ← create new grid point
      add $\hat{b}_t$ to $\mathcal{B}$
      $Q(\hat{b}_t, \hat{a}_{m,t}) \leftarrow R, N(\hat{b}_t, \hat{a}_{m,t}) \leftarrow 1$
    end if
    $R \leftarrow \gamma R$ ← discount the reward
  end for
until converged

top hypothesis and its associated partition, and the type of the last user act.

In order to use such a policy, a simple distance metric in belief space is used to find the closest grid point to a given arbitrary belief state:

$$|\hat{b}_i - \hat{b}_j| = \sum_{d=1}^{2} \alpha_d \cdot \sqrt{(\hat{b}_i(d) - \hat{b}_j(d))^2} + \sum_{d=3}^{5} \alpha_d \cdot (1 - \delta(\hat{b}_i(d), \hat{b}_j(d))) \quad (1)$$

where the α's are weights, d ranges over the 2 continuous and 3 discrete components of $\hat{b}$ and $\delta(x, y)$ is 1 iff $x = y$ and 0 otherwise.

Associated with each belief point is a function $Q(\hat{b}, \hat{a}_m)$ which records the expected reward of taking summary action $\hat{a}_m$ when in belief state $\hat{b}$. Q is estimated by repeatedly executing dialogues and recording the sequence of belief point-action pairs $\langle \hat{b}_t, \hat{a}_{m,t} \rangle$. At the end of each dialogue, each $Q(\hat{b}_t, \hat{a}_{m,t})$ estimate is updated with the actual discounted reward. Dialogues are conducted using the current policy π but to allow exploration of unvisited regions of the state-action space, a random action is selected with probability ϵ.

Once the Q values have been estimated, the policy is found by setting

$$\pi(\hat{b}) = \arg\max_{\hat{a}_m} Q(\hat{b}, \hat{a}_m), \qquad \forall \hat{b} \in \mathcal{B} \quad (2)$$

Belief points are generated on demand during the policy optimisation process. Starting from a single belief point, every time a belief point is encountered which is sufficiently far from any existing point in the policy grid, it is added to the grid as a new point. The inventory of grid points is thus growing over time until a predefined maximum number of stored belief vectors is reached.

The training schedule adopted in this work is comparable to the one presented in (Young et al., 2009). Training starts in a noise free environment using a small number of grid points and it continues until the performance of the policy asymptotes. The resulting policy is then taken as an initial policy for the next stage in which the noise level is increased, the set of grid points is expanded and the number of iterations is increased. In practice a total of 750 to 1000 grid points have been found to be sufficient and the total number of simulated dialogues needed for training is around 100,000.

4 k-nn Monte-Carlo Policy Optimization

In this work, we use the k nearest neighbor method to obtain a better estimate of the value function, represented by the belief points' Q values. The algorithm maintains a set of sample vectors $\hat{b}$ along with their Q value vector $Q(\hat{b}, a)$. When a new belief state $\hat{b}'$ is encountered, its Q values are obtained by looking up its k-nearest neighbours in the database, then averaging their Q-values.

To obtain good estimates for the value function interpolation, local weights are used based on the belief point distance. A Kullback-Leibler (KL) divergence (relative entropy) could be used as a distance function between the belief points. However, while the KL-divergence between two continuous distributions is well defined, this is not the case for sample sets. In accordance with the locally weighted learning theory (Atkeson et al., 1997), a simple weighting scheme based on a nearly Euclidean distance (eq. 1) is used to interpolate the policy over a set of points:

$$\pi_{\text{knn}}(\hat{b}) = \arg\max_{\hat{a}_m} \sum_{\{\hat{b}_k\}_{\text{knn}}} Q(\hat{b}_k, \hat{a}_m) \times \Phi(\hat{b}_k, \hat{b})$$

In our experiments, we set the weighting coefficients with the kernel function $\Phi(\hat{b}_1, \hat{b}_2) = e^{-|\hat{b}_1 - \hat{b}_2|^2}$.

Since it can be impossible to construct a full system act from the best summary act, a back-off strategy is used: an N-best list of summary acts, ranked by their Q values, is scrolled through until a feasible summary act is found. The resulting overall process of mapping between master and summary space and back is illustrated in Figure 1. The complete k-nn version policy optimisation algorithm is described in Algorithm 1.

The user simulator results for semantic error rates ranging from 0 to 50% with a 5% step are shown in Figure 2 for $k \in \{1, 3, 5, 7\}$, averaged over 3000 dialogues. The results demonstrate that the k-nn policies outperform the baseline 1-nn policy, especially on high noise levels. While our initial expectations are met, increasing k above 3 does not improve performances. This is likely to be due to the small size of the summary space as well as the use of discrete dimensions. However enlarging the summary space and the sample set is conceivable with k-nn time-efficient optimisations (as in (Lefèvre, 2003)).

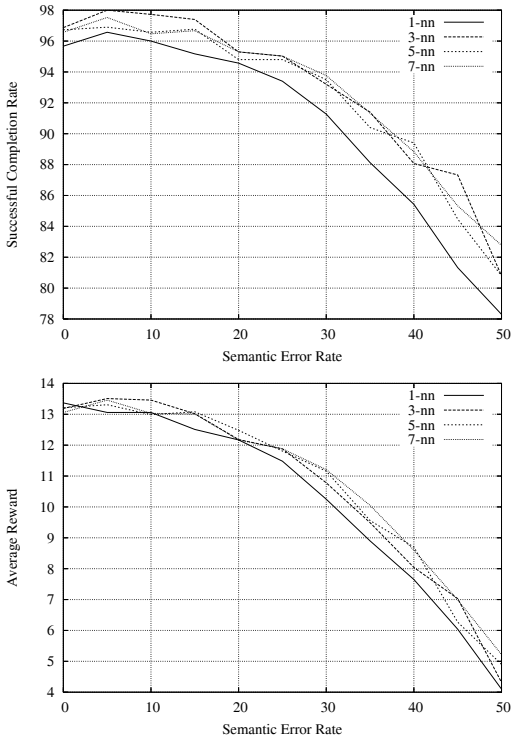


Figure 2: Comparison of the percentage of successful simulated dialogues and the average reward between the k-nn strategies on different error rates.

5 Conclusion

In this paper, an extension to a grid-based policy optimisation technique has been presented and evaluated within the CUED HIS-based dialogue system. The Monte-Carlo control policy optimisation algorithm is complemented with a k-nearest neighbour technique to ensure a better generalization of the trained policy along with an increased robustness to noise in the user input. Preliminary results from an evaluation with a simulated user confirm that the k-nn policies outperform the 1-nn baseline on high noise, both in terms of successful dialogue completion and accumulated reward.

Acknowledgements

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).

References

C Atkeson, A Moore, and S Schaal. 1997. Locally weighted learning. *AI Review*, 11:11–73, April.

F Lefèvre. 2003. Non-parametric probability estimation for HMM-based automatic speech recognition. *Computer Speech & Language*, 17(2-3):113 – 136.

J Pineau, G Gordon, and S Thrun. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *Proc IJCAI*, pages pp1025–1032, Mexico.

J Schatzmann, B Thomson, K Weilhammer, H Ye, and SJ Young. 2007. Agenda-Based User Simulation for Bootstrapping a POMDP Dialogue System. In *HLT/NAACL*, Rochester, NY.

RS Sutton and AG Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, Mass.

JD Williams and SJ Young. 2007. Scaling POMDPs for Spoken Dialog Management. *IEEE Audio, Speech and Language Processing*, 15(7):2116–2129.

SJ Young, M Gašić, S Keizer, F Mairesse, J Schatzmann, B Thomson, and K Yu. 2009. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, In Press, Uncorrected Proof.

Comparison of Classification and Ranking Approaches to Pronominal Anaphora Resolution in Czech*

Nguỵ Giang Linh, Václav Novák, Zdeněk Žabokrtský
Charles University in Prague
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, CZ-11800
{linh,novak,zabokrtsky}.ufal.mff.cuni.cz

Abstract

In this paper we compare two Machine Learning approaches to the task of pronominal anaphora resolution: a conventional classification system based on C5.0 decision trees, and a novel perceptron-based ranker. We use coreference links annotated in the Prague Dependency Treebank 2.0 for training and evaluation purposes. The perceptron system achieves f-score 79.43% on recognizing coreference of personal and possessive pronouns, which clearly outperforms the classifier and which is the best result reported on this data set so far.

1 Introduction

Anaphora Resolution (AR) is a well established task in Natural Language Processing (Mitkov, 2002). Classification techniques (e.g., single candidate model aimed at answering: "Is there a coreference link between the anaphor and this antecedent candidate, or not?") are very often used for the task, e.g. in Mccarthy and Lehnert (1995) and Soon et al. (2001). However, as argued already in Yang et al. (2003), better results are achieved when the candidates can compete in a pairwise fashion. It can be explained by the fact that in this approach (called twin-candidate model), more information is available for the decision making. If we proceed further along this direction, we come to the ranking approach described in Denis and Baldridge (2007), in which the entire candidate set is considered at once and which leads to further significant shift in performance, more recently documented in Denis and Baldridge (2008).

In this paper we deal with supervised approaches to pronominal anaphora in Czech.[1] For training and evaluation purposes, we use coreferences links annotated in the Prague Dependency Treebank, (Jan Hajič, et al., 2006). We limit ourselves only to textual coreference (see Section 2) and to personal and possessive pronouns. We make use of a rich set of features available thanks to the complex annotation scenario of the treebank.

We experiment with two of the above mentioned techniques for AR: a classifier and a ranker. The former is based on a top-down induction of decision trees (Quinlan, 1993). The latter uses a simple scoring function whose optimal weight vector is estimated using perceptron learning inspired by Collins (2002). We try to provide both implementations with as similar input information as possible in order to be able to compare their performance for the given task.

Performance of the presented systems can be compared with several already published works, namely with a rule-based system described in Kučová and Žabokrtský (2005), some of the "classical" algorithms implemented in Němčík (2006), a system based on decision trees (Nguỵ, 2006), and a rule-based system evaluated in Nguỵ and Žabokrtský (2007). To illustrate the real complexity of the task, we also provide performance evaluation of a baseline solution.

The work on this project was supported by the grants MSM 0021620838, GAAV ČR 1ET101120503 and 1ET201120505, MŠMT ČR LC536, and GAUK 4383/2009

[1] Currently one can see a growing interest in unsupervised techniques, e.g. Charniak and Elsner (2009) and Ng (2008). However, we make only a very tiny step in this direction: we use a probabilistic feature based on collocation counts in large unannotated data (namely in the Czech National Corpus).

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 276–285,
Queen Mary University of London, September 2009.

The most important result claimed in this paper is that, to the best of our knowledge, the presented ranker system outperforms all the previously published systems evaluated on the PDT data. Moreover, the performance of our ranker (f-score 79.43%) for Czech data is not far from the performance of the state-of-the-art system for English described in Denis and Baldridge (2008) (f-score for 3rd person pronouns 82.2 %).[2]

A side product of this work lies in bringing empirical evidence – for a different language and different data set – for the claim of Denis and Baldridge (2007) that the ranking approach is more appropriate for the task of AR than the classification approach.

The paper is structured as follows. The data with manually annotated links we use are described in Section 2. Section 3 outlines preprocessing the data for training and evaluation purposes. The classifier-based and ranker-based systems are described in Section 4 and Section 5 respectively. Section 6 summarizes the achieved results by evaluating both approaches using the test data. Conclusions and final remarks follow in Section 7.

2 Coreference links in the Prague Dependency Treebank 2.0

The Prague Dependency Treebank 2.0[3] (PDT 2.0, Jan Hajič, et al. (2006)) is a large collection of linguistically annotated data and documentation, based on the theoretical framework of Functional Generative Description (FGD; introduced by Sgall (1967) and later elaborated, e.g. in by Sgall et al. (1986)). The PDT 2.0 data are Czech newspaper texts selected from the Czech National Corpus[4] (CNC).

The PDT 2.0 has a three-level structure. On the lowest *morphological* level, a lemma and a positional morphological tag are added to each token. The middle *analytical* level represents each sentence as a surface-syntactic dependency tree. On the highest *tectogrammatical* level, each sentence is represented as a complex deep-syntactic dependency tree, see Mikulová and others (2005) for details. This level includes also annotation of coreferential links.

The PDT 2.0 contains 3,168 newspaper texts (49,431 sentences) annotated on the tectogrammatical level. Coreference has been annotated manually in all this data. Following the FGD, there are two types of coreference distinguished: *grammatical* coreference and *textual* coreference (Panevová, 1991). The main difference between the two coreference types is that the antecedent in grammatical coreference can be identified using grammatical rules and sentence syntactic structure, whereas the antecedent in textual coreference can not.

The further division of grammatical and textual coreference is based on types of anaphors:

Grammatical anaphors: relative pronouns, reflexive pronouns, reciprocity pronouns, restored (surface-unexpressed) "subjects" of infinitive verbs below verbs of control,

Textual anaphors: personal and possessive pronouns, demonstrative pronouns.

The data in the PDT 2.0 are divided into three groups: training set (80%), development test set (10%), and evaluation test set (10%). The training and development test set can be freely exploited, while the evaluation test data should serve only for the very final evaluation of developed tools.

Table 1 shows the distribution of each anaphor type. The total number of coreference links in the PDT 2.0 data is 45,174.[5] Personal pronouns including those zero ones and possessive pronouns form 37.4% of all anaphors in the entire corpus (16,888 links).

An example tectogrammatical tree with depicted coreference links (arrows) is presented in Figure 1. For the sake of simplicity, only three node attributes are displayed below the nodes: tectogrammatical lemma, functor, and semantic part of speech (tectogrammatical nodes themselves are complex data structures and around twenty attributes might be stored with them).

Tectogrammatical lemma is a canonical word form or an artificial value of a newly created node

[2] However, it should be noted that exact comparison is not possible here, since the tasks are slightly different for the two languages, especially because of typological differences between Czech and English (frequent pro-drop in Czech) and different information available in the underlying data resource on the other hand (manually annotated morphological and syntactical information available for Czech).

[3] http://ufal.mff.cuni.cz/pdt2.0/

[4] http://ucnk.ff.cuni.cz/

[5] In terms of the number of coreference links, PDT 2.0 is one of the largest existing manually annotated resources. Another comparably large resource is BBN Pronoun Coreference and Entity Type Corpus (Weischedel and Brunstein, 2005), which contains a stand-off annotation of coreference links in the Penn Treebank texts.

Type/Count	train	dtest	etest
Personal pron.	12,913	1,945	2,030
Relative pron.	6,957	948	1,034
Under-control pron.	6,598	874	907
Reflexive pron.	3,381	452	571
Demonstrative pron.	2,582	332	344
Reciprocity pron.	882	110	122
Other	320	35	42
Total	34,983	4,909	5,282

Table 1: Distribution of the different anaphor types in the PDT 2.0.

on the tectogrammatical level. E.g. the (artificial) tectogrammatical lemma #PersPron stands for personal (and possessive) pronouns, be they expressed on the surface (i.e., present in the original sentence) or restored during the annotation of the tectogrammatical tree structure (zero pronouns).

Functor captures the deep-syntactic dependency relation between a node and its governor in the tectogrammatical tree. According to FGD, functors are divided into *actants* (ACT – actor, PAT – patient, ADDR – addressee, etc.) and *free modifiers* (LOC – location, BEN – benefactor, RHEM – rhematizer, TWHEN – temporal modifier, APP – appurtenance, etc.).

Semantic parts of speech correspond to basic onomasiological categories (substance, feature, factor, event). The main semantic POS distinguished in PDT 2.0 are: semantic nouns, semantic adjectives, semantic adverbs and semantic verbs (for example, personal and possessive pronouns belong to semantic nouns).

3 Training data preparation

The training phase of both presented AR systems can be outlined as follows:

1. detect nodes which are anaphors (Section 3.1),
2. for each anaphor a_i, collect the set of antecedent candidates $\mathrm{Cand}(a_i)$ (Section 3.2),
3. for each anaphor a_i, divide the set of candidates into positive instances (true antecedents) and negative instances (Section 3.3),
4. for each pair of an anaphor a_i and an antecedent candidate $c_j \in \mathrm{Cand}(a_i)$, compute the feature vector $\Phi(c, a_i)$ (Section 3.4),
5. given the anaphors, their sets of antecedent candidates (with related feature vectors), and the division into positive and negative candidates, train the system for identifying the true antecedents among the candidates.

Steps 1-4 can be seen as training data preprocessing, and are very similar for both systems. System-specific details are described in Section 4 and Section 5 respectively.

3.1 Anaphor selection

In the presented work, only third person personal (and possessive) pronouns are considered,[6] be they expressed on the surface or reconstructed. We treat as anaphors all tectogrammatical nodes with lemma #PersPron and third person stored in the gram/person grammateme. More than 98 % of such nodes have their antecedents (in the sense of textual coreference) marked in the training data. Therefore we decided to rely only on this highly precise rule when detecting anaphors.[7]

In our example tree, the node #PersPron representing his on the surface and the node #PersPron representing the zero personal pronoun he will be recognized as anaphors.

3.2 Candidate selection

In both systems, the predicted antecedent of a given anaphor a_i is selected from an easy-to-compute set of antecedent candidates denoted as $\mathrm{Cand}(a_i)$. We limit the set of candidates to semantic nouns which are located either in the same sentence before the anaphor, or in the preceding sentence. Table 2 shows that if we disregard cataphoric and longer anaphoric links, we loose a chance for correct answer with only 6 % of anaphors.

[6] The reason is that antecedents of most other types of anaphors annotated in PDT 2.0 can be detected – given the tree topology and basic node attributes – with precision higher than 90 %, as it was shown already in Kučová and Žabokrtský (2005). For instance, antecedents of reflexive pronouns are tree-nearest clause subjects in most cases, while antecedents of relative pronouns are typically parents of the relative clause heads.

[7] It is not surprising that no discourse status model (as used e.g. in Denis and Baldridge (2008)) is practically needed here, since we limit ourselves to personal pronouns, which are almost always "discourse-old".

Antecedent location	Percnt.
Previous sentence	37 %
Same sentence, preceding the anaphor	57 %
Same sentence, following the anaphor	5 %
Other	1 %

Table 2: Location of antecedents with respect to anaphors in the training section of PDT 2.0.

3.3 Generating positive and negative instances

If the true antecedent of a_i is not present in $\mathrm{Cand(a_i)}$, no training instance is generated. If it is present, the sets of negative and positive instances are generated based on the anaphor. This preprocessing step differs for the two systems, because the classifier can be easily provided with more than one positive instance per anaphor, whereas the ranker can not.

In the classification-based system, all candidates belonging to the coreferential chain are marked as positive instances in the training data. The remaining candidates are marked as negative instances.

In the ranking-based system, the coreferential chain is followed from the anaphor to the nearest antecedent which itself is not an anaphor in grammatical coreference.[8] The first such node is put on the top of the training rank list, as it should be predicted as the winner (E.g., the nearest antecedent of the zero personal pronoun he in the example tree is the relative pronoun who, however, it is a grammatical anaphor, so its antecedent Brien is chosen as the winner instead). All remaining (negative) candidates are added to the list, without any special ordering.

3.4 Feature extraction

Our model makes use of a wide range of features that are obtained not only from all three levels of the PDT 2.0 but also from the Czech National Corpus and the EuroWordNet. Each training or testing instance is represented by a feature vector. The features describe the anaphor, its antecedent candidate and their relationship, as well as their contexts. All features are listed in Table 4 in the Appendix.

When designing the feature set on personal pronouns, we take into account the fact that Czech personal pronouns stand for persons, animals and things, therefore they agree with their antecedents in many attributes and functions. Further we use the knowledge from the Lappin and Leass's algorithm (Lappin and Leass, 1994), the Mitkov's robust, knowledge-poor approach (Mitkov, 2002), and the theory of topic-focus articulation (Kučová et al., 2005). We want to take utmost advantage of information from the antecedent's and anaphor's node on all three levels as well.

Distance: Numeric features capturing the distance between the anaphor and the candidate, measured by the number of sentences, clauses, tree nodes and candidates between them.

Morphological agreement: Categorial features created from the values of tectogrammatical gender and number[9] and from selected morphological categories from the positional tag[10] of the anaphor and of the candidate. In addition, there are features indicating the strict agreement between these pairs and features formed by concatenating the pair of values of the given attribute in the two nodes (e.g., masc_neut).

Agreement in dependency functions: Categorial features created from the values of tectogrammatical functor and analytical functor (with surface-syntactic values such as Sb, Pred, Obj) of the anaphor and of the candidate, their agreement and joint feature. There are two more features indicating whether the candidate/anaphor is an actant and whether the candidate/anaphor is a subject on the tectogrammatical level.[11]

Context: Categorial features describing the context of the anaphor and of the candidate:

- parent – tectogrammatical functor and the semantic POS of the effective parent[12] of the

[8] Grammatical anaphors are skipped because they usually do not provide sufficient information (e.g., reflexive pronouns provide almost no cues at all). The classification approach does not require such adaptation – it is more robust against such lack of information as it treats the whole chain as positive instances.

[9] Sometimes gender and number are unknown, but we can identify the gender and number of e.g. relative or reflexive pronouns on the tectogrammatical level thanks to their antecedent.

[10] A positional tag from the morphological level is a string of 15 characters. Every positions encodes one morphological category using one character.

[11] A subject on the tectogrammatical level can be a node with the analytical functor Sb or with the tectogrammatical functor Actor in a clause without a subject.

[12] The "true governor" in terms of dependency relations.

anaphor and the candidate, their agreement and joint feature; a feature indicating the agreement of both parents' tectogrammatical lemma and their joint feature; a joint feature of the pair of the tectogrammatical lemma of the candidate and the effective parent's lemma of the anaphor; and a feature indicating whether the candidate and the anaphor are siblings.[13]

- coordination – a feature that indicates whether the candidate is a member of a coordination and a feature indicating whether the anaphor is a possessive pronoun and is in the coordination with the candidate

- collocation – a feature indicating whether the candidate has appeared in the same collocation as the anaphor within the text[14] and a feature that indicates the collocation assumed from the Czech National Corpus.[15]

- boundness – features assigned on the basis of contextual boundness (available in the tectogrammatical trees) {contextually bound, contrastively contextually bound, or contextually non-bound}[16] for the anaphor and the candidate; their agreement and joint feature.

- frequency – 1 if the candidate is a denotative semantic noun and occurs more than once within the text; otherwise 0.

Semantics: Semantically oriented feature that indicates whether the candidate is a person name for the present and a set of 63 binary ontological attributes obtained from the EuroWordNet.[17] These attributes determine the positive or negative relation between the candidate's lemma and the semantic concepts.

4 Classifier-based system

Our classification approach uses C5.0, a successor of C4.5 (Quinlan, 1993), which is probably the most widely used program for inducing decision trees. Decision trees are used in many AR systems such as Aone and Bennett (1995), Mccarthy and Lehnert (1995), Soon et al. (2001), and Ng and Cardie (2002).[18]

Our classifier-based system takes as input a set of feature vectors as described in Section 3.4 and their classifications (1 – true antecedent, 0 – non-antecedent) and produces a decision tree that is further used for classifying new pairs of candidate and anaphor.

The classifier antecedent selection algorithm works as follows. For each anaphor a_i, feature vectors $\Phi(c, a_i)$ are computed for all candidates $c \in \mathrm{Cand}(a_i)$ and passed to the trained decision tree. The candidate classified as positive is returned as the predicted antecedent. If there are more candidates classified as positive, the nearest one is chosen.

If no candidate is classified as positive, a system of handwritten fallback rules can be used. The fallback rules are the same rules as those used in the baseline system in Section 6.2.

5 Ranker-based system

In the ranker-based AR system, every training example is a pair (a_i, y_i), where a_i is the anaphoric expression and y_i is the true antecedent. Using the candidate extraction function Cand, we aim to rank the candidates so that the true antecedent would always be the first candidate on the list. The ranking is modeled by a linear model of the features described in Section 3.4. According to the model, the antecedent $\hat{y}_i$ for an anaphoric expression a_i is found as:

$$\hat{y}_i = \underset{c \in \mathrm{Cand}(a_i)}{\mathrm{argmax}} \; \Phi(c, a_i) \cdot \overrightarrow{w}$$

The weights $\overrightarrow{w}$ of the linear model are trained using a modification of the averaged perceptron al-

[13] Both have the same effective parent.

[14] If the anaphor's effective parent is a verb and the candidate is a denotative semantic noun and has appeared as a child of the same verb and has had the same functor as the anaphor.

[15] The probability of the candidate being a subject preceding the verb, which is the effective parent of the anaphor.

[16] Contextual boundness is a property of an expression (be it expressed or absent in the surface structure of the sentence) which determines whether the speaker (author) uses the expression as given (for the recipient), i.e. uniquely determined by the context.

[17] The Top Ontology used in EuroWordNet (EWN) contains the (structured) set of 63 basic semantic concepts like Place, Time, Human, Group, Living, etc. For the majority of English synsets (set of synonyms, the basic unit of EWN), the appropriate subset of these concepts are listed. Using the Inter Lingual Index that links the synsets of different languages, the set of relevant concepts can be found also for Czech lemmas.

[18] Besides C5.0, we plan to use also other classifiers in the future (especially Support Vector Machine, which is often employed in AR experiments, e.g. by Ng (2005) and Yang et al. (2006)) in order to study how the classifier choice influences the AR system performance on our data and feature sets.

gorithm (Collins, 2002). This is averaged perceptron learning with a modified loss function adapted to the ranking scenario. The loss function is tailored to the task of correctly ranking the true antecedent, the ranking of other candidates is irrelevant. The algorithm (without averaging the parameters) is listed as Algorithm 1. Note that the training instances where $y_i \notin \mathrm{Cand}(a_i)$ were excluded from the training.

input : N training examples (a_i, y_i),
number of iterations T
init : $\overrightarrow{w} \leftarrow \overrightarrow{0}$;
for $t \leftarrow 1$ **to** T, $i \leftarrow 1$ **to** N **do**
$\hat{y}_i \leftarrow \mathrm{argmax}_{c \in \mathrm{Cand}(a_i)} \Phi(c, a_i) \cdot \overrightarrow{w}$;
if $\hat{y}_i \neq y_i$ **then**
$\overrightarrow{w} = \overrightarrow{w} + \Phi(y_i, a_i) - \Phi(\hat{y}_i, a_i)$;
end
end
output: weights $\overrightarrow{w}$

Algorithm 1: Modified perceptron algorithm for ranking. Φ is the feature extraction function, a_i is the anaphoric expression, y_i is the true antecedent.

Antecedent selection algorithm using a ranker: For each third person pronoun create a feature vector from the pronoun and the semantic noun preceding the pronoun and is in the same sentence or in the previous sentence. Use the trained ranking features weight model to get out the candidate's total weight. The candidate with the highest features weight is identified as the antecedent.

6 Experiments and evaluation

6.1 Evaluation metrics

For the evaluation we use the standard metrics:[19]

$$\text{Precision} = \frac{\text{number of correctly predicted anaphoric third person pronouns}}{\text{number of all predicted third person pronouns}}$$

$$\text{Recall} = \frac{\text{number of correctly predicted anaphoric third person pronouns}}{\text{number of all anaphoric third person pronouns}}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

We consider an anaphoric third person pronoun to be correctly predicted when we can successfully indicate its antecedent, which can be any antecedent from the same coreferential chain as the anaphor.

Both the AR systems were developed and tested on PDT 2.0 training and development test data. Finally they were tested on evaluation test data for the final scoring, summarized in Section 6.3.

6.2 Baseline system

We have made some baseline rules for the task of AR and tested them on the PDT 2.0 evaluation test data. Their results are reported in Table 3. Baseline rules are following: For each third person pronoun, consider all semantic nouns which precede the pronoun and are not further than the previous sentence, and:

- select the nearest one as its antecedent (BASE 1),
- select the nearest one which is a clause subject (BASE 2),
- select the nearest one which agrees in gender and number (BASE 3),
- select the nearest one which agrees in gender and number; if there is no such noun, choose the nearest clause subject; if no clause subject was found, choose the nearest noun (BASE 3+2+1).

6.3 Experimental results and discussion

Scores for all three systems (baseline, clasifier with and without fallback, ranker) are given in Table 3. Our baseline system based on the combination of three rules (BASE 3+2+1) reports results superior to the ones of the rule-based system described in Kučová and Žabokrtský (2005). Kučová and Žabokrtský proposed a set of filters for personal pronominal anaphora resolution. The list of candidates was built from the preceding and the same sentence as the personal pronoun. After applying each filter, improbable candidates were cut off. If there was more than one candidate left at the end, the nearest one to the anaphor was chosen as its antecedent. The reported final success rate was 60.4 % (counted simply as the number of correctly predicted links divided by the number of pronoun anaphors in the test data section).

An interesting point of the classifier-based system lies in the comparison with the rule-based

[19] Using simple accuracy would not be adequate, as there can be no link (or more than one) leading from an anaphor in the annotated data. In other words, finding whether a pronoun has an antecedent or not is a part of the task. A deeper discussion about coreference resolution metrics can be found in Luo (2005).

Rule	P	R	F
BASE 1	17.82%	18.00%	17.90%
BASE 2	41.69%	42.06%	41.88%
BASE 3	59.00%	59.50%	59.24%
BASE 3+2+1	62.55%	63.03%	62.79%
CLASS	69.9%	70.44%	70.17%
CLASS+3+2+1	76.02%	76.60%	76.30%
RANK	79.13%	79.74%	**79.43%**

Table 3: Precision (P), Recall (R) and F-measure (F) results for the presented AR systems.

system of Nguy and Žabokrtský (2007). Without the rule-based fallback (CLASS), the classifier falls behind the Nguy and Žabokrtský's system (74.2%), while including the fallback (CLASS+3+2+1) it gives better results.

Overall, the ranker-based system (RANK) significantly outperforms all other AR systems for Czech with the f-score of 79.43%. Comparing with the model for third person pronouns of Denis and Baldridge (2008), which reports the f-score of 82.2%, our ranker is not so far behind. It is important to say that our system relies on manually annotated information[20] and we solve the task of anaphora resolution for third person pronouns on the tectogrammatical level of the PDT 2.0. That means these pronouns are not only those expressed on the surface, but also artificially added (reconstructed) into the structure according to the principles of FGD.

7 Conclusions

In this paper we report two systems for AR in Czech: the classifier-based system and the ranker-based system. The latter system reaches f-score 79.43% on the Prague Dependency Treebank test data and significantly outperforms all previously published results. Our results support the hypothesis that ranking approaches are more appropriate for the AR task than classification approaches.

References

Chinatsu Aone and Scott William Bennett. 1995. Evaluating automated and manual acquisition of anaphora resolution strategies. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 122–129, Morristown, NJ, USA. Association for Computational Linguistics.

António Branco, Tony McEnery, Ruslan Mitkov, and Fátima Silva, editors. 2007. *Proceedings of the 6th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2007)*, Lagos (Algarve), Portugal. CLUP-Center for Linguistics of the University of Oporto.

Eugene Charniak and Micha Elsner. 2009. EM works for pronoun anaphora resolution. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 148–156, Athens, Greece, March. Association for Computational Linguistics.

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of EMNLP*, volume 10, pages 1–8.

Pascal Denis and Jason Baldridge. 2007. A ranking approach to pronoun resolution. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI2007)*, pages 1588–1593, Hyderabad, India, January 6–12.

Pascal Denis and Jason Baldridge. 2008. Specialized models and ranking for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP2008)*, pages 660–669, Honolulu, Hawaii, USA, October 25–27.

Jan Hajič, et al. 2006. Prague Dependency Treebank 2.0. CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2006T01, Philadelphia.

Lucie Kučová and Zdeněk Žabokrtský. 2005. Anaphora in Czech: Large Data and Experiments with Automatic Anaphora. *LNCS/Lecture Notes in Artificial Intelligence/Proceedings of Text, Speech and Dialogue*, 3658:93–98.

Lucie Kučová, Kateřina Veselá, Eva Hajičová, and Jiří Havelka. 2005. Topic-focus articulation and anaphoric relations: A corpus based probe. In Klaus Heusinger and Carla Umbach, editors, *Proceedings of Discourse Domains and Information Structure workshop*, pages 37–46, Edinburgh, Scotland, UK, Aug. 8-12.

Shalom Lappin and Herbert J. Leass. 1994. "an algorithm for pronominal anaphora resolution". *Computational Linguistics*, 20(4):535–561.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32, Morristown, NJ, USA. Association for Computational Linguistics.

[20] In the near future, we plan to re-run the experiments using sentence analyses created by automatic tools (all needed tools are available in the TectoMT software framework (Žabokrtský et al., 2008)) instead of manually created analyses, in order to examine the sensitivity of the AR system to annotation quality.

J Mccarthy and Wendy G. Lehnert. 1995. Using decision trees for coreference resolution. In *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1050–1055.

Marie Mikulová et al. 2005. Anotace na tektogramatické rovině Pražského závislostního korpusu. Anotátorská příručka (t-layer annotation guidelines). Technical Report TR-2005-28, ÚFAL MFF UK, Prague, Prague.

Ruslan Mitkov. 2002. *Anaphora Resolution*. Longman, London.

Václav Němčík. 2006. Anaphora Resolution. Master's thesis, Faculty of Informatics, Masaryk University.

Vincent Ng and Claire Cardie. 2002. Improving machine learning approaches to coreference resolution. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 104–111, Morristown, NJ, USA. Association for Computational Linguistics.

Vincent Ng. 2005. Supervised ranking for pronoun resolution: Some recent improvements. In Manuela M. Veloso and Subbarao Kambhampati, editors, *AAAI*, pages 1081–1086. AAAI Press / The MIT Press.

Vincent Ng. 2008. Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP2008)*, pages 640–649, Honolulu, Hawaii, USA.

Giang Linh Nguy and Zdeněk Žabokrtský. 2007. Rule-based approach to pronominal anaphora resolution applied on the prague dependency treebank 2.0 data. In Branco et al. (Branco et al., 2007), pages 77–81.

Giang Linh Nguy. 2006. Proposal of a Set of Rules for Anaphora Resolution in Czech. Master's thesis, Faculty of Mathematics and Physics, Charles University.

Jarmila Panevová. 1991. Koreference gramatická nebo textová? In *Etudes de linguistique romane et slave*. Krakow.

J. Ross Quinlan. 1993. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Petr Sgall, Eva Hajičová, and Jarmila Panevová. 1986. *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*. D. Reidel Publishing Company, Dordrecht.

Petr Sgall. 1967. *Generativní popis jazyka a česká deklinace*. Academia, Prague, Czech Republic.

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

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. TectoMT: Highly Modular MT System with Tectogrammatics Used as Transfer Layer. In *Proceedings of the 3rd Workshop on Statistical Machine Translation, ACL*.

Ralph Weischedel and Ada Brunstein. 2005. BBN Pronoun Coreference and Entity Type Corpus. CD-ROM, Linguistic Data Consortium, LDC Catalog No.: LDC2005T33, Philadelphia.

Xiaofeng Yang, Guodong Zhou, Jian Su, and Chew Lim Tan. 2003. Coreference resolution using competition learning approach. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 176–183, Morristown, NJ, USA. Association for Computational Linguistics.

Xiaofeng Yang, Jian Su, and Chew Lim Tan. 2006. Kernel-based pronoun resolution with structured syntactic knowledge. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL2006)*, pages 41–48, Sydney, Australia, July 17–21.

A Appendix

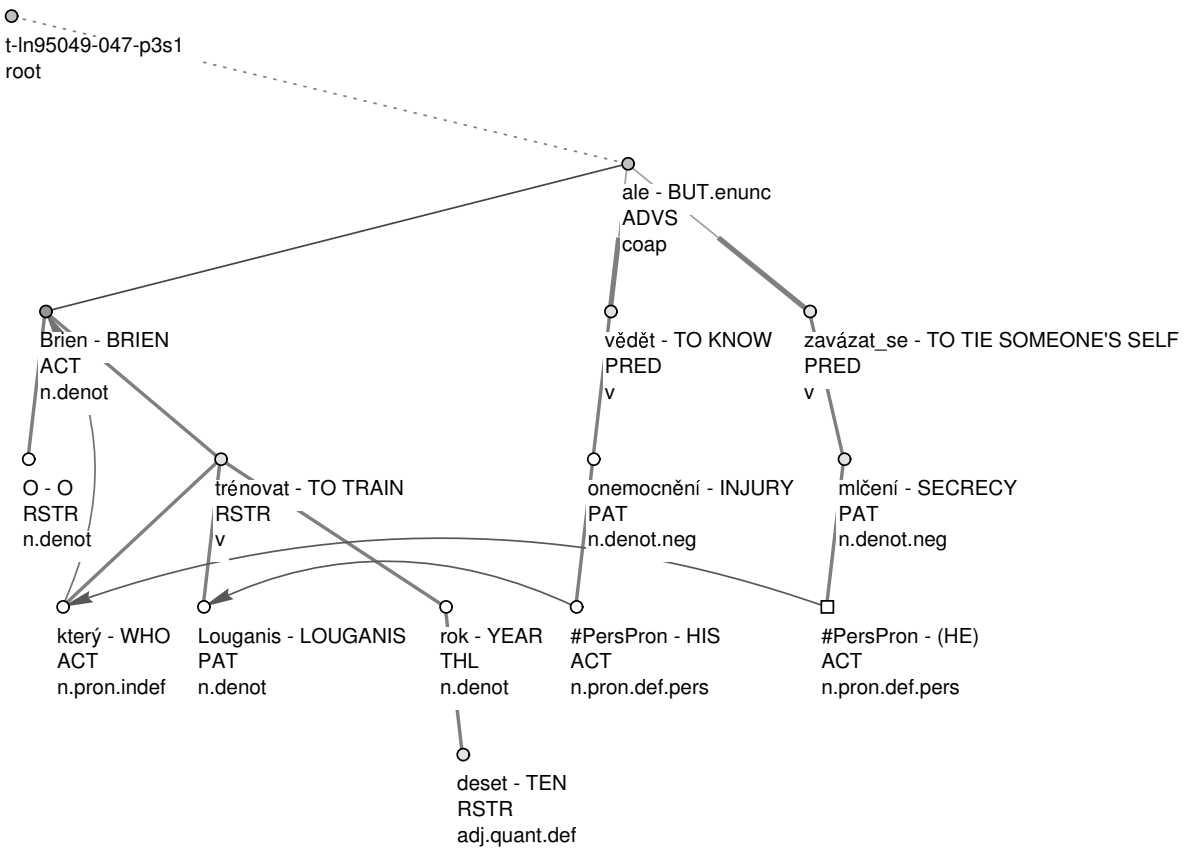


Figure 1: Simplified tectogrammatical tree representing the sentence *O'Brien, který Louganise trénoval deset let, o jeho onemocnění věděl, ale zavázal se mlčením.* (Lit.: O'Brien, who Louganis trained for ten years, about his injury knew, but (he) tied himself to secrecy.) Note two coreferential chains {Brien, who, (he)} and {Louganis, his}.

Distance	
sent_dist	sentence distance between c and a_i
clause_dist	clause distance between c and a_i
node_dist	tree node distance between c and a_i
cand_ord	mention distance between c and a_i
Morphological Agreement	
gender	t-gender of c and a_i, agreement, joint
number	t-number of c and a_i, agreement, joint
apos	m-POS of c and a_i, agreement, joint
asubpos	detailed POS of c and a_i, agreement, joint
agen	m-gender of c and a_i, agreement, joint
anum	m-number of c and a_i, agreement, joint
acase	m-case of c and a_i, agreement, joint
apossgen	m-possessor's gender of c and a_i, agreement, joint
apossnum	m-possessor's number of c and a_i, agreement, joint
apers	m-person of c and a_i, agreement, joint
Functional Agreement	
afun	a-functor of c and a_i, agreement, joint
fun	t-functor of c and a_i, agreement, joint
act	c/a_i is an actant, agreement
subj	c/a_i is a subject, agreement
Context	
par_fun	t-functor of the parent of c and a_i, agreement, joint
par_pos	t-POS of the parent of c and a_i, agreement, joint
par_lemma	agreement between the parent's lemma of c and a_i, joint
clem_aparlem	joint between the lemma of c and the parent's lemma of a_i
c_coord	c is a member of a coordination
app_coord	c and a_i are in coordination & a_i is a possessive pronoun
sibl	c and a_i are siblings
coll	c and a_i have the same collocation
cnk_coll	c and a_i have the same CNC collocation
tfa	contextual boundness of c and a_i, agreement, joint
c_freq	c is a frequent word
Semantics	
cand_pers	c is a person name
cand_ewn	semantic position of c's lemma within the EuroWordNet Top Ontology

Table 4: Features used by the perceptron-based model

Spoken Tutorial Dialogue and the Feeling of Another's Knowing

Diane Litman
University of Pittsburgh
Pittsburgh, PA 15260 USA
litman@cs.pitt.edu

Kate Forbes-Riley
University of Pittsburgh
Pittsburgh, PA 15260 USA
forbesk@cs.pitt.edu

Abstract

We hypothesize that monitoring the accuracy of the "feeling of another's knowing" (FOAK) is a useful predictor of tutorial dialogue system performance. We test this hypothesis in the context of a wizarded spoken dialogue tutoring system, where student learning is the primary performance metric. We first present our corpus, which has been annotated with respect to student correctness and uncertainty. We then discuss the derivation of FOAK measures from these annotations, for use in building predictive performance models. Our results show that monitoring the accuracy of FOAK is indeed predictive of student learning, both in isolation and in conjunction with other predictors.

1 Introduction

Detecting and exploiting knowledge of a speaker's uncertainty has been studied in several research communities. Spoken language researchers have identified statistically significant relationships between speaker uncertainty and linguistic properties of utterances such as prosody and lexical content (Liscombe et al., 2005; Dijkstra et al., 2006; Pon-Barry, 2008). Spoken dialogue researchers in turn are studying whether responding to user states such as uncertainty can improve system performance as measured by usability and efficiency (Tsukahara and Ward, 2001; Pon-Barry et al., 2006; Forbes-Riley and Litman, 2009a). In the psycholinguistics community, uncertainty has been studied in the context of metacognitive abilities, e.g. the ability to monitor the accuracy of one's own knowledge ("Feeling of Knowing" (FOK)), and the ability to monitor the FOK of someone else ("Feeling of Another's Knowing" (FOAK)) (Smith and Clark, 1993; Brennan and Williams, 1995).

Here we take a spoken dialogue systems perspective on FOAK, and investigate whether monitoring the accuracy of FOAK is a useful construct for predictive performance modeling. Our study uses data previously collected with a wizarded spoken dialogue tutoring system, where student learning is the primary performance metric. Section 2 reviews several relevant constructs and measures from the area of metacognition. Section 3 introduces our dialogue corpus and its user correctness and uncertainty annotations. Section 4 presents our method for measuring monitoring accuracy of FOAK from these annotations, while Section 5 shows how we use these measures to build predictive performance models. Our results show that monitoring the accuracy of FOAK is indeed a significant positive predictor of learning, both in isolation and over and above other predictors. As discussed in Section 6, increasing monitoring accuracy of FOAK is thus one avenue for also potentially increasing performance, which we plan to explore in future versions of our system.

2 Feeling of Another's Knowing

"Feeling of knowing" (FOK) refers to peoples' ability to accurately monitor their own knowledge, e.g. to know whether they have answered a question correctly. Psycholinguistics research has shown that speakers display FOK in conversation using linguistic cues such as filled pauses and prosody (Smith and Clark, 1993). Of perhaps more relevance to dialogue systems, research has also shown that *listeners* can use the same cues to monitor the FOK of someone else, i.e. *"feel-*

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 286–289,
Queen Mary University of London, September 2009.

ing of another's knowing" (FOAK) (Brennan and Williams, 1995).

To quantify knowledge monitoring, measures of *monitoring accuracy* have been proposed. For example, consider an FOK experimental paradigm, where subjects 1) respond to a set of general knowledge questions, 2) take a FOK survey, judging whether or not[1] they think they would recognize the answer to each question in a multiple choice test, and 3) take such a recognition test. As shown in Figure 1, such data can be summarized in an array where each cell represents a mutually exclusive option: the row labels represent the possible FOK judgments (Y/N), while the columns represent the possible results of the multiple choice test (Y/N).

	Recognition=Y	Recognition=N
Judgment=Y	a	b
Judgment=N	c	d

$$\textbf{Gamma} = \frac{(a)(d)-(b)(c)}{(a)(d)+(b)(c)} \qquad \textbf{HC} = \frac{(a+d)-(b+c)}{(a+d)+(b+c)}$$

Figure 1: Measuring Monitoring Accuracy.

Given such an array, the relationship between the correctness and the judgment of FOK for answers can be measured using the standard formulas in Figure 1: **Gamma** and the Harmann coefficient **(HC)** measure relative and absolute knowledge monitoring accuracy, respectively. We use both metrics in the performance modeling experiments described in Section 5, as both measures have been well studied and there is a lack of consensus regarding their relative benefits (e.g. (Nietfeld et al., 2006)). Note that both range in value from -1 (no monitoring accuracy) to 1 (perfect monitoring accuracy).

3 Data

This research uses a corpus of 405 dialogues between 81 students and ITSPOKE-WOZ. The corpus was collected in an experiment showing that performance improved when ITSPOKE-WOZ responded to the uncertainty in student answers, over and above correctness (Forbes-Riley and Litman, 2009a).

ITSPOKE-WOZ is a semi-automatic version of ITSPOKE (**I**ntelligent **T**utoring **SPOKE**n dialogue system), which is a speech-enhanced version of the Why2-Atlas qualitative physics tutor (VanLehn et al., 2002). Recent versions of ITSPOKE have also been enhanced to detect and respond to student uncertainty, by parameterizing the dialogue manager so that it can adapt contingently based on student correctness and/or uncertainty (to test various ways of responding to uncertainty). In ITSPOKE-WOZ, a few components (speech recognition, language understanding, and uncertainty detection) have been replaced by a human "wizard", to test the upper-bound performance of adapting to uncertainty in a spoken dialogue system (i.e. without the noise introduced by speech and language processing).

The experimental procedure was as follows: Students (1) read background material, (2) took a multiple-choice pretest, (3) worked through five problems with ITSPOKE-WOZ (1 per dialogue), (4) took a survey, and (5) took a posttest similar to the pretest. Each student turn in the corpus was annotated[2] in real-time by the wizard during the experiment, using binary tags representing student *Uncertainty* (uncertain (**U**), nonuncertain (**nonU**)) and student *Correctness* (correct (**C**), incorrect (**I**)). Figure 2 shows an annotated dialogue excerpt.

ITSPOKE-WOZ$_6$: Now let's find the forces exerted on the car in the VERTICAL direction during the collision. First, what vertical force is always exerted on an object near the surface of the earth?

USER$_6$: gravity **(nonU,C)**

ITSPOKE-WOZ$_7$: Fine. In addition to the downward force of gravity, one other force is exerted on the car in the VERTICAL direction during the collision. We haven't discussed this force yet, because we only dealt with freefalling objects before this problem. But you may recall this other vertical force from the reading. What is it?

USER$_7$: um ... the friction from the air ... or the wind **(U,I)**

ITSPOKE-WOZ$_8$: Great try, but I think you know there's a mistake in your answer. Let's work it out. We know that gravity can't be the only vertical force on the car, because ...

Figure 2: Annotated Corpus Excerpt.

4 Predictors

To derive our performance models, we use **posttest** score as the dependent measure, and include **pretest** score among the independent measures. This allows us to measure learning *gain*, not just final test score. We also include additional independent measures computed from the wizard's uncertainty and correctness annotations,

[1] Likert scale rating schemes are also possible.

[2] In prior ITSPOKE corpora, this wizard displayed interannotator agreement of 0.85 and 0.62 Kappa on correctness and uncertainty, respectively (Forbes-Riley and Litman, 2008).

as detailed below. Note that all measures are computed on a per student basis (over all 5 dialogues). Table 1 shows means and standard deviations of all measures across all 81 students.

Measure	Mean	Std. Dev.
pretest	.51	.15
posttest	.75	.14
%C	.79	.09
%U	.23	.11
Gamma	.77	.17
HC	.59	.16

Table 1: Descriptive Corpus Statistics.

The percentage of student turns annotated as correct (**%C**) and as uncertain (**%U**) normalize the raw counts of the wizard's **C** and **U** annotations. Similar measures predict learning in prior experiments by ourselves and others (e.g (Litman et al., 2009)) and thus serve as useful baselines. In our corpus, 79% of a student's turns are answered correctly on average, while 77% are answered without uncertainty.

The monitoring accuracy measures **Gamma** and **HC** were introduced in Section 2. To construct an array like that shown in Figure 1, we map the first and second rows to our uncertainty annotations **NonU** and **U**, and map the columns to our correctness annotations **C** and **I**. In (Dijkstra et al., 2006), high and low FOK/FOAK judgments are similarly associated with speaker certainty and uncertainty, respectively. Note that in our annotation scheme, **NonU** answers are either certain or neutral.

5 Results: Predicting Student Learning

Given the above measures, our first prediction experiment measures the partial Pearson's correlation between each of the independent measures and **posttest**, after first controlling for **pretest** to account for learning gain. Our goal here is examine the predictive utility of the correctness, uncertainty, and monitoring dimensions in isolation.

Table 2 shows the statistically significant results of the partial correlations. The table shows the independent measure, the corresponding Pearson's Correlation Coefficient (R), and the significance of the correlation (p). As can be seen, both monitoring measures are positively correlated with learning, with **HC** providing better predictive utility than **Gamma**. However, **%C** is even more predictive of learning than either monitoring measure. Interestingly, the uncertainty measure **%U** in and of itself does not show predictive utility in this data.

Measure	R	p
%C	.52	.00
Gamma	.36	.00
HC	.42	.00

Table 2: Partial Correlations with Posttest ($p < .05$).

Our second prediction experiment uses PARADISE to build a learning model that can potentially include multiple independent measures. As in prior PARADISE applications (e.g. (Möller, 2005)), we train the models using stepwise multiple linear regression, which automatically determines the measures to include in the model. Our goal here is to explore whether monitoring accuracy provides any added value to our correctness and uncertainty measures.

When all measures are made available for predicting learning, we see that monitoring accuracy as measured by **HC** does add value over and above correctness: the stepwise procedure includes **HC** in the model, as it significantly accounts for more variance than just including **%C** and **pretest**. In particular, the application of PARADISE shows that the following performance function provides the best significant training fit to our data ($R^2 = .71$, $p < .01$):

$$\textbf{postest} = .44{*}\textbf{\%C} + .21{*}\textbf{pretest} + .20{*}\textbf{HC}$$

The equation shows each selected measure and its (standardized) weight; larger weights indicate parameters with greater relative predictive power in accounting for **posttest** variance. **%C** is significant at $p < .01$, while **pretest** and **HC** are each significant at $p < .05$, with the coefficients all positive. Like the correlations, our regression demonstrates the predictive utility of the accuracy and monitoring measures, but not the uncertainty measure. The model further shows that while correctly answering the system's questions (**%C**) is predictive of learning, also including FOAK monitoring accuracy (**HC**) significantly increases the model's predictive power.

6 Conclusion and Future Directions

This paper explores whether knowledge monitoring accuracy is a useful construct for understanding dialogue system performance. In particular,

we demonstrate the utility of combining previously studied correctness and uncertainty annotations, using a measure of FOAK monitoring accuracy. Our results show that while the correctness of a user's response predicts learning, the uncertainty with which a user conveys a response does not. In contrast, the ability to monitor FOAK accuracy predicts learning, in isolation and over and above correctness. We believe that monitoring accuracy will be a relevant construct for other dialogue applications involving knowledge asymmetry, such as problem solving, instruction giving, and trouble shooting (e.g. (Janarthanam and Lemon, 2008)).

In future work we plan to use our results to inform a modification of our system aimed at improving inferred user knowledge monitoring abilities; we will better measure such improvements by incorporating FOK ratings into our testing. In addition, we recently found interactions between learning and both user domain expertise and gender (Forbes-Riley and Litman, 2009b); we will investigate whether similar interactions extend to knowledge monitoring metrics. Since our corpus contains dialogues with both uncertainty-adaptive and non-adaptive versions of ITSPOKE-WOZ, we also plan to examine whether differing dialogue strategies influence the learned predictive models. Finally, we plan to replicate our analyses in a dialogue corpus we recently collected using a fully automated version of our system.

Acknowledgements

NSF #0631930 supports this work. We thank H. Ai, P. Jordan, and the reviewers for helpful comments.

References

S. E. Brennan and M. Williams. 1995. The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of Memory and Language*.

C. Dijkstra, E. Krahmer, and M. Swerts. 2006. Manipulating uncertainty: The contribution of different audiovisual prosodic cues to the perception of confidence. In *Proc. Speech Prosody*.

K. Forbes-Riley and D. J. Litman. 2008. Analyzing dependencies between student certainness states and tutor responses in a spoken dialogue corpus. In L. Dybkjaer and W. Minker, editors, *Recent Trends in Discourse and Dialogue*. Springer.

K. Forbes-Riley and D. Litman. 2009a. Adapting to student uncertainty improves tutoring dialogues. In *Proc. Intl. Conf. on Artificial Intelligence in Education*.

K. Forbes-Riley and D. Litman. 2009b. A user modeling-based performance analysis of a wizarded uncertainty-adaptive dialogue system corpus. In *Proc. Interspeech*, Brighton, UK, September.

S. Janarthanam and O. Lemon. 2008. User simulations for online adaptation and knowledge-alignment in troubleshooting dialogue systems. In *Proc. SEMdial*.

J. Liscombe, J. Venditti, and J. Hirschberg. 2005. Detecting certainness in spoken tutorial dialogues. In *Proc. Interspeech*.

D. Litman, J. Moore, M. Dzikovska, and E. Farrow. 2009. Using natural language processing to analyze tutorial dialogue corpora across domains and modalities. In *Proc. Intl. Conf. on Artificial Intelligence in Education*.

S. Möller. 2005. Parameters for quantifying the interaction with spoken dialogue telephone services. In *Proc. SIGdial Workshop on Discourse and Dialogue*.

J. L. Nietfeld, C. K. Enders, and G. Schraw. 2006. A Monte Carlo comparison of measures of relative and absolute monitoring accuracy. *Educational and Psychological Measurement*.

H. Pon-Barry, K. Schultz, E. O. Bratt, B. Clark, and S. Peters. 2006. Responding to student uncertainty in spoken tutorial dialogue systems. *Intl. Journal of Artificial Intelligence in Education*.

H. Pon-Barry. 2008. Prosodic manifestations of confidence and uncertainty in spoken language. In *Proc. Interspeech*.

V. L. Smith and H. H. Clark. 1993. On the course of answering questions. *Journal of Memory and Language*.

W. Tsukahara and N. Ward. 2001. Responding to subtle, fleeting changes in the user's internal state. In *Proc. SIG-CHI on Human factors in computing systems*.

K. VanLehn, P. W. Jordan, C. Rosé, D. Bhembe, M. Böttner, A. Gaydos, M. Makatchev, U. Pappuswamy, M. Ringenberg, A. Roque, S. Siler, R. Srivastava, and R. Wilson. 2002. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. In *Proc. Intl. Conf. on Intelligent Tutoring Systems*.

Evaluating automatic extraction of rules for sentence plan construction

Amanda Stent
AT&T Labs – Research
Florham Park, NJ, USA
stent@research.att.com

Martin Molina
Department of Artificial Intelligence
Universidad Politécnica de Madrid, Spain
martin.molina@upm.es

Abstract

The freely available SPaRKy sentence planner uses hand-written weighted rules for sentence plan construction, and a user- or domain-specific second-stage ranker for sentence plan selection. However, coming up with sentence plan construction rules for a new domain can be difficult. In this paper, we automatically extract sentence plan construction rules from the RST-DT corpus. In our rules, we use only domain-independent features that are available to a sentence planner at runtime. We evaluate these rules, and outline ways in which they can be used for sentence planning. We have integrated them into a revised version of SPaRKy.

1 Introduction

Most natural language generation (NLG) systems have a pipeline architecture consisting of four core stages: content selection, discourse planning, sentence planning, and surface realization (Reiter and Dale, 2000; Rambow et al., 2001). A sentence planner maps from an input discourse plan to an output sentence plan. As part of this process it performs several tasks, including sentence ordering, sentence aggregation, discourse cue insertion and perhaps referring expression generation (Stent et al., 2004; Walker et al., 2007; Williams and Reiter, 2003).

The developer of a sentence planner must typically write rules by hand (e.g. (Stent et al., 2004; Walker et al., 2007)) or learn a domain-specific model from a corpus of training data (e.g. (Williams and Reiter, 2003)). Unfortunately, there are very few corpora annotated with discourse plans, and it is hard to automatically label a corpus for discourse structure. It is also hard to hand-write sentence planning rules starting from a "blank slate", as it were.

In this paper, we outline a method for extracting sentence plan construction rules from the only publicly available corpus of discourse trees, the RST Discourse Treebank (RST-DT) (Carlson et al., 2002). These rules use only domain-independent information available to a sentence planner at run-time. They have been integrated into the freely-available SPaRKy sentence planner. They serve as a starting point for a user of SPaRKy, who can add, remove or modify rules to fit a particular domain.

We also describe a set of experiments in which we look at each sentence plan construction task in order, evaluating our rules for that task in terms of coverage and discriminative power. We discuss the implications of these experiments for sentence planning.

The rest of this paper is structured as follows: In Section 2 we describe the sentence planning process using SPaRKy as an example. In Sections 3 through 5 we describe how we obtain sentence plan construction rules. In Section 6, we evaluate alternative rule sets. In Section 7, we describe our modifications to the SPaRKy sentence planner to use these rules. In Section 8 we conclude and present future work.

2 Sentence Planning in SPaRKy

The only publicly available sentence planner for data-to-text generation is SPaRKy (Stent et al., 2004). SPaRKy takes as input a discourse plan (a tree with rhetorical relations on the internal nodes and a proposition representing a text *span* on each leaf), and outputs one or more sentence plans

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 290–297,
Queen Mary University of London, September 2009.

(each a tree with discourse cues and/or punctuation on the internal nodes). SPaRKy is a two-stage sentence planner. First, possible sentence plans are *constructed* through a sequence of decisions made using only local information about single nodes in the discourse plan. Second, the possible sentence plans are *ranked* using a user- or domain-specific sentence plan ranker that evaluates the global quality of each sentence plan (Walker et al., 2007).

Sentence plan construction in SPaRKy involves three tasks: *span ordering*, *sentence aggregation* (deciding whether to realize a pair of propositions as a single clause, a single sentence, or two sentences), and *discourse cue selection*[1]. SPaRKy uses a single set of hand-written weighted rules to perform these tasks. In the current distributed version of SPaRKy, there are 20 rules covering 9 discourse cues (*and*, *because*, *but*, *however*, *on the other hand*, *since*, *while*, *with*, and the default, *period*). Each rule operates on the children of one rhetorical relation, and may impose an ordering, insert punctuation or merge two propositions, and/or insert a discourse cue. During sentence plan construction, SPaRKy walks over the input discourse plan, at each node finding all matching rules and applying one which it selects probabilistically according to the rule weights (with some randomness to permit variation).

While the developer of a NLG system will always have to adapt the sentence planner to his or her domain, it is often hard to come up with sentence planning rules "from scratch". As a result of the work described here a SPaRKy user will have a solid foundation for sentence plan construction.

3 Data

We use the Wall Street Journal Penn Treebank corpus (Marcus et al., 1993), which is a corpus of text annotated for syntactic structure. We also use two additional annotations done on (parts of) that corpus: PropBank (Kingsbury and Palmer, 2003), which consists of annotations for predicate-argument structure; and the RST-DT (Carlson et al., 2002), which consists of annotations for rhetorical structure.

We had to process this data into a form suitable for feature extraction. First, we produced a flattened form of the syntactic annotations, in which each word was labeled with its part-of-speech tag and the path to the root of the parse tree. Each word was also assigned indices in the sentence (so we could apply the PropBank annotations) and in the document (so we could apply the RST-DT annotations)[2].

Second, we attach to each word one or more labels from the PropBank annotations (each label consists of a predicate index, and either a predicate name or a semantic role type and index).

Third, we extract relation information from the RST-DT. For each relation, we extract the relation name, the types of each child ("Nucleus" or "Satellite"), and the start and end word indices for each child. Finally, we extract from the word-level annotations the marked-up words for each text span in each rhetorical relation.

4 Features

Features are individual rule conditions. In the standard NLG pipeline, no information about the realized text is available to the sentence planner. However, existing sentence planners use lexical and word sequence information to improve performance for a particular domain. Williams and Reiter (2003) appear to do surface realization before sentence planning, while Walker et al. (2007) perform surface realization between sentence plan construction and sentence plan ranking. We are concerned with sentence plan construction only; also, we want to produce sentence plan construction rules that are as domain-independent as possible. So we use no features that rely on having realized text. However, we assume that the input propositions have been fairly well fleshed-out, so that one has information about predicate-argument structure, tense, and the information status of entities to be realized.

A relation has a label as well as one or more child text *spans*. The features we extract from our data include both per-span and per-relation features. In our experiments we use a subset of these features which is fairly domain-independent and does not overly partition our data. The complete set of features (*full*) is as well as our *reduced* set are given in Table 1.

[1] SPaRKy also does some referring expression generation, in a single pass over each completed sentence plan.

[2] The Penn Treebank and the RST-DT segment words and punctuation slightly differently, which makes it hard to align the various annotations.

Feature type	*Full feature set*	*Reduced feature set*
Per-relation	relation, relation is leaf, parent relation, span coref, combined verb type class, combined verb type, identifier of shortest span, temporal order of spans	relation, relation is leaf, parent relation, span coref, combined verb type class, identifier of shortest span, temporal order of spans
Per-span, span identifier	span identifier	span identifier
Per-span, span length	number of NPs in span	
Per-span, span verb	verb type class, verb type, verb part of speech, verb is negated, verb has modal	
Per-span, arguments	argument status for ARG0 to ARG5 plus ARGM-{EXT, DIR, LOC, TMP, REC, PRD, ADV, MNR, CAU, PNC}	

Table 1: Features used in evaluation

4.1 Per-Span Features

We extract per-span features from *basic* spans (leaves of the RST tree) and from *complex* spans (internal nodes of the RST tree). For each span we compute: identifier, text, length, verb information, span argument information, discourse cue information, and span-final punctuation.

Identifier We need a way to refer to the child spans in the rules. For relations having only one child span of each type (Satellite or Nucleus), we order the spans by type. Otherwise, we order the spans alphabetically by span text. The span identifier for each child span is the index of the span in the resulting list.

Text We extract the text of the span, and the indices of its first and last words in the Penn Treebank. We only use this information during data extraction. However, in a system like that of Williams and Reiter (Williams and Reiter, 2003), where sentence planning is done after or with surface realization, these features could be used. They could also be used to train a sentence plan ranker for SPaRKy specific to the news domain.

Length We use the number of base NPs in the span (as we cannot rely on having the complete realization during sentence planning).

Verb We extract verb type, which can be *N/A* (there is no labeled predicate for the span), *stat* (the span's main verb is a form of "to be"), a single PropBank predicate (e.g. *create.01*), or *mixed* (the span contains more than one predicate). We then abstract to get the verb type class: *N/A*, *pb* (a PropBank predicate), *stat*, or *mixed*.

If the span contains a single predicate or multiple predicates all having the same part-of-speech tag, we extract that (as an indicator of tense). We also extract information about negation and modals (using the PropBank tags ARGM-NEG and ARGM-MOD).

Arguments We extract the text of the arguments of the predicate(s) in the span: ARG0 to ARG5, as well as ARGM-{EXT, DIR, LOC, TMP, REC, PRD, ADV, MNR, CAU, PNC}. We then abstract to get an approximation of information status. An argument status feature covers zero or more instantiations of the argument and can have the value *N/A* (no instantiations), *proper* (proper noun phrase(s)), *pro* (pronoun(s)), *def* (definite noun phrase(s)), *indef* (indefinite noun phrase(s)), *quant* (noun phrase (s) containing quantifiers), *other* (we cannot determine a value), or *mixed* (the argument instantiations are not all of the same type).

Discourse Cues We extract discourse cue information from basic spans and from the first basic span in complex spans. We identify discourse cue(s) appearing at the start of the span, inside the span, and at the end of the span. PropBank includes the argument label ARGM-DIS for discourse cues; however, we adopt a more expansive notion of discourse cue. We say that a discourse cue can be *either*: any sequence of words all labeled ARGM-DIS and belonging to the same predicate, anywhere in the span; *or* any cue from a (slightly expanded version of) the set of cues studied by Marcu (Marcu, 1997), if it appears at the start of a span, at the end of a span, or immediately before or after a comma, *and* if its lowest containing phrase tag is one of {ADJP, ADVP, CONJP, FRAG, NP-ADV, PP, UCP, SBAR, WH} *or* its part of speech tag is one of {CC, WDT}[3].

Punctuation We extract punctuation (*N/A* or . or ? or ! or ; or : or ,) at the end of the span.

[3]We constructed these rules by extracting from the WSJ Penn Treebank all instances of the cues in Marcu's list, and then examining instances where the word sequence was not actually a discourse cue. Some mistakes still occur in cue extraction.

4.2 Per-Relation Features

For each relation we compute: name, the combined verb type and verb class of the child spans, whether any argument instantiations in the child spans are coreferential, and which child span is shortest (or the temporal order of the child spans).
Relation, Parent Relation The core relation label for the relation and its parent relation (e.g. *attribution* for *attribution-e* and *attribution-n*).
Relation is Leaf True if child spans of the relation are leaf spans (not themselves relations).
Combined Verb The shared verb for the relation: the child spans' verb type if there is only one non-*N/A* verb type among the child spans; otherwise, *mixed*. We then abstract from the shared verb type to the shared verb type class.
Span Coreference We use the information PropBank gives about intra-sentential coreference. We do not employ any algorithm or annotation to identify inter-sentential coreference.
Shortest Span The identifier of the child span with the fewest base NPs.
Temporal Order of Spans For some relations (e.g. *sequence, temporal-before, temporal-after*), the temporal order is very important. For these relations we note the temporal order of the child spans rather than the shortest span.

5 Rule Extraction

Each rule we extract consists of a set of per-relation and per-span features (the *conditions*), and a *pattern* (the *effects*). The conditions contain either: the relation only, features from the reduced feature set, or features from the full feature set. The pattern can be an ordering of child spans, a set of between-span punctuation markers, a set of discourse cues, or an ordering of child spans mixed with punctuation markers and discourse cues. Each extracted rule is stored as XML.

We only extract rules for relations having two or more children. We also exclude RST-DT's *span* and *same-unit* relations because they are not important for our task. Finally, because the accuracy of low-level (just above the span) rhetorical relation annotation is greater than that of high-level relation annotation, we extract rules from two data sets: one only containing *first-level* relations (those whose children are all basic spans), and one containing *all* relations regardless of level in the RST tree. The output from the rule extraction process is six alternative rule sets for each sentence plan construction task: first-level or all data, with either the relation condition alone, the reduced feature set, or the full feature set.

Concession rule:
conditions:
type child="0": nucleus, type child="1": satellite, shortest: 0, isCoref: 0, isLeaf: 1, isSamePredClass: mixed, numChildren: 2, relation: concession, parentRel: antithesis
effects:
order: 1 0, punc child="1": comma, cues child="1": while
example:
(1) While some automotive programs have been delayed, (0) they have n't been canceled

Sequence rule:
conditions:
type child="0": nucleus, type child="1": nucleus, type child="2": nucleus, type child="3": nucleus, isCoref: 1, isLeaf: 1, isSamePredClass: mixed, numChildren: 4, relation: sequence, parentRel: circumstance, temporalOrder: 0 1 2 3
effects:
order: 0 1 2 3, punc child="0": comma, punc child="1": comma, punc child="2": n/a, cues child="3": and
example:
(0) when you can get pension fund money, (1) buy a portfolio, (2) sell off pieces off it (3) and play your own game

Purpose rule:
conditions:
type child="0": nucleus, type child="1": satellite, shortest: 0, isCoref: 0, isLeaf: 0, isSamePredClass: shared, numChildren: 2, relation: purpose, parentRel: list
effects:
order: 0 1, punc child="0": n/a, cues child="1": so
example:
(0) In a modern system the government 's role is to give the people as much choice as possible
(1) so they are capable of making a choice

Figure 1: Glosses of extracted sentence planning rules for three relations (reduced feature set)

The maximum number of patterns we could have is 7680 per relation, if we limit ourselves to condition sets, relation instances with only two child spans, and a maximum of one discourse cue to each span (two possible orderings for child spans * four possible choices for punctuation * 480 choices for discourse cue on each span). By contrast, for our *all* data set there are 5810 unique rules conditioned on the reduced feature set (109.6 per relation) and 292 conditioned on just the relation (5.5 per relation). Example rules are given in Figure 1. Even though the data constrains sentence planning choices considerably, we still have many rules (most differing only in discourse cues).

6 Rule Evaluation

6.1 On Evaluation

There are two basic approaches to NLG, *text-to-text generation* (in which a model learned from a text corpus is applied to produce new texts from text input) and *data-to-text generation* (in which non-text input is converted into text output). In text-to-text generation, there has been considerable work on sentence fusion and information ordering, which are partly sentence planning tasks. For evaluation, researchers typically compare automatically produced text to the original human-produced text, which is assumed to be "correct" (e.g. (Karamanis, 2007; Barzilay and McKeown, 2005; Marsi and Krahmer, 2005)). However, an evaluation that considers the only "correct" answer for a sentence planning task to be the answer in the original text is overly harsh. First, although we assume that all the possibilities in the human-produced text are "reasonable", some may be awkward or incorrect for particular domains, while other less frequent ones in the newspaper domain may be more "correct" in another domain. Our purpose is to lay out sentence plan construction possibilities, not to reproduce the WSJ authorial voice. Second, because SPaRKy is a two-stage sentence planner and we are focusing here on sentence plan construction, we can only evaluate the local decisions made during that stage, not the overall quality of SPaRKy's output.

Evaluations of sentence planning tasks for data-to-text generation have tended to focus solely on discourse cues (e.g. (Eugenio et al., 1997; Grote and Stede, 1998; Moser and Moore, 1995; Nakatsu, 2008; Taboada, 2006)). By contrast, we want good coverage for all core sentence planning tasks. Although Walker *et al.* performed an evaluation of SPaRKy (Stent et al., 2004; Walker et al., 2007), they evaluated the output from the sentence planner as a whole, rather than evaluating each stage separately. Williams and Reiter, in the work most similar to ours, examined a subset of the RST-DT corpus to see if they could use it to perform span ordering, punctuation selection, and discourse cue selection and placement. However, they assumed that surface realization was already complete, so they used lexical features. Their sentence planner is not publicly available.

In the following sections, we evaluate the information in our sentence plan construction rules in terms of *coverage* and *discriminative power*. The first type of evaluation allows us to assess the degree to which our rules are general and provide system developers with an adequate number of choices for sentence planning. The second type of evaluation allows us to evaluate whether our reduced feature set helps us choose from the available possibilities better than a feature set consisting simply of the relation (i.e. is the complicated feature extraction necessary). Because we include the full feature set in this evaluation, it can also be seen as a text-to-text generation type of evaluation for readers who would like to use the sentence planning rules for news-style text generation.

6.2 Coverage

In our evaluation of coverage, we count the number of relations, discourse cues, and patterns we have obtained, and compare against other data sets described in the research literature.

6.2.1 Relation Coverage

There are 57 unique core relation labels in the RST-DT. We exclude *span* and *same-unit*. Two others, *elaboration-process-step* and *topic-comment*, never occur with two or more child spans. Our *first-level* and *all* rules cover all of the remaining 53. The most frequently occurring relations are *elaboration-additional*, *list*, *attribution*, *elaboration-object-attribute*, *contrast*, *circumstance* and *explanation-argumentative*.

By contrast, the current version of SPaRKy covers only 4 relations (*justify, contrast, sequence*, and *infer*)[4].

Mann and Thompson originally defined 24 relations (Mann and Thompson, 1987), while Hovy and Maier listed about 70 (Hovy and Maier, 1992).

6.2.2 Discourse Cue Coverage

Our *first-level* rules cover 92 discourse cues, and our *all* rules cover 205 discourse cues. The most commonly occurring discourse cues in both cases are *and*, *but*, *that*, *when*, *as*, *who* and *which*.

By contrast, the current version of SPaRKy covers only about 9 discourse cues.

In his dissertation Marcu identified about 478 discourse cues. We used a modified version of Marcu's cue list to extract discourse cues from our corpus, but some of Marcu's discourse cues do not occur in the RST-DT.

[4]Curiously, only two of these relations (*contrast* and *sequence*) appear in the RST-DT data (although *infer* may be equivalent to *span*).

6.2.3 Sentence Plan Pattern Coverage

For the *first-level* data we have 140 unique sentence plan patterns using the relation condition alone, and 1767 conditioning on the reduced feature set. For the *all* data we have 292 unique patterns with relation condition alone and 5810 with the reduced feature set. Most patterns differ only in choice of discourse cue(s).

No system developer will want to examine all 5810 rules. However, she or he may wish to look at the patterns for a particular relation. In our use of SPaRKy, for example, we have extended the patterns for the *sequence* relation by hand to cover temporal sequences of up to seven steps.

6.3 Discriminative Power

In this evaluation, we train decision tree classifiers for each sentence plan construction task. We experiment with both the *first-level* and *all* data sets and with both the *reduced* and *full* feature sets. For each experiment we perform ten-fold cross-validation using the J48 decision tree implementation provided in Weka (Witten and Eibe, 2005) with its default parameters. We also report performance for a model that selects a pattern conditioning only on the relation. Finally, we report performance of a baseline which always selects the most frequent pattern.

We evaluate using 1-best classification accuracy, by comparing with the choice made in the Penn Treebank for that task. We test for significant differences between methods using Cochran's Q, followed by post-hoc McNemar tests if significant differences existed. We also report the features with information gain greater than 0.1.

6.3.1 Span Ordering

We have one input feature vector for each relation instance that has two children[5]. In the feature vector, child spans are ordered by their identifiers, and the pattern is either *0_1* (first child, then second child) or *1_0* (second child, then first child).

Classification accuracy for all methods is reported in Table 2. All methods perform significantly better than baseline ($p < .001$), and both the reduced and full feature sets give results significantly better than using the relation alone ($p < .001$). The full feature set performs significantly better than the reduced feature set for the *all* data set ($p < .001$), but not for the *first-level* data set.

Most of the relations have a strong preference for one ordering or the other. Most mistakes are made on those that don't (e.g. *attribution, list*).

	First-level	All
Baseline	71.8144	71.4356
Per-relation	84.2707	82.3894
Reduced	89.6092	90.3147
Full	90.2129	91.9666

Table 2: Span ordering classification accuracy. For first-level data, n = 3147. For all data, n = 10170. Labels = {*0_1, 1_0*}.

	First-level	All
Baseline	74.5154	50.4425
Per-relation	74.5154	64.2773
Reduced	77.8201	72.1731
Full	74.3883	66.1357

Table 3: Between-span punctuation classification accuracy. For first-level data, n = 3147. For all data, n = 10170. Labels = {*semicolon, comma, full, N/A*}.

6.3.2 Punctuation Insertion

We have one input feature vector for each relation instance that has two children. We assume that span ordering is performed prior to punctuation insertion, so the child spans are ordered as they appear in the data. The pattern is the punctuation mark that should appear between the two child spans (one of *N/A* or *comma* or *semicolon* or *full*[6]), which indicates whether the two children should be realized as separate sentences, as separate clauses, or merged.

Classification accuracy for all methods is reported in Table 3. For the *all* data set, all methods perform significantly better than baseline ($p < .001$), and both the reduced and full feature sets give results significantly better than using the relation alone ($p < .001$). Furthermore, the reduced feature set performs significantly better than the full feature set ($p < .001$). By contrast, for the *first-level* data set, the reduced feature set performs significantly better than all the other data sets, while there are no statistically significant differences in performance between the baseline, per-relation and full feature sets.

The most common type of error was misclassifying *comma, semicolon* or *full* as *N/A*: for the

[5] The number of relation instances with three or more child spans is less than 2% of the data. Removing these relations made it feasible for us to train classifiers without crashing Weka.

[6] *full* indicates a sentence boundary (. or ? or !).

	First-level	All
Baseline	62.6629	68.4267
Per-relation	68.605	70.1377
Reduced	73.6257	73.9135
Full	74.3565	74.5919

Table 4: Discourse cue classification accuracy. For first-level data, n = 3147 and no. labels = 92. For all data, n = 10170 and no. labels = 203.

first-level data this is what the models trained on the per-relation and full feature sets do most of the time. The second most common type of error was misclassifying *comma, semicolon* or *N/A* as *full*.

6.3.3 Discourse cue selection

We have one input feature vector for each relation instance having two children. We use the same features as in the previous experiment, and as in the previous experiment, we order the child spans as they appear in the data. The pattern is the first discourse cue appearing in the ordered child spans[7].

Classification accuracy for all methods is reported in Table 4. All methods perform significantly better than baseline ($p < .001$), and both the reduced and full feature sets give results significantly better than using the relation alone ($p < .001$). The performance differences between the reduced and full feature sets are not statistically significant for either data set.

For this task, 44 of the 92 labels in the *first-level data*, and 97 of the 203 labels in the *all data*, occurred only once. These cues were typically mislabeled. Commonly occurring labels were typically labeled correctly.

6.4 Discussion

Our methods for rule extraction are not general in the sense that they rely on having access to particular types of annotation which are not widely available nor readily obtainable by automatic means. However, our extracted rules have quite broad coverage and will give NLG system developers a jump start when using and adapting SPaRKy.

Our reduced feature set compares favorably in discriminative power to both our full feature set and the per-relation feature set. It achieves a very good fit to the input data for the span ordering task and a good fit to the input data for the punctuation and discourse cue insertion tasks, especially for the *first-level* data set. Factors affect performance include: the punctuation insertion data is highly imbalanced (by far the most common label is *N/A*), while for the discourse cue insertion task there is a problem of data sparsity.

[7] Some relations have multiple cues, either independent cues such as *but* and *also*, or cues that depend on each other such as *on the one hand* and *on the other hand*. Using all cues is infeasible, and there are too few span-internal and span-final cues to break up the cue classification for this evaluation.

7 Revised SPaRKy

One way to use these results would be to model the sentence planning task as a cascade of classifiers, but this method does not permit the system developer to add his or her own rules. So we continue to use SPaRKy, which is rule-based. We have made several changes to the Java version of SPaRKy to support application of our sentence plan construction rules. We modified the classes for storing and managing rules to read our XML rule format and process rule conditions and patterns. We stripped out the dependence on RealPro and added hooks for SimpleNLG (Gatt and Reiter, 2009). We modified the rule application algorithm so that users can choose to use a single rule set with patterns covering all three sentence planning tasks, or one rule set for each sentence planning task. Also, since there are now many rules, we give the user the option to specify which relations jSPaRKy should load rules for at each run.

Information about the revised jSparky, including how to obtain it, is available at *http://www.research.att.com/~stent/sparky2.0/* or by contacting the first author.

8 Conclusions and Future Work

In this paper we described how we extracted less domain-dependent sentence plan construction rules from the RST-DT corpus. We presented evaluations of our extracted rule sets and described how we integrated them into the freely-available SPaRKy sentence planner.

In future work, we will experiment with discourse cue clustering. We are also looking at alternative ways of doing sentence planning that permit a tighter interleaving of sentence planning and surface realization for improved efficiency and output quality.

References

R. Barzilay and K. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

L. Carlson, D. Marcu, and M. E. Okurowski. 2002. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the SIGdial workshop on discourse and dialogue*.

B. Di Eugenio, J. D. Moore, and M. Paolucci. 1997. Learning features that predict cue usage. In *Proceedings of the EACL*.

A. Gatt and E. Reiter. 2009. SimpleNLG: A realisation engine for practical applications. In *Proceedings of the European Workshop on Natural Language Generation*.

B. Grote and M. Stede. 1998. Discourse marker choice in sentence planning. In *Proceedings of the 9th International Workshop on Natural Language Generation*.

E. Hovy and E. Maier. 1992. Parsimonious or profligate: how many and which discourse structure relations? Available from http://handle.dtic.mil/100.2/ADA278715.

N. Karamanis. 2007. Supplementing entity coherence with local rhetorical relations for information ordering. *Journal of Logic, Language and Information*, 16(4):445–464.

P. Kingsbury and M. Palmer. 2003. PropBank: the next level of TreeBank. In *Proceedings of the Workshop on Treebanks and Lexical Theories*.

B. Lavoi and O. Rambow. 1997. A fast and portable realizer for text generation systems. In *Proceedings of ANLP*.

W. Mann and S. Thompson. 1987. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, Information Sciences Institute, Los Angeles, CA.

D. Marcu. 1997. *The rhetorical parsing, summarization, and generation of natural language texts*. Ph.D. thesis, Department of Computer Science, University of Toronto.

M. Marcus, M. A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

E. Marsi and E. Krahmer. 2005. Explorations in sentence fusion. In *Proceedings of the European Workshop on Natural Language Generation*.

M. Moser and J. D. Moore. 1995. Using discourse analysis and automatic text generation to study discourse cue usage. In *Proceedings of the AAAI 1995 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*.

C. Nakatsu. 2008. Learning contrastive connective in sentence realization ranking. In *Proceedings of SIGdial 2008*.

O. Rambow, S. Bangalore, and M. A. Walker. 2001. Natural language generation in dialog systems. In *Proceedings of HLT*.

E. Reiter and R. Dale. 2000. *Building natural language generation systems*. Cambridge University Press, Cambridge, UK.

A. Stent, R. Prasad, and M. A. Walker. 2004. Trainable sentence planning for complex information presentations in spoken dialog systems. In *Proceedings of the ACL*.

M. Taboada. 2006. Discourse markers as signals (or not) of rhetorical relations. *Journal of Pragmatics*, 38(4):567–592.

M. A. Walker, A. Stent, F. Mairesse, and R. Prasad. 2007. Individual and domain adaptation in sentence planning for dialogue. *Journal of Artificial Intelligence Research*, 30:413–456.

S. Williams and E. Reiter. 2003. A corpus analysis of discourse relations for natural language generation. In *Proceedings of Corpus Linguistics*.

I. Witten and F. Eibe. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2nd edition.

Eliciting interactional phenomena in human-human dialogues

Joakim Gustafson

KTH Speech Music & Hearing
jocke@speech.kth.se

Miray Merkes

KTH Speech Music & Hearing
miray@kth.se

Abstract

In order to build a dialogue system that can interact with humans in the same way as humans interact with each other, it is important to be able to collect conversational data. This paper introduces a dialogue recording method where an eavesdropping human operator sends instructions to the participants in an ongoing human-human task-oriented dialogue. The purpose of the instructions is to control the dialogue progression or to elicit interactional phenomena. The recordings were used to build a Swedish synthesis voice with disfluent diphones.

1 Background

Our research group have a long-standing interest in human conversational behaviour and a special interest in its mimicry and evaluation in spoken dialogue systems (Edlund et al., 2008). In human-human conversations both parties continuously and simultaneously contribute actively and interactively to the conversation. Listeners actively contribute by providing feedback during the other's speech, and speakers continuously monitor the reactions to their utterances (Clark, 1996). If spoken dialogue systems are to achieve the responsiveness and flexibility found in human-human interaction, it is essential that they process information incrementally and continuously rather than in turn sized chunks (Dohsaka & Shimazu, 1997, Skantze & Schlangen, 2009). These systems need to be able to stop speaking in different manners depending on whether it has finished what it planned to say or if it was interrupted mid-speech by the user. In order to be responsive, the system might also need to start talking before it has decided exactly what to say. In this case it has to be able to generate interactional cues that restrain the user from start speaking while the system plans the last part.

To date very few spoken dialogues systems can generate crucial and commonly used interactional cues. Adell et al. (2007) have developed a set of rules for synthesizing filled pauses and repetitions with PSOLA. Unit selection synthesizers are often used in dialogue systems, but a problem with these is that even though most databases have been carefully designed and read, they are not representative of "speech in use" (Campbell & Mokhiari, 2003). There are examples of synthesizers that have been trained on speech in use, like Sundaram & Narayanan (2003) that used a limited-domain dialogue corpus of transcribed human utterances as input for offline training of a machine learning system that could insert fillers and breathing at the appropriate places in new domain-related texts. However, these were synthesized with a unit selection voice that had been trained on lecture speech.

When modelling talk-in-use it is important to study representative data. The problem with studying real dialogues is that the interesting interactional phenomena often are sparsely occurring and very context dependent. When conducting research on spontaneous speech you have the option to use controlled or uncontrolled conditions. Anderson et al., 1991) recorded unscripted conversations in a map task exercise that had been carefully designed to elicit interactional phenomena. When using controlled conditions in a study you risk to manipulate the data, while in uncontrolled conditions there's a risk that the conversation goes out of hand which leads to a lot of unnecessary material (Bock, 1996). Bock suggests a set of eliciting methods to be used when studying disfluent speech. If the goal is to study speech errors and interruptions, a situation with two competing humans is useful. If the goal is to study hesitations and self-interruptions, distracting events can be used to disrupt the flow of speech.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 298–301,
Queen Mary University of London, September 2009.

This paper presents a new method for elicitation of interactional phenomena, with the goal of reducing the amount of necessary dialogue recordings. In this method an eavesdropping human operator sends instructions two subjects as they engage in a task-oriented dialogue. The purpose of these instructions is either to control the dialogue progression or to elicit certain interactional phenomena. The recordings from two sessions were used to build a synthesis voice with disfluent diphones. In a small synthesis study on generation of disfluent conversational utterances this voice was compared with a commercial Swedish diphone voice based on read speech. The subjects rated the created voice as more natural than the commercial voice.

2 Method

A dialogue collection environment has been developed that allows a human operator (Wizard) to eavesdrop an ongoing computer-mediated human-human conversation. It also allows the Wizard to send instructions to the interlocutors during their conversation, see Figure 1. The purpose of the instructions is to control the progression of the task-oriented dialogue and to elicit interactional phenomena, e.g. interruptions and hesitations. The Wizard has access to graphical and textual instructions. Graphical instructions are pictures that are manipulated or text labels that are changed. Textual instructions are scrolled in from the right at the bottom of the screen. They can be of three categories: *Emotional* instructions that tell the receiver to act emotional (e.g. act grumpy); *Task-related* instructions that require the receiver to initiate a certain subtasks (e.g. buy a red car); and *Dialogue flow related* instructions that tell the receiver to change his way of speaking, (e.g. speak fast, do not pause).

3 The pilot study

The DEAL system is a speech-enabled computer game currently under development, that will be used for conversational training for second language learners of Swedish (Hjalmarsson, 2008). In this system an embodied conversational character (ECA) acts as a shopkeeper in a flea trade-market and the user is a customer. The developed environment was adapted to the DEAL domain, and in a pilot study two human subjects were instructed to act as shopkeeper and customer. They were given written persona descriptions and were then placed in separate rooms. They interacted via a three-party

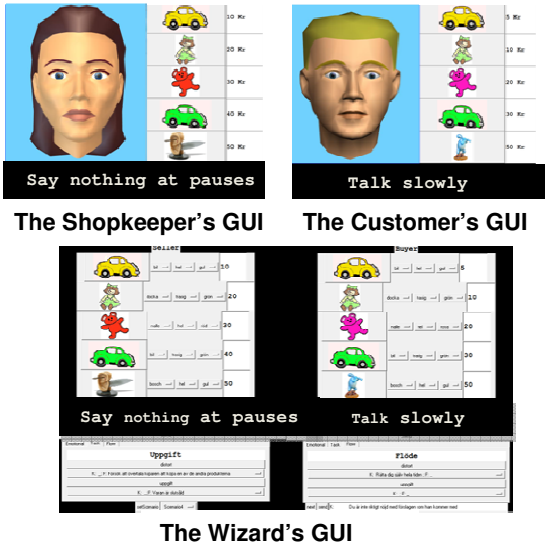


Figure 1. The GUIs used by the wizard and subjects.

Skype call, which allowed the Wizard to eavesdrop their conversation. In order to get a situation that was similar to the DEAL system, the subjects saw an avatar with lip movements driven by, and in synchrony with, the other subjects' speech. In order to achieve this, the SynFace system was used, which introduced a 200 ms delay in each direction (Beskow et al., 2004). Apart from the avatar the interfaces also contained pictures of objects currently for sale with accompanying prices, see Figure 1. At the bottom of the screen there was a black area where the subjects got the textual instructions from the Wizard.

The eavesdropping Wizard was placed in a third room, with an interface that allowed her to control the current set of objects and prices on the subjects' screens. The Wizard interface also contained an area for the textual instructions. In order to distort the dialogue flow some of the instructions involved sending instructions to both subjects at the same time. A main idea is to instruct one of the interlocutors to display a verbal behavior that will elicit interactional phenomena in the other dialogue partner's contributions. Table 1 shows some examples of the different types of textual instructions to the subjects and their intended effect on the shopkeeper party in an ongoing conversation. The Wizard interface also gave access to automated instructions that follows a pre-scripted manuscript in order to facilitate consistent instructions across different sessions. This also made it possible to transmit multiple successive instructions with high speed and a minimum risk of mistakes.

Shopkeeper reaction	Graphical	Emotional	Task related	Dialog flow related
Hesitation	Show an ambiguous picture (S)	Be wining and talk about how unfair life is (S)	Sell blue car (S) Buy red car (C)	Talk slowly (S) Say nothing at pauses (C)
Interruption	Change picture in mid speech (S)	Be a annoying customer (C)	Tell your price (S) Tell your price (C)	Speak without pauses (S) Try to speak all the time (C)
Change of sub-task	Show a picture (S)	Discuss the advantages of a certain item (S)	Sell the red car (S)	Ask a lot of questions (C) Answer with questions (S)

Table 1. Examples of instruction types and their intended reaction in the shopkeeper's subsequent turn(s). The receiver of the instruction is indicated by S (Shopkeeper) and C (Customer).

4 The effect of the Wizard's instruction

Two half-hour conversations were recorded where the same male subject (acting as shopkeeper) interacted with two different female subjects (acting as customers). The audio recordings were synchronized with the instructions that had been submitted by the Wizard during the conversation. The effects of the instructions were analyzed by inspecting both subjects' turns following an instruction from the Wizard. The analysis was focused on the disruptive effect of the instructions, and it showed that they often lead to turns that contained hesitations, interruptions and pauses. The task-related instructions lead to disfluent speech in half of the succeeding turns, while the dialogue flow related instructions, the emotional instructions and the graphical instructions led to disfluent turns in two thirds of the cases. The analysis of the instructions' effect on the disfluency rates revealed that the ones that changed the task while the subjects talked were very efficient, e.g. changing the price while it was discussed. The effect on the disfluency rates was most substantial when contradictive instructions were given to both subjects at the same time.

In order to get a baseline of disfluency rates in human-human dialogues in the current domain, the dialogue data was compared with data recorded in a previous DEAL recording. In this study 8 dialogues were recorded where two subjects role-played as a shopkeeper and a customer, but without the controlling Wizard used in the present study (Hjalmarsson, 2008). In these recordings approximately one third of the turns contained disfluent speech. This indicates that the disfluency rates found after the instructions in the current study are a higher than in the previous DEAL recording. Finally we analyzed the effect of the instructions on the dialogue progression. The instructions were very helpful in keeping the discussion going and the task oriented instructions provided useful guidance to the subjects in their role-playing.

5 A speech synthesis experiment

In a second experiment the goal was to evaluate two methods for collecting conversational data for building a corpus-based conversational speech synthesizer: collecting a controlled human-human role-playing dialogue or a recording a human that reads a dialogue transcription with tags for interruptions and hesitations. In this experiment the recordings of the male subject that acted as shopkeeper were used. 20 of his utterances that contained hesitations, interruptions and planned pauses were selected. New versions of these utterances were created, where the disruptions were removed. In order to verify that the disruptive sections could be synthesized in new places a set of test sentences were constructed that included their immediate contexts. Finally, new versions of the new test sentences were created, that had added tags for disruptions. All types of utterances were read by the original male speaker. Both the original dialogue recordings and the read utterances were phonetically transcribed and aligned in order to build a small diphone voice with the EXPROS tool (Gustafson & Edlund, 2008). This diphone voice contained fillers, truncated phonemes and audible breathing.

All types of utterances were re-synthesized with the newly created voice and with a Swedish commercial diphone voice that was trained on clear read speech. While re-synthesizing the original recordings all prosodic features (pitch, duration and loudness) were kept. The main difference between the two voices was the voice quality: the commercial voice is trained on clear read speech, while the new voice was created from the dialogue recordings contains both reduced and truncated diphones.

Secondly, a number of utterances were synthesized, where disfluent sections were inserted into fluently read sentences. For both voices the disfluent sections' original pitch, duration and loudness were kept. As in the previous case the main differ-

ence between the two cases is that the newly created also made use of its disfluent diphones. The disfluent sections were either taken from the original dialogue recordings or from the set of read sentences with tags for disfluencies.

6 Preliminary synthesis evaluation

16 subjects participated in a listening test, where they were told to focus on the disrupted parts of the utterances. They were instructed to indicate when they could detect the following disruptions: hesitation, pause, interruption and correction. They were also asked to assess on a six-graded likert scale how natural these sounded and how easy it was to detect the disrupted parts. Results show that disrupted utterances that were synthesized with the new voice were rated as natural in two thirds of the cases, while the ones that were generated with commercial synthesis voice, that lacked disfluent diphones, was rated as natural in half of the cases. Kruskal-Wallis rank sums were performed, and the interrupted utterances generated by new voice was significantly more natural than those generated with the commercial voice ($p=0.001$). When comparing how easy it was to detect the disrupted parts both versions are comparable (90% of them were easy to detect, with no significant difference).

In order to analyze the difference between real and pretended disruptions, the subjects were asked to compare re-synthesis of the of disrupted dialogue turns with corresponding read versions. They were asked to judge which of the two they thought contained a pretended disruption. When comparing re-synthesis of complete utterances from either of these types they were able to detect the version with pretended disruptions in 60% of the cases. In cases where the disfluent parts were moved to new fluently read sentences the users could not tell which version contained a pretended disruption. This is probably because they rated how the whole sentence sounded, rather than only the disrupted part. These differences were significant according to a chi-square test. Finally, the subjects' ability to identify the different types of disfluencies when synthesized by the two voices was compared. For both voices, about 80% of the hesitations and interruptions were correctly identified, while only 70% of the planned pauses were correctly identified. For both voices about 85% of the missed pauses were instead identified as hesitations or interruptions. For the new voice most of them were identified as hesitations, while they were mostly misinterpreted as interruptions for the commercial voice. The share of inserted interruptions is the only significant identification difference between the two voices. This is not surprising since they both used the pitch, power and durations from the original human recordings, while only the new voice also had access to truncated diphones.

This pilot study showed that the instructions from the Wizards were useful both to control the dialogue flow and to elicit interactional phenomena. Finally, the male participant reported that it was hard to pretend to be disfluent while reading dialogue transcripts where this was tagged.

Acknowledgements

This research is supported by MonAMI, an Integrated Project under the European Commission (IP-035147).

References

Adell, J., Bonafonte, A., & Escudero, D. (2007). Filled pauses in speech synthesis: towards conversational speech. In *Proc. of Conference on Text, Speech and Dialogue(LNAI 07)*

Anderson, A., Bader, M., Bard, E., Boyle, E., Doherty, G., Garrod, S., Isard, S., Kowtko, J., McAllister, J., Miller, J., Sotillo, C., Thompson, H., & Weinert, R. (1991). The HCRC Map Task corpus. *Language and Speech, 34*(4).

Beskow, J., Karlsson, I., Kewley, J., & Salvi, G. (2004). SYNFACE - A talking head telephone for the hearing-impaired. In Miesenberger, K., Klaus, J., Zagler, W., & Burger, D. (Eds.), *Computers Helping People with Special Needs*. Springer-Verlag.

Bock, K. (1996). Language production: Methods and methodologies. In *Psychonomic Bulletin and Review*.

Campbell, N., & Mokhiari, P. (2003). Using a Non-Spontaneous Speech Synthesiser as a Driver for a Spontaneous Speech Synthesiser. In *Proceedings of ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*. Tokyo, Japan.

Clark, H. H. (1996). *Using language.* Cambridge, UK: Cambridge University Press.

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*.

Edlund, J., Gustafson, J., Heldner, M., & Hjalmarsson, A. (2008). Towards human-like spoken dialogue systems. *Speech Communication, 50*(8-9).

Gustafson, J., & Edlund, J. (2008). expros: a toolkit for exploratory experimentation with prosody in customized diphone voices. In *Proceedings of PIT 2008*.

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

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

Sundaram, S., & Narayanan, S. (2003). An empirical text transformation method for spontaneous speech synthesizers. In *Proceedings of Interspeech 2003*, Switzerland.

TELIDA: A Package for Manipulation and Visualization of Timed Linguistic Data

Titus von der Malsburg, Timo Baumann, David Schlangen
Department of Linguistics
University of Potsdam, Germany
{malsburg|timo|das}@ling.uni-potsdam.de

Abstract

We present a toolkit for manipulating and visualising time-aligned linguistic data such as dialogue transcripts or language processing data. The package complements existing editing tools by allowing for conversion between their formats, information extraction from the raw files, and by adding sophisticated, and easily extended methods for visualising the dynamics of dialogue processing. To illustrate the versatility of the package, we describe its use in three different projects at our site.

1 Introduction

Manual inspection and visualization of raw data is often an important first step in the analysis of linguistic data, be that transcripts of conversations or records of the performance of processing modules. Dialogue data or speech processing data in general are typically temporally aligned, which poses additional challenges for handling and visualization. A number of tools are available for working with timed data, each with different focus: as a small selection, Praat (Boersma, 2001) and Wavesurfer (Sjölander and Beskow, 2000) excel at acoustic analysis and are helpful for transcription work, Anvil (Kipp, 2001) helps with the analysis of video material, Exmaralda (Schmidt, 2004) offers a suite of specialized tools for discourse analysis.

We developed TELIDA (TimEd LInguistic DAta) to complement the strengths of these tools. TELIDA comprises (a) a suite of Perl modules that offer flexible data structures for storing timed data; tools for converting data in other formats to and from this format; a command-line based interface for querying such data, enabling for example statistical analysis outside of the original creators of transcriptions or annotations; and (b) a lightweight but powerful visualization tool, *TEDview*, that has certain unique features, as will be described in Section 2.3. TELIDA is available for download from `http://www.ling.uni-potsdam.de/~timo/code/telida/`.

2 Overview of TELIDA

2.1 Data Structures

Like the tools mentioned above, we handle timed data as discrete *labels* which span a certain time and contain some data. To give an example, in a word-aligned transcription of a recording, a single word would correspond to one label. Sequences of (non-overlapping) labels are collected into what we call *alignments*. In our example of the word-aligned transcription, all words from one speaker might be collected in one alignment.

This so far is a conceptualization that is common to many tools. In Praat for example, our alignments would be called a *tier*. TELIDA adds a further, novel, abstraction, by treating alignments as belief states that can have a time (namely that of their formation) as well. Concretely, an incremental ASR may hypothesize a certain way of analyzing a stretch of sound at one point, but at a later point might slighlty adapt this analysis; in our conceptualization, this would be two alignments that model the same original data, each with a time stamp. For other applications, timed belief states may contain other information, e.g. new states of parse constructions or dialogue manager information states. We also allow to store several of such *alignment sequences* (= successive belief states) in parallel, to represent n-best lists.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 302–305,
Queen Mary University of London, September 2009.

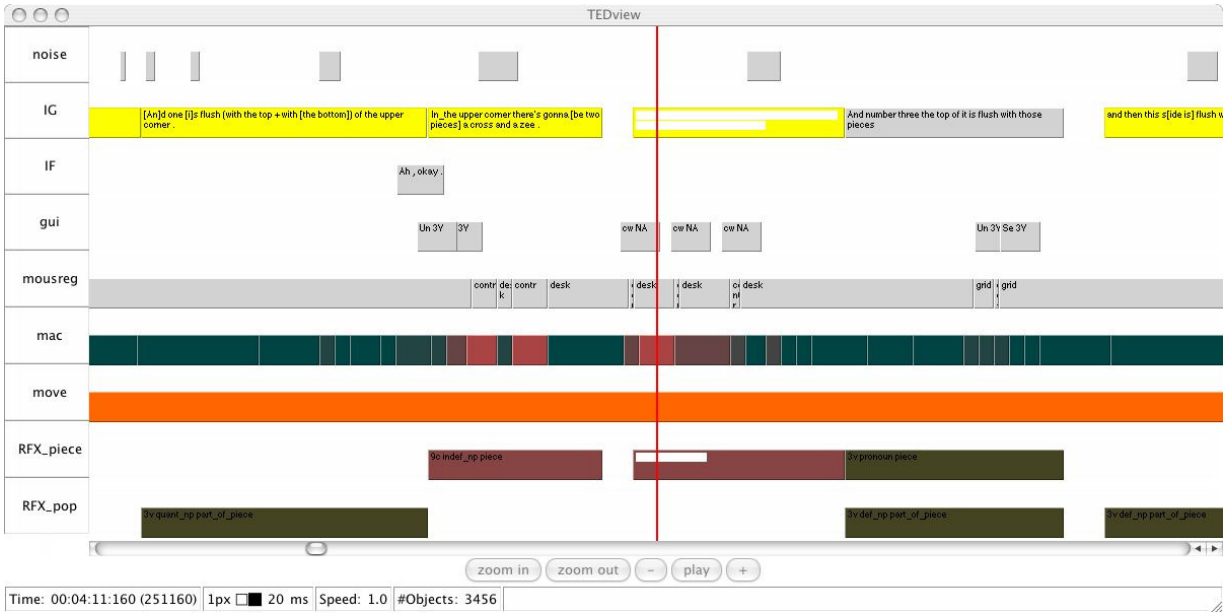


Figure 1: TEDview Showing Annotated Dialogue Data

A *document* finally can consist of collections of such alignments that reference the same timeline, but model different aspects of the base-data. For example, we may want to store information about turns, how they decompose into words, and into phonemes; or, for dual-channel dialogue, have separate alignments for the different speakers.

2.2 Data Manipulation Tools

In order to process timed linguistic data, we implemented a Perl library and command-line tools, *TGtool* and *INtool* for non-incremental and incremental data respectively. They facilitate handling (showing, merging, editing, ...) and processing (search-and-replace, hypothesis filtering, ...) of data and interface to TEDview for interactive visualization.

2.3 TEDview

TEDview is the visualization component of TELIDA. It organizes the different sources of information (i.e., alignments or alignment sequences) in horizontal tracks. Similar as in many of the abovementioned tools, time progresses from left to right in those tracks. The content of tracks consists of events that are displayed as bars if they have a temporal extent or as diamonds otherwise. TEDview uses a player metaphor and therefore has a cursor that marks the current time and a play-mode that can be used to replay recorded sequences of events (in real-time or sped-up / slowed-down). Unlike in other tools, TEDview has a steady cursor (the red line in the Figures) across which events flow, and this cursor can be moved, e.g. to give a configuration where no future events are shown.

Information encapsulated by events is displayed in two different ways:

a) *Labels* are represented as bars, with the label information shown as text. (Figure 1 shows a configuration with only labels.)

b) Events without duration are displayed as diamonds at the appropriate time (all other Figures). Such events can carry a "payload"; depending on its type, different display methods are chosen:

- If the payload is an alignment, it is displayed on the same track, as a sequence of labels.
- In all other cases TEDview determines the data type of the information and selects an appropriate plug-in for displaying it in a separate inspector window. These data types can be syntax trees, probability distributions, etc.

To avoid visual clutter, only the information contained in the diamonds that most recently passed the cursor are displayed. In this way, TEDview can elegantly visualize the dynamics of information state development.

Events can be fed to TEDview either from a file, in a use case where pre-recorded material is replayed for analysis, or online, via a network connection, in use cases where processing components are monitored or profiled in real-time. The format used to encode events and their encapsu-

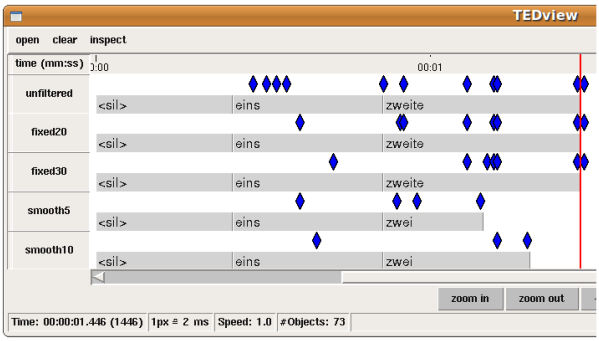


Figure 2: TEDview showing different filtering strategies for incremental ASR: Diamonds correspond to edits of the hypothesis.

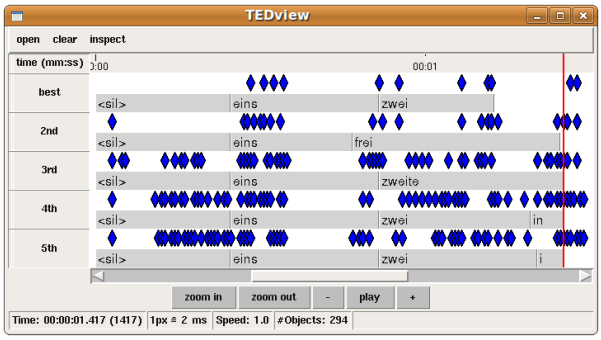


Figure 3: TEDview showing 5-best incremental ASR hypotheses.

lated information is a simple and generic XML format (which the data manipulation tools can create out of other formats, if necessary), i.e. the format does not make any assumptions as to what the events represent. For this reason TEDview can be used to visualize almost any type of discrete temporal data. Intervals can be adorned with display information, for example to encode further information via colouring. Plug-ins for special datatypes can be written in the programming language Python with its powerful library of extension modules; this enabled us to implement an inspector for syntax trees in only 20 lines of code.

3 Use Cases

To illustrate the versatility of the tool, we now describe how we use it in several projects at our site. (Technical manuals can be downloaded from the page listed above.)

3.1 Analysis of Dialogue Data

In the DEAWU project (see e.g. (Schlangen and Fernández, 2007)), we used the package to maintain transcriptions made in Praat and annotations made in MMAX2 (Müller and Strube, 2006), and to visualize these together in a time-aligned view. As Figure 1 shows, we made heavy use of the possibility of encoding information via colour. In the example, there is one track (*mac*, for *mouse activity*) where a numerical value (how much the mouse travels in a certain time frame) is visualized through the colouring of the interval. In other tracks other information is encoded through colour as well. We found this to be of much use in the "getting to know the data" phase of the analysis of our experiment. We have also used the tool and the data in teaching about dialogue structure.

3.2 Analysis of SDS Performance

In another project, we use TELIDA to analyze and visualize the incremental output of several modules of a spoken dialogue system we are currently developing.

In incremental speech recognition, what is considered the best hypothesis frequently changes as more speech comes in. We used TEDview to analyze these changes and to develop filtering methods to reduce the jitter and to reduce edits of the ASR's incremental hypothesis (Baumann et al., 2009a). Figure 2 shows incremental hypotheses and different settings of two filtering strategies.

When evaluating the utility of using n-best ASR hypotheses, we used TEDview to visualize the best hypotheses (Baumann et al., 2009b). An interesting result we got from this analysis is that typically the best hypothesis seems to be more stable than lower-ranked hypotheses, as can be seen in Figure 3.

We also evaluated the behaviour of our incremental reference resolution module, which outputs distributions over possible referents (Schlangen et al., 2009). We implemented a TEDview plug-in to show distributions in bar-charts, as can be seen in Figure 4.

3.3 Analysis of Cognitive Models

In another project, we use TEDview to visualize the output of an ACT-R (Anderson et al., 2004) simulation of human sentence parsing developed by (Patil et al., 2009). This model produces predictions of parsing costs based on working-memory load which in turn are used to predict eye tracking measures in reading. Figure 5 shows an example where the German sentence "*Den Ton gab der Künstler seinem Gehilfen*" (the artist gives the clay to his assistant) is being parsed, taking

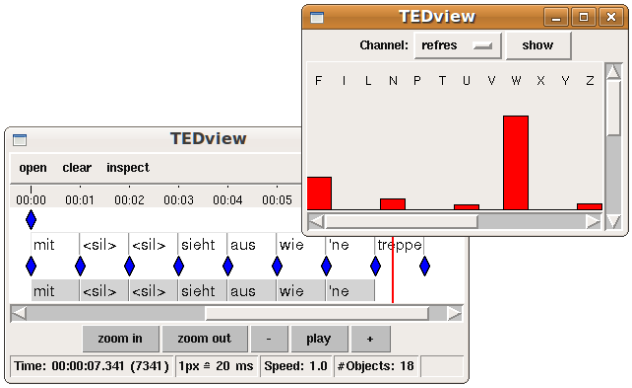


Figure 4: TEDview showing the output of our incremental reference resolution module. Distributions are shown with a bar-chart plug-in.

about 3 seconds of simulated time. The items in the channel labeled "Memory" indicate retrievals of items from memory, the items in the channel labeled "Parse" indicate that the parser produced a new hypothesis, and the inspector window on the right shows the latest of these hypotheses according to cursor time. The grey bars finally in the remaining channels show the activity of the production rules. Such visualizations help to quickly grasp the behaviour of a model, and so greatly aid development and debugging.

4 Conclusions

We presented TELIDA, a package for the manipulation and visualization of temporally aligned (linguistic) data. The package enables convenient handling of dynamic data, especially from incremental processing, but more generally from all kinds of belief update. We believe that it can be of use to anyone who is interested in exploring complex state changes over time, be that in dialogue annotations or in system performance profiles.

Acknowledgments This work was funded by a grant from DFG in the Emmy Noether Programme.

References

J.R. Anderson, D. Bothell, M.D. Byrne, S. Douglass, C. Lebiere, and Y. Qin. 2004. An integrated theory of the mind. *Psychological Review*, 111(4):1036–1060.

Timo Baumann, Michaela Atterer, and David Schlangen. 2009a. Assessing and Improving the Performance of Speech Recognition for Incremental Systems. In *Proceedings of NAACL-HLT 2009*, Boulder, USA.

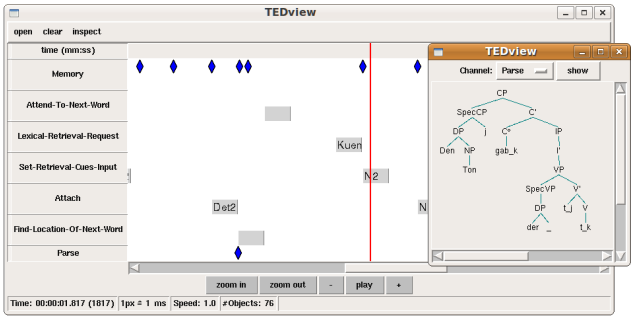


Figure 5: TEDview visualizing the dynamics of an ACT-R simulation, including the current parse-tree.

Timo Baumann, Okko Buß, Michaela Atterer, and David Schlangen. 2009b. Evaluating the Potential Utility of ASR N-Best Lists for Incremental Spoken Dialogue Systems. In *Proceedings of Interspeech 2009*, Brighton, UK.

Paul Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9–10):341–345.

Michael Kipp. 2001. Anvil - a generic annotation tool for multimodal dialogue. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*, pages 1367–1370, Aalborg, Denmark.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang.

Umesh Patil, Marisa Ferrara Boston, John T. Hale, Shravan Vasishth, and Reinhold Kliegl. 2009. The interaction of surprisal and working memory cost during reading. In *Proc. of the CUNY sentence processing conference*, Davis, USA.

David Schlangen and Raquel Fernández. 2007. Speaking through a noisy channel - experiments on inducing clarification behaviour in human-human dialogue. In *Proceedings of Interspeech 2007*, Antwerp, Belgium.

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 *Proc. of SigDial 2009*, London, UK.

Thomas Schmidt. 2004. Transcribing and annotating spoken language with exmaralda. In *Proceedings of the LREC-Workshop on XML based richly annotated corpora, Lisbon 2004*, Paris. ELRA. EN.

K. Sjölander and J. Beskow. 2000. Wavesurfer—an open source speech tool. In *Sixth International Conference on Spoken Language Processing*, Beijing, China. ISCA.

Cascaded Lexicalised Classifiers for Second-Person Reference Resolution

Matthew Purver
Department of Computer Science
Queen Mary University of London
London E1 4NS, UK
mpurver@dcs.qmul.ac.uk

Raquel Fernández
ILLC
University of Amsterdam
1098 XH Amsterdam, Netherlands
raquel.fernandez@uva.nl

Matthew Frampton and **Stanley Peters**
CSLI
Stanford University
Stanford, CA 94305, USA
frampton,peters@csli.stanford.edu

Abstract

This paper examines the resolution of the second person English pronoun *you* in multi-party dialogue. Following previous work, we attempt to classify instances as generic or referential, and in the latter case identify the singular or plural addressee. We show that accuracy and robustness can be improved by use of simple lexical features, capturing the intuition that different uses and addressees are associated with different vocabularies; and we show that there is an advantage to treating referentiality and addressee identification as separate (but connected) problems.

1 Introduction

Resolving second-person references in dialogue is far from trivial. Firstly, there is the *referentiality* problem: while we generally conceive of the word *you*[1] as a deictic addressee-referring pronoun, it is often used in non-referential ways, including as a discourse marker (1) and with a generic sense (2). Secondly, there is the *reference* problem: in addressee-referring cases, we need to know who the addressee is. In two-person dialogue, this is not so difficult; but in multi-party dialogue, the addressee could in principle be any one of the other participants (3), or any group of more than one (4):

(1) It's not just, you know, noises like something hitting.

(2) Often, you need to know specific button sequences to get certain functionalities done.

(3) I think it's good. You've done a good review.

(4) I don't know if you guys have any questions.

[1] We include *your, yours, yourself, yourselves*.

This paper extends previous work (Gupta et al., 2007; Frampton et al., 2009) in attempting to automatically treat both problems: detecting referential uses, and resolving their (addressee) reference. We find that accuracy can be improved by the use of lexical features; we also give the first results for treating both problems simultaneously, and find that there is an advantage to treating them as separate (but connected) problems via cascaded classifiers, rather than as a single joint problem.

2 Related Work

Gupta et al. (2007) examined the referentiality problem, distinguishing generic from referential uses in multi-party dialogue; they found that 47% of uses were generic and achieved a classification accuracy of 75%, using various discourse features and discriminative classifiers (support vector machines and conditional random fields). They attempted the reference-resolution problem, using only discourse (non-visual) features, but accuracy was low (47%).

Addressee identification in general (i.e. independent of the presence of *you*) has been approached in various ways. Traum (2004) gives a rule-based algorithm based on discourse structure; van Turnhout et al. (2005) used facial orientation as well as utterance features; and more recently Jovanovic (2006; 2007) combined discourse and gaze direction features using Bayesian networks, achieving 77% accuracy on a portion of the AMI Meeting Corpus (McCowan et al., 2005) of 4-person dialogues.

In recent work, therefore, Frampton et al. (2009) extended Gupta et al.'s method to include multi-modal features including gaze direction, again using Bayesian networks on the AMI corpus. This gave a small improvement on the ref-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 306–309,
Queen Mary University of London, September 2009.

erentiality problem (achieving 79% accuracy), and a large improvement on the reference-resolution task (77% accuracy distinguishing singular uses from plural, and 80% resolving singular individual addressee reference).

However, they treated the two tasks in isolation, and also broke the addressee-reference problem into two separate sub-tasks (singular vs. plural reference, and singular addressee reference). A full computational *you*-resolution module would need to treat all tasks (either simultaneously as one joint classification problem, or as a cascaded sequence) – with inaccuracy at one task necessarily affecting performance at another – and we examine this here. In addition, we examine the effect of lexical features, following a similar insight to Katzenmaier et al. (2004); they used language modelling to help distinguish between user- and robot-directed utterances, as people use different language for the two – we expect that the same is true for human participants.

3 Method

We used Frampton et al. (2009)'s AMI corpus data: 948 *"you"*-containing utterances, manually annotated for referentiality and accompanied by the AMI corpus' original addressee annotation. The very small number of two-person addressee cases were joined with the three-person (i.e. all non-speaker) cases to form a single "plural" class. 49% of cases are generic; 32% of referential cases are plural, and the rest are approximately evenly distributed between the singular participants. While Frampton et al. (2009) labelled singular reference by physical location relative to the speaker (giving a 3-way classification problem), our lexical features are more suited to detecting actual participant identity – we therefore recast the singular reference task as a 4-way classification problem and re-calculate their performance figures (giving very similar accuracies).

Discourse Features We use Frampton et al. (2009)'s discourse features. These include simple durational and lexical/phrasal features (including mention of participant names); AMI dialogue act features; and features expressing the similarity between the current utterance and previous/following utterances by other participants. As dialogue act features are notoriously hard to tag automatically, and "forward-looking" information about following utterances may be unavailable in an on-line system, we examine the effect of leaving these out below.

Visual Features Again we used Frampton et al. (2009)'s features, extracted from the AMI corpus manual focus-of-attention annotations which track head orientiation and eye gaze. Features include the target of gaze (any participant or the meeting whiteboard/projector screen) during each utterance, and information about mutual gaze between participants. These features may also not always be available (meeting rooms may not always have cameras), so we investigate the effect of their absence below.

Lexical Features The AMI Corpus simulates a set of scenario-driven business meetings, with participants performing a design task (the design of a remote control). Participants are given specific roles to play, for example that of project manager, designer or marketing expert. It therefore seems possible that utterances directed towards particular individuals will involve the use of different vocabularies reflecting their expertise. Different words or phrases may also be associated with generic and referential discussion, and extracting these automatically may give benefits over attempting to capture them using manually-defined features. To exploit this, we therefore added the use of lexical features: one feature for each distinct word or n-gram seen more than once in the corpus. Although such features may be corpus- or domain-specific, they are easy to extract given a transcript.

4 Results and Discussion

4.1 Individual Tasks

We first examine the effect of lexical features on the individual tasks, using 10-way cross-validation and comparing performance with Frampton et al. (2009). Table 1 shows the results for the referentiality task in terms of overall accuracy and per-class F1-scores; 'MC Baseline' is the majority-class baseline; results labelled 'EACL' are Frampton et al. (2009)'s figures, and are presented for all features and for reduced feature sets which might be more realistic in various situations: '-V' removes visual features; '-VFD' removes visual features, forward-looking discourse features and dialogue-act tag features.

As can be seen, adding lexical features ('+words' adds single word features, '+3grams' adds n-gram features of lengths 1-3) improves the

Features	Acc	F_{gen}	F_{ref}
MC Baseline	50.9	0	67.4
EACL	**79.0**	80.2	77.7
EACL -VFD	73.7	74.1	73.2
+words	85.3	85.7	84.9
+3grams	**87.5**	87.4	87.5
+3grams -VFD	87.2	86.9	87.6
3grams only	**85.9**	85.2	86.4

Table 1: Generic *vs.* referential uses

Features	Acc	F_{sing}	F_{plur}
MC Baseline	67.9	80.9	0
EACL	**77.1**	83.3	63.2
EACL -VFD	71.4	81.5	37.1
+words	83.1	87.8	72.5
+3grams	**85.9**	90.0	76.6
+3grams -VFD	**87.1**	91.0	77.6
3grams only	**86.9**	90.8	77.0

Table 2: Singular *vs.* plural reference.

performance significantly – accuracy is improved by 8.5% absolute above the best EACL results, which is a 40% reduction in error. Robustness to removal of potentially problematic features is also improved: removing all visual, forward-looking and dialogue act features makes little difference. In fact, using *only* lexical n-gram features, while reducing accuracy by 2.6%, still performs better than the best EACL classifier.

Table 2 shows the equivalent results for the singular-plural reference distinction task; in this experiment, we used a correlation-based feature selection method, following Frampton et al. (2009). Again, performance is improved, this time giving a 8.8% absolute accuracy improvement, or 38% error reduction; robustness to removing visual and dialogue act features is also very good, even improving performance.

For the individual reference task (again using feature selection), we give a further 'NS baseline' of taking the next speaker; note that this performs rather well, but requires forward-looking information so should not be compared to '-F' results. Results are again improved (Table 3), but the improvement is smaller: a 1.4% absolute accuracy improvement (7% error reduction); we conclude from this that visual information is most important for this part of the task. Robustness to feature unavailability still shows some improvement: excluding all visual, forward-looking and dialogue-act features has less effect than on the EACL system (60.7% vs. 56.6% accuracy), and a system using only n-grams and the next speaker identity gives a respectable 74.5%.

Features	Acc	F_{P1}	F_{P2}	F_{P3}	F_{P4}
MC baseline	30.7	0	0	0	47.0
NS baseline	70.7	71.6	71.1	72.7	68.2
EACL	**80.3**	82.8	79.7	75.9	81.4
EACL -V	73.8	79.2	70.7	74.1	71.4
EACL -VFD	56.6	58.9	55.5	64.0	47.3
+words	81.4	83.9	79.7	79.3	81.8
+3grams	**81.7**	83.9	80.3	79.3	82.5
+3grams -V	74.8	81.3	71.7	75.2	71.4
+3grams -VFD	60.7	66.3	55.9	66.2	53.0
3grams only	60.7	63.1	58.1	52.9	63.4
3grams +NS	**74.5**	76.7	73.8	75.0	72.7

Table 3: Singular addressee detection.

Feature Analysis We examined the contribution of particular lexical features using Information Gain methods. For the referentiality task, we found that generic uses of *you* were more likely to appear in utterances containing words related to the main meeting topic, such as *button*, *channel*, or *volume* (properties of the to-be-designed remote control). In contrast, words related to meeting management, such as *presentation*, *email*, *project* and *meeting* itself, were predictive of referential uses. The presence of first person pronouns and discourse and politeness markers such as *okay*, *please* and *thank you* was also indicative of referentiality, as were n-grams capturing interrogative structures (e.g. *do you*).

For the plural/singular distinction, we found that the plural first person pronoun *we* correlated with plural references of *you*. Other predictive n-grams for this task were *you mean* and *you know*, which were indicative of singular and plural references, respectively. Finally, for the individual reference task, useful lexical features included participant names, and items related to their roles. For instance, the n-grams *sales*, *to sell* and *make money* correlated with utterances addressed to the "marketing expert", while utterances containing *speech recognition* and *technical* were addressed to the "industrial designer".

Discussion The best F-score of the three subtasks is for the generic/referential distinction; the

Features	Acc	F_{gen}	F_{plur}	F_{P1}	F_{P2}	F_{P3}	F_{P4}
MC baseline	49.1	65.9	0	0	0	0	0
EACL	**58.3**	73.3	24.3	57.6	57.0	36.0	51.1
+3grams	60.9	74.8	42.0	57.7	52.2	35.6	50.2
3grams only	**67.5**	84.8	61.6	39.1	39.3	30.6	38.6
Cascade +3grams	**78.1**	87.4	59.1	64.1	76.4	75.0	82.6

Table 4: Combined task: generic *vs.* plural *vs.* singular addressee.

worst is for the detection of plural reference (F_{plur} in Table 2). This is not surprising: humans find the former task easy to annotate – Gupta et al. (2007) report good inter-annotator agreement ($\kappa = 0.84$) – but the latter hard. In their analysis of the AMI addressee annotations, Reidsma et al. (2008) observe that most confusions amongst annotators are between the group-addressing label and the labels for individuals; whereas if annotators agree that an utterance is addressed to an individual, they also reach high agreement on that addressee's identity.

4.2 Combined Task

We next combined the individual tasks into one combined task; for each *you* instance, a 6-way classification as generic, group-referring or referring to one of the 4 participants. This was attempted both as a single classification exercise using a single Bayesian network; and as a cascaded pipeline of the three individual tasks; see Table 4. Both used correlation-based feature selection.

For the single joint classifier, n-grams again improve performance over the EACL features. Using *only* n-grams gives a significant improvement, perhaps due to the reduction in the size of the feature space on this larger problem. Accuracy is reasonable (67.5%), but while F-scores are good for the generic class (above 80%), others are low.

However, use of three cascaded classifiers improves performance to 78% and gives large per-class F-score improvements, exploiting the higher accuracy of the first two stages (generic/referential, singular/plural), and the fact that different features are good for different tasks.

5 Conclusions

We have shown that the use of simple lexical features can improve performance and robustness for all aspects of second-person pronoun resolution: referentiality detection and reference identification. An overall 6-way classifier is feasible, and cascading individual classifiers can help. Future plans include testing on ASR transcripts, and investigating different classification techniques for the joint task.

References

M. Frampton, R. Fernández, P. Ehlen, M. Christoudias, T. Darrell, and S. Peters. 2009. Who is "you"? combining linguistic and gaze features to resolve second-person references in dialogue. In *Proceedings of the 12th Conference of the EACL*.

S. Gupta, J. Niekrasz, M. Purver, and D. Jurafsky. 2007. Resolving "you" in multi-party dialog. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*.

N. Jovanovic, R. op den Akker, and A. Nijholt. 2006. Addressee identification in face-to-face meetings. In *Proceedings of the 11th Conference of the EACL*.

N. Jovanovic. 2007. *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings*. Ph.D. thesis, University of Twente, The Netherlands.

M. Katzenmaier, R. Stiefelhagen, and T. Schultz. 2004. Identifying the addressee in human-human-robot interactions based on head pose and speech. In *Proceedings of the 6th International Conference on Multimodal Interfaces*.

I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner. 2005. The AMI Meeting Corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*.

D. Reidsma, D. Heylen, and R. op den Akker. 2008. On the contextual analysis of agreement scores. In *Proceedings of the LREC Workshop on Multimodal Corpora*.

D. Traum. 2004. Issues in multi-party dialogues. In F. Dignum, editor, *Advances in Agent Communication*, pages 201–211. Springer-Verlag.

K. van Turnhout, J. Terken, I. Bakx, and B. Eggen. 2005. Identifying the intended addressee in mixed human-humand and human-computer interaction from non-verbal features. In *Proceedings of ICMI*.

Attention and Interaction Control in a Human-Human-Computer Dialogue Setting

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

Joakim Gustafson
Dept. of Speech Music and Hearing
KTH, Stockholm, Sweden
jocke@speech.kth.se

Abstract

This paper presents a simple, yet effective model for managing attention and interaction control in multimodal spoken dialogue systems. The model allows the user to switch attention between the system and other humans, and the system to stop and resume speaking. An evaluation in a tutoring setting shows that the user's attention can be effectively monitored using head pose tracking, and that this is a more reliable method than using push-to-talk.

1 Introduction

Most spoken dialogue systems are based on the assumption that there is a clear beginning and ending of the dialogue, during which the user pays attention to the system constantly. However, as the use of dialogue systems is extended to settings where several humans are involved, or where the user needs to attend to other things during the dialogue, this assumption is obviously too simplistic (Bohus & Horvitz, 2009). When it comes to interaction, a strict turn-taking protocol is often assumed, where user and system wait for their turn and deliver their contributions in whole utterance-sized chunks. If system utterances are interrupted, they are treated as either fully delivered or basically unsaid.

This paper presents a simple, yet effective model for managing attention and interaction control in multimodal (face-to-face) spoken dialogue systems, which avoids these simplifying assumptions. We also present an evaluation in a tutoring setting where we explore the use of head tracking for monitoring user attention, and compare it with a more traditional method: push-to-talk.

2 Monitoring user attention

In multi-party dialogue settings, gaze has been identified as an effective cue to help disambiguate the addressee of a spoken utterance (Vertegaal et al., 2001). When it comes to human-machine interaction, Maglio et al. (2000) showed that users tend to look at speech-controlled devices when talking to them, even if they do not have the manifestation of an embodied agent. Bakx et al. (2003) investigated the use of head pose for identifying the addressee in a multi-party interaction between two humans and an information kiosk. The results indicate that head pose should be combined with acoustic and linguistic features such as utterances length. Facial orientation in combination with speech-related features was investigated by Katzenmaier et al. (2004) in a human-human-robot interaction, confirming that a combination of cues was most effective. A common finding in these studies is that if a user does not look at the system while talking he is most likely not addressing it. However, when the user looks at the system while speaking, there is a considerable probability that she is actually addressing a bystander.

3 The MonAMI Reminder

This study is part of the 6th framework IP project MonAMI[1]. The goal of the MonAMI project is to develop and evaluate services for elderly and disabled people. Based on interviews with potential users in the target group, we have developed the MonAMI Reminder, a multimodal spoken dialogue system which can assist elderly and disabled people in organising and initiating their daily activities (Beskow et al., 2009). The dialogue system uses Google Calendar as a backbone to answer questions about events. However,

[1] http://www.monami.info/

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 310–313,
Queen Mary University of London, September 2009.

it can also take the initiative and give reminders to the user.

The MonAMI Reminder is based on the HIGGINS platform (Skantze, 2007). The architecture is shown in Figure 1. A microphone and a camera are used for system input (speech recognition and head tracking), and a speaker and a display are used for system output (an animated talking head). This is pretty much a standard dialogue system architecture, with some exceptions. Dialogue management is split into a Discourse Modeller and an Action Manager, which consults the discourse model and decides what to do next. There is also an Attention and Interaction Controller (AIC), which will be discussed next.

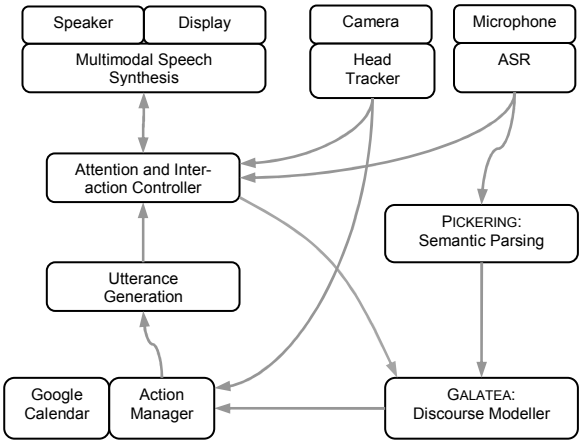


Figure 1. The system architecture in the MonAMI Reminder.

4 Attention and interaction model

The purpose of the AIC is to act as a low level monitor and controller of the system's speaking and attentional behaviour. The AIC uses a state-based model to track the attentional and interactional state of the user and the system, shown in Figure 2. The states shown in the boxes can be regarded as the combined state of the system (columns) and the user (rows)[2]. Depending on the combined state, events from input and output components will have different effects. As can be seen in the figure, some combination of states cannot be realised, such as the system and user speaking at the same time (if the user speaks while the system is speaking, it will automatically change to the state INTERRUPTED). Of course, the user might speak while the system is speaking without the system detecting this, but the model should be regarded from the system's perspective, not from an observer.

The user's attention is monitored using a camera and an off-the-shelf head tracking software. As the user starts to look at the system, the state changes from NONATTENTIVE to ATTENTIVE. When the user starts to speak, a *UserStartSpeak* event from the ASR will trigger a change to the LISTENING state. The Action Manager might then trigger a *SystemResponse* event (together with what should be said), causing a change into the SPEAKING state. Now, if the user would look away while the system is speaking, the system would enter the HOLDING state – the system would pause and then resume when the user looks back. If the user starts to speak while the system is speaking, the controller will enter the INTERRUPTED state. The Action Manager might then either decide to answer the new request, resume speaking (e.g., if there was just a back-channel or the confidence was too low), or abort speaking (e.g., if the user told the system to shut up).

There is also a CALLING state, in which the system might try to grab the user's attention. This is very important for the current application when the system needs to remind the user about something.

4.1 Incremental multimodal speech synthesis

The speech synthesiser used must be capable of reporting the timestamp of each word in the synthesised string. These are two reasons for this. First, it must be possible to resume speaking after returning from the states INTERRUPTED and HOLDING. Second, the AIC is responsible for reporting what has actually been said by the system back to the Discourse Modeller for continuous self monitoring (there is a direct feedback loop as can be seen in Figure 1). This way, the Discourse Modeller may relate what the system says to what the user says on a high resolution time scale (which is necessary for handling phenomena such as backchannels, as discussed in Skantze & Schlangen, 2009).

Currently, the system may pause and resume speaking at any word boundary and there is no specific prosodic modelling of these events. The synthesis of interrupted speech is something that we will need to improve.

[2] This is somewhat similar to the "engagement state" used in Bohus & Horvitz (2009).

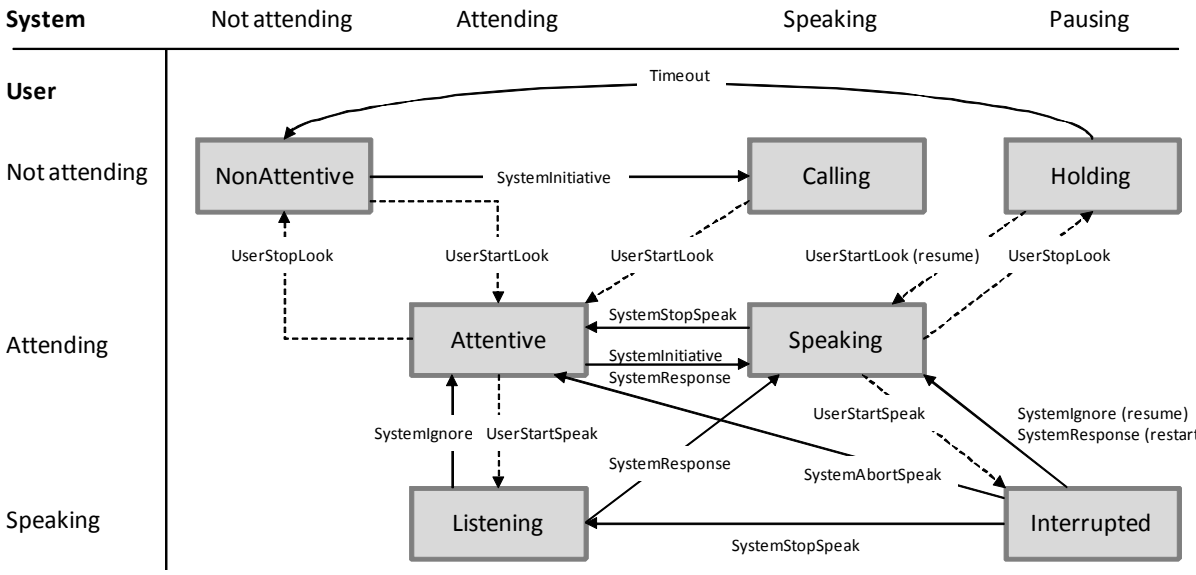


Figure 2. The attention and interaction model. Dashed lines indicate events coming from input modules. Solid lines indicate events from output modules. Note that some events and transitions are not shown in the figure.

An animated talking head is shown on a display, synchronised with the synthesised speech (Beskow, 2003). The head is making small continuous movements (recorded from real human head movements), giving it a more life-like appearance. The head pose and facial gestures are triggered by the different states and events in the AIC, as can be seen in Figure 3. Thus, when the user approaches the system and starts to look at it, the system will look up, giving a clear signal that it is now attending to the user and ready to listen.

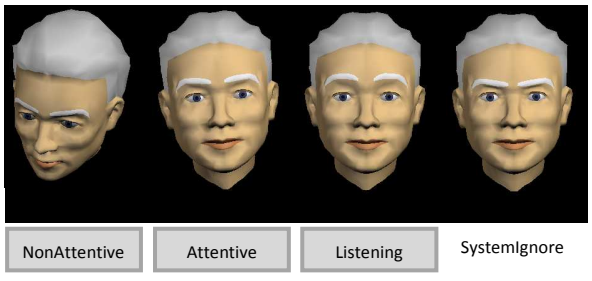


Figure 3. Examples of facial animations triggered by the different states and events shown in Figure 2.

5 Evaluation

In the evaluation, we not only wanted to check whether the AIC model worked, but also to understand whether user attention could be effectively modelled using head tracking. Similarly to Oh et al. (2002), we wanted to compare "look-to-talk" with "push-to-talk". To do this, we used a human-human-computer dialogue setting, where a tutor was explaining the system to a subject (shown in Figure 4). Thus, the subject needed to frequently switch between speaking to the tutor and the system. A second version of the system was also implemented where the head tracker was not used, but where the subject instead pushed a button to switch between the attentional states (a sort-of push-to-talk). The tutor first explained both versions of the system to the subject and let her try both. The tutor gave the subjects hints on how to express themselves, but avoided to remind them about how to control the attention of the system, as this was what we wanted to test. After the introduction, the tutor gave the subject a task where both of them were supposed to find a suitable slot in their calendars to plan a dinner or lunch together. The tutor used a paper calendar, while the subject used the MonAMI Reminder. At the end of the experiment, the tutor interviewed the subject about her experience of using the system. 7 subjects (4 women and 3 men) were used in the evaluation, 3 lab members and 4 elderly persons in the target group (recruited by the Swedish Handicap Institute).

There was no clear consensus on which version of the system was the best. Most subjects liked the head tracking version better when it worked but were frustrated when the head tracker occasionally failed. They reported that a combined version would perhaps be the best, where head pose could be the main method for handling attention, but where a button or a verbal call for attention could be used as a fall-back.

When looking at the interaction from an objective point of view, however, the head tracking

Figure 4. The human-human-computer dialogue setting used in the evaluation. The tutor is sitting on the left side and the subject on the right side

version was clearly more successful in terms of number of misdirected utterances. When talking to the system, the subjects always looked at the system in the head tracking condition and never forgot to activate it in the push-to-talk condition. However, on average 24.8% of all utterances addressed to the tutor in the push-to-talk condition were picked up by the system, since the user had forgotten to deactivate it. The number of utterances addressed to the tutor while looking at the system in the head tracking condition was significantly lower, only 5.1% on average (paired t-test; $p<0.05$).

These findings partly contradict findings from previous studies, where head pose has not been that successful as a sole indicator when the user is looking at the system, as discussed in section 2 above. One explanation for this might be that the subjects were explicitly instructed about how the system worked. Another explanation is the clear feedback (and entrainment) that the agent's head pose provided.

Two of the elderly subjects had no previous computer experience. During pre-interviews they reported that they were intimidated by computers, and that they got nervous just thinking about having to operate them. However, after only a short tutorial session with the spoken interface, they were able to navigate through a computerized calendar in order to find two empty slots. We think that having a human tutor that guides the user through their first interactions with this kind of system is very important. One of the tutor's tasks is to explain why the system fails to understand out-of-vocabulary expressions. By doing this, the users' trust in the system is increased and they become less confused and frustrated. We are confident that monitoring and modelling the user's attention is a key component of spoken dialogue systems that are to be used in tutoring settings.

Acknowledgements

This research is supported by MonAMI, an Integrated Project under the European Commission's 6th Framework Program (IP-035147), and the Swedish research council project GENDIAL (VR #2007-6431).

References

Bakx, I., van Turnhout, K., & Terken, J. (2003). Facial orientation during multi-party interaction with information kiosks. In *Proceedings of the Interact 2003*.

Beskow, J., Edlund, J., Granström, B., Gustafson, J., Skantze, G., & Tobiasson, H. (2009). The MonAMI Reminder: a spoken dialogue system for face-to-face interaction. In *Proceedings of Interspeech 2009*.

Beskow, J. (2003). *Talking heads - Models and applications for multimodal speech synthesis*. Doctoral dissertation, KTH, Department of Speech, Music and Hearing, Stockholm, Sweden.

Bohus, D., & Horvitz, E. (2009). Open-World Dialog: Challenges, Directions, and Prototype. In *Proceedings of IJCAI'2009 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*. Pasadena, CA.

Katzenmaier, M., Stiefelhagen, R., Schultz, T., Rogina, I., & Waibel, A. (2004). Identifying the Addressee in Human-Human-Robot Interactions based on Head Pose and Speech. In *Proceedings of ICMI 2004*.

Maglio, P. P., Matlock, T., Campbell, C. S., Zhai, S., & Smith, B. A. (2000). Gaze and speech in attentive user interfaces. In *Proceedings of ICMI 2000*.

Oh, A., Fox, H., Van Kleek, M., Adler, A., Gajos, K., Morency, L-P., & Darrell, T. (2002). Evaluating Look-to-Talk: A Gaze-Aware Interface in a Collaborative Environment. In *Proceedings of CHI 2002*.

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

Skantze, G. (2007). *Error Handling in Spoken Dialogue Systems – Managing Uncertainty, Grounding and Miscommunication*. Doctoral dissertation, KTH, Department of Speech, Music and Hearing, Stockholm, Sweden.

Vertegaal, R., Slagter, R., van der Veer, G., & Nijholt, A. (2001). Eye gaze patterns in conversations: there is more to conversational agents than meets the eyes. In *Proceedings of ACM Conf. on Human Factors in Computing Systems*.

Ranking Help Message Candidates Based on Robust Grammar Verification Results and Utterance History in Spoken Dialogue Systems

Kazunori Komatani Satoshi Ikeda Yuichiro Fukubayashi
Tetsuya Ogata Hiroshi G. Okuno
Graduate School of Informatics
Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto 606-8501, Japan
{komatani,sikeda,fukubaya,ogata,okuno}@kuis.kyoto-u.ac.jp

Abstract

We address an issue of out-of-grammar (OOG) utterances in spoken dialogue systems by generating help messages for novice users. Help generation for OOG utterances is a challenging problem because language understanding (LU) results based on automatic speech recognition (ASR) results for such utterances are always erroneous as important words are often misrecognized or missed from such utterances. We first develop grammar verification for OOG utterances on the basis of a Weighted Finite-State Transducer (WFST). It robustly identifies a grammar rule that a user intends to utter, even when some important words are missed from the ASR result. We then adopt a ranking algorithm, RankBoost, whose features include the grammar verification results and the utterance history representing the user's experience.

1 Introduction

Studies on spoken dialogue systems have recently proceeded from in-laboratory systems to ones deployed to the open public (Raux et al., 2006; Komatani et al., 2007; Nisimura et al., 2005). Accordingly, opportunities are increasing as general citizens use the systems. This situation means that novice users directly access the systems with no instruction, which is quite different from in-laboratory experiments where some instructions can be given. In such cases, users often experience situations where their utterances are not correctly recognized. This is because of a gap between the actual system and a user's mental model, that is, a user's expectation of the system. Actually, a user's utterance often cannot be interpreted by the system because of the system's limited grammar for language understanding (LU). We call such an unacceptable utterance an "out-of-grammar (OOG) utterance." When users' utterances are OOG, they cannot change their utterances into acceptable ones unless they are informed what expressions are acceptable by the system.

We aim to manage the problem of OOG utterances by providing help messages showing an example of acceptable language expressions when a user utterance is not acceptable. We prepare help messages corresponding to each grammar rule the system has. We therefore assume that appropriate help messages can be provided if a user's intention, i.e., a grammar rule the user originally intends to use by his utterance, is correctly estimated.

Issues for generating such help messages include:

1. Estimating a grammar rule corresponding to user intention even from OOG utterances, and
2. Complementing missing information in a single utterance.

The first issue focuses on the fact that automatic speech recognition (ASR) results, used as main input data, are erroneous for OOG utterances. Estimating a grammar rule that the user intends to use becomes accordingly difficult especially when content words, which correspond to database entries such as place names and their attributes, are not correctly recognized. That is, any type of ASR error in any position should be taken into consideration in ASR results of OOG utterances. On the

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 314–321,
Queen Mary University of London, September 2009.

other hand, the second issue focuses on the fact that an ASR result for an OOG utterance does not necessarily contain sufficient information to estimate the user intention. This is because of ASR errors or that users may omit some elements from their utterances because they are in context.

We develop a grammar verification method based on Weighted Finite-State Transducer (WFST) as a solution to the first issue. The grammar verification method robustly estimates which a grammar rule is intended to use by a user's utterance. The WFST is automatically generated to represent an ASR result in which any possibility of error is taken into consideration. We furthermore adopt a boosting algorithm, RankBoost (Freund et al., 2003), to put help messages in order of probability to address the second issue. Because it is difficult even for human annotators to uniquely determine which help message should be provided for each case, we adopt an algorithm that can be used for training on several data examples that have a certain order of priority. We also incorporate features representing the user's utterance history for preventing message repetition.

2 Related Work

Various studies have been done on generating help messages in spoken dialogue systems. Gorrell et al. (2002) trained a decision tree to classify causes of errors for OOG utterances. Hockey et al. (2003) also classified OOG utterances into the three categories of endpointing errors, unknown vocabulary, and subcategorization mistakes, by comparing two kinds of ASR results. This was called Targeted Help and provided a user with immediate feedback tailored to what the user said. Lee et al. (2007) also addressed error recovery by generating help messages in an example-based dialog modeling framework. These studies, however, determined what help messages should be provided mainly on the basis of literal ASR results. Therefore, help messages would be degraded by ASR results in which a lot of information was missing, especially for OOG utterances. The same help messages would be repeated when the same ASR results were obtained.

An example dialogue enabled by our method, especially the part of the method described in Section 4, is shown in Figure 1. Here, user utterances are transcriptions, and utterance numbers

---

U1: Tell me your recommending sites.
Underlined parts are not in-vocabulary and no valid LU result is obtained. The estimated grammar is [Obtaining info on a site] although the most appropriate help message is that corresponding to [Searching tourist sites].

S1: I did not understand. You can say "Tell me the address of Kiyomizu Temple" for example, if getting information on a site.
The help message corresponding to [Obtaining info on a site] is provided.

U2: Tell me your recommending sites.
The user repeats the same utterance probably because the help message (S1) was not helpful. The estimated grammar is [Obtaining info on a site] again.

S2: I did not understand. You can say "Search shrines or museums" for example, if searching tourist sites.
Another help message corresponding to [Searching tourist sites] is provided after ranking candidates by also using the user's utterance history.

[] denotes grammar rules.

Figure 1: Example dialogue enabled by our method

start with "S" and "U" denote system and user utterances, respectively. In this example, ASR results for the user utterances (U1 and U2) do not contain sufficient information because the utterances are short and contain out-of-vocabulary words. These two results are similar, and accordingly, the help message after U2 provided by methods like Targeted Help (Gorrell et al., 2002; Hockey et al., 2003) is the same as Utterance S1 because they are only based on ASR results. Our method can provide different help messages as Utterance S2 after ranking candidates by considering the utterance history and grammar verification results. Because the candidates are arranged in the order of probability, the most appropriate help message can be provided in fewer attempts.

This ranking method for help message candidates is also useful in multimodal interfaces with speech input. Help messages are necessary when ASR is used as its input modality, and such messages were actually implemented in City Browser (Gruenstein and Seneff, 2007), for example. This system lists template-based help messages on the screen by using ASR results and internal states of the system. The order of help messages is important, especially in portable devices with a small screen, on which the number of help messages dis-

played at one time is limited, as Hartmann and Schreiber (2008) pointed out. Even in cases where sufficiently large screens are available, too many help messages without any order will distract the user's attention and thus spoil its usability.

3 Grammar Verification based on WFST

We estimate a user's intention even from OOG utterances as a grammar rule that the user intends to use by his utterance. We call this estimation grammar verification. This process is applied to ASR outputs based on a statistical language model (LM) in this paper. We use two transducers: a finite-state transducer (FST) representing the task grammar, and weighted FST (WFST) representing an ASR result and its confidence score. Hereafter, we denote these two as "grammar FST" and "input WFST" and depict examples in Figure 2.

A strong point of our method is that it takes all three types of ASR error into consideration. The input WFST is designed to represent all cases where any word in an ASR result is an inserted or substituted error, or any word is deleted. Its weight is designed to reflect confidence scores of ASR results. By composing this WFST and the grammar FST, we can obtain all possible sequences and their accumulated weights when arbitrary sequences represented by the input WFST are input into the grammar FST. The optimal results having the maximum accumulated weight consist of the LU result and the grammar rule that is the nearest to the ASR result. The result can be obtained even when any element in it is misrecognized or absent from the ASR result.

An LU result is a set of concepts that consist of slots and their values corresponding to database entries the system handles. For example, an LU result "month=2, day=22" consists of two concepts, such as the value of slot month is 2, and the value of slot day is 22.

3.1 Design of input WFST and grammar FST

In input WFSTs and grammar FSTs, each arc representing state transitions has a label in the form of "a:b/c" denoting its input symbol, output symbol, and weight, in this order. Input symbol ε means a state transition without any input symbol, that is, an epsilon transition. Output symbol ε means no output in the state transition. For example, a state transition "please:ε/1.0" is executed when an input symbol is "please," no output symbol is generated, and 1.0 is added to the accumulated weight. Weights are omitted in the grammar FST because no weight is given in it.

An input WFST is automatically constructed from an ASR result. Sequential state transitions are assigned to each word in the ASR result, and each of them is paralleled by filler transitions, as shown in Figure 2 where the ASR result was "Every Monday please" for example. Filler transitions such as INS, DEL, and SUB are assigned to each state for representing every kind of error such as insertion, deletion, and substitution errors. All input symbols in the input WFST are ε, by which the WFST represents all possible sequences containing arbitrary errors. For example, the input WFST in Figure 2 represents all possible sequences such as "Every Monday please," "Every Monday *F*," "*F* Monday *F*," and so on. Here, every word can be replaced by the symbol *F* that represents an insertion or substitution error. Moreover, the error symbol DEL can be inserted into its output symbol sequence at any position, which corresponds to deletion errors in ASR results. Each weight per state transition is summed up and then the optimal result is determined. The weights will be explained in Section 3.2.

A grammar FST is generated from a task grammar, which is written by a system developer for each task. It determines whether an input sequence conforms to the task grammar. We also assign filler transitions to each state for handling each type of error of ASR results considered in the input WFST. A filler transition, either of INS, DEL, or SUB, is added to each state in the FST except for states within keyphrases, which are explicitly indicated by a system developer. In the example shown in Figure 2, "SUB $ Monday date-repeat=Mon please" is output for an input sequence "SUB Monday please". Here, date-repeat=Mon denotes an LU result, and $ is a symbol for marking words corresponding to a concept.

3.2 Weights assigned to input WFST

We defined two kinds of weights:

1. Rewards for accepted words (w_{acc}), and
2. Penalties for each kind of error (w_{sub}, w_{del}, w_{ins}).

An accumulated weight for a single utterance is defined as the sum of these weights as shown be-

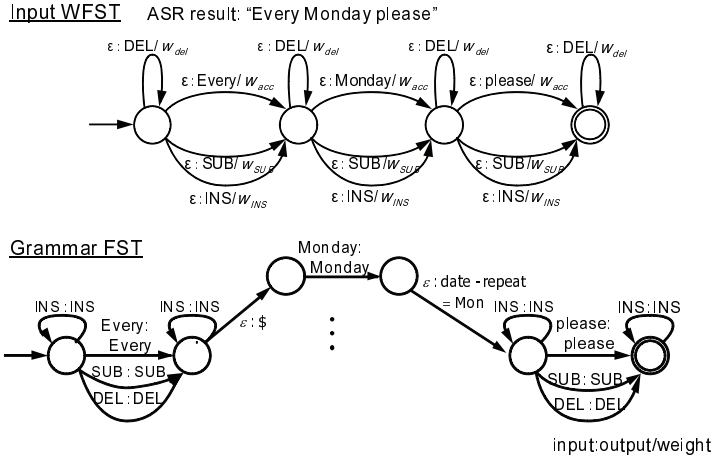


Figure 2: Example of input WFST and grammar FST

low.

$$w = \sum_{E_{\text{accepted}}} w_{acc} + \sum_{E_{\text{error}}} (w_{sub} + w_{del} + w_{ins})$$

Here, E_{accepted} denotes a set of accepted words corresponding to elements of each grammar rule, and E_{error} denotes a set of words that are not accepted and that have either error symbol. Note that the weights are not given beforehand but are calculated and given to the input WFST in runtime according to each ASR result.

A weight for an accepted word e_{asr} is defined by using its confidence score $CM(e_{asr})$ (Lee et al., 2004) and its word length. A word length in mora is denoted as $l(\cdot)$, which is normalized by that of the longest word in the vocabulary.

$$w_{acc} = CM(e_{asr})l(e_{asr})$$

This weight w_{acc} gives preference to sequences containing longer words with higher confidence scores.

Weights for each type of error have negative values because they are penalties:

$$w_{sub} = -\{CM(e_{asr})l(e_{asr}) + l(e_{gram})\}/2$$

$$w_{del} = -\{\overline{l(e)} + l(e_{gram})\}/2$$

$$w_{ins} = -\{CM(e_{asr})l(e_{asr}) + \overline{l(e)}\}/2$$

where $\overline{l(e)}$ is the average word length in the vocabulary and e_{gram} is a grammar element i.e., either a word or a class. A deletion error is a case when a grammar element does not correspond to any word in the ASR result. A substitution error is a case when an element is replaced by another word in the ASR result. An insertion error is a case when no grammar element corresponds to the ASR result. Every weight is defined as an average of a word length of a grammar element and the corresponding one in the ASR result multiplied by its confidence score. When correspondences cannot be defined in insertion and deletion errors, $\overline{l(e)}$ is used instead. In the case when e_{gram} is a class in the grammar, the average word length in that class is used as $l(e_{gram})$.

3.3 Example of calculating the weights

We show how a weight is calculated by using the example in Figure 3. In this example, the user utterance was "Tell me a liaison of Koetsu-ji (a temple name)." The word "liaison" was not in the system vocabulary. The ASR result accordingly contained errors for that part; the result was "Tell me all Sakyo-ward Koetsu-ji."

Weights are calculated for each grammar rule the system has. This example shows calculations for two grammar rules: [get_info] accepting "Tell me ⟨item name⟩ of ⟨temple name⟩," and [search_ward] accepting "Tell me ⟨facility name⟩ of ⟨ward name⟩." Here, [] and ⟨⟩ denote a grammar rule and a class in grammars. Two alignment results are also shown for grammar [get_info] in this example. Weights are calculated for any alignment as shown here, and the alignment result with the largest weight is selected. In this example, weight $+0.16$ for the grammar [get_info] was the largest.

We consequently obtained the result that grammar rule [get_info] had the highest score for this OOG utterance and its accumulated weight was

User utterance: "Tell me a liaison of Koetsu-ji". (Underlined parts denote OOG.)

ASR result	tell	me	all	Sakyo-ward (ward)		Koetsu-ji (temple)	
grammar [get_info]	tell	me		⟨item name⟩	of	⟨temple name⟩	
WFST output	tell	me	INS	SUB	DEL	Koetsu-ji	
weights	+0.09	+0.06	−0.04	−0.11	−0.02	+0.18	+0.16
grammar [get_info]	tell	me	⟨item name ⟩	of		⟨temple name⟩	
WFST output	tell	me	SUB	SUB		Koetsu-ji	
weights	+0.09	+0.06	−0.21	−0.10		+0.18	+0.02
grammar [search_ward]	tell	me		⟨facility type⟩	in	⟨ward name⟩	
WFST output	tell	me	INS	SUB	DEL	SUB	
weights	+0.09	+0.06	−0.04	−0.12	−0.02	−0.21	−0.24

Figure 3: Example of calculating weights in our grammar verification

+0.16. The result also indicated each type of error as a result of the alignment: ⟨item name⟩ was substituted by "Sakyo-ward", "of" in the grammar [get_info] was deleted, and "all" in the ASR result was inserted.

4 Ranking Help Message Candidates by Integrating Dialogue Context

We furthermore develop a method to rank help message candidates per grammar rule by integrating the grammar verification result and the user's utterance history. This complements information that is often absent from utterances or misrecognized in ASR and prevents that the same help messages are repeated. An outline of the method is depicted in Figure 4.

4.1 Features used in Ranking

Features used in our methods are listed in Table 1. These features are calculated for each help message candidate corresponding to each grammar rule. Features H1 to H5 represent how reliable a grammar verification result is. Feature H1 is a grammar verification score, that is, the resulting accumulated weight described in Section 3. Feature H2 is calculated by normalizing H1 by the total score of all grammar rules. This represents how reliable the grammar verification result is relatively compared to others. Features H3 to H5 represent how partially the user utterance matches with the grammar rule.

Features H6 and H7 correspond to a dialogue context. Feature H6 reflects the case in which users tend to repeat similar utterances when their utterances were not understood by the system. Feature H7 represents whether and how the user knows about the language expression of the grammar rule. This feature corresponds to the *known degree* we previously proposed (Fukubayashi et al., 2006), and prevents a help message the user already knows from being provided repeatedly.

Table 1: Features of each instance (help message candidate)

H1:	accumulated weight of GV (GV score)
H2:	GV score normalized by the total GV score of other instances
H3:	ratio of # of accepted words in GV result to # of all words
H4:	maximum number of successively accepted words in GV result
H5:	number of accepted slots in GV result
H6:	how before the grammar rule was selected as GV result (in # of utterances)
H7:	maximum GV score for the grammar rule until then
H8:	whether it belongs to the "command" class
H9:	whether it belongs to the "query" class
H10:	whether it belongs to the "request-info" class
H11-H17:	products of H8 and each of H1 to H7
H18-H24:	products of H9 and each of H1 to H7
H25-H31:	products of H10 and each of H1 to H7

GV: grammar verification

Features H8 to H10 represent properties of utterances corresponding to the grammar rules, which are categorized into three classes such as "command," "query," and "request-info." In the sightseeing task, the numbers of grammar rules for the three classes were 8, 4, and 11, respectively. More specifically, utterances in either "query" or "request-info" class tend to appear successively because they are used when users try and compare several query conditions; on the other hand, utterances in "command" class tend to appear independently of the context. Features H11 to H31 are the products of features H8, H9, and H10 and each feature from H1 to H7. These were defined to consider combinations of properties of utterances represented by H8, H9, and H10 and their reliability represented by H1 to H7, because RankBoost

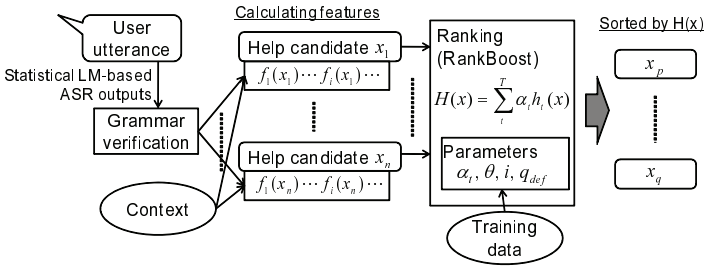


Figure 4: Outline of our ranking method for help message candidates

does not consider them.

4.2 Ranking Algorithm

We adopt RankBoost (Freund et al., 2003), a boosting algorithm based on machine learning, to rank help message candidates. This algorithm can be used for training on several data examples having a certain order of priority. This attribute fits for the problem in this paper; it is difficult even for human annotators to determine the unique appropriate help message to be provided. Target instances x of the algorithm are help message candidates corresponding to grammar rules in this paper.

RankBoost trains a score function $H(x)$ and arranges instances x in the order. Here, $H(x') < H(x'')$ means x'' is ranked higher than x'. This score function is defined as a linear combination of weak rankers giving partial information regarding the order:

$$H(x) = \sum_{t}^{T} \alpha_t h_t(x)$$

where T, $h_t()$, and α_t denote the number of boosting iterations, a weak ranker, and its associated weight, respectively. The weak ranker h_t is defined by comparing the value of a feature f_i of an instance x with a threshold θ. That is,

$$h_t(x) = \begin{cases} 1 \text{ if } f_i(x) > \theta \\ 0 \text{ if } f_i(x) \leq \theta \\ q_{def} \text{ if } f_i(x) = \perp \end{cases} \quad (1)$$

where $q_{def} \in \{0, 1\}$. Here, $f_i(x)$ denotes the value of the i-th feature of instance x, and $\perp$ denotes that no value is given in $f_i(x)$.

5 Experimental Evaluation

5.1 Target Data

Data were collected by 30 subjects in total by using a multi-domain spoken dialogue system that handles five domains such as restaurant, hotel, sightseeing, bus, and weather (Komatani et al., 2008). The data consisted of 180 dialogues and 11,733 utterances. Data from five subjects were used to determine the number of boosting iterations and to improve LMs for ASR. We used utterances in the restaurant, hotel, and sightseeing domains because the remaining two, bus and weather, did not have many grammar rules. We then extracted OOG utterances on the basis of the grammar verification results to evaluate the performance of our method for such utterances. We regarded an utterance whose accumulated weight was negative as OOG. As a result, 1,349 OOG utterances by 25 subjects were used for evaluation, hereafter. These consisted of 363 utterances in the restaurant domain, 563 in the hotel domain, and 423 in the sightseeing domain. These data were collected under the following conditions: subjects were given no instructions on concrete language expressions the system accepts. System responses were made only by speech, and no screen for displaying outputs was used. Subjects were given six scenarios describing tasks to be completed.

We used Julius[1] that is a statistical-LM-based ASR engine. We constructed class 3-gram LMs for ASR by using 10,000 sentences generated from the task grammars and the 600 utterances collected by the five subjects. The vocabulary sizes for the restaurant, hotel, and sightseeing domains were 3,456, 2,625, and 3,593, and ASR accuracies for them were 45.8%, 57.1%, and 43.5%, respectively. These ASR accuracies were not very high because the target utterances were all OOG. A set of possible thresholds in the weak rankers described in Section 4.2 consisted of all feature values that appeared in the training data. The numbers of boosting iterations were determined on the basis of accuracies for the data by the five sub-

[1] http://julius.sourceforge.jp/

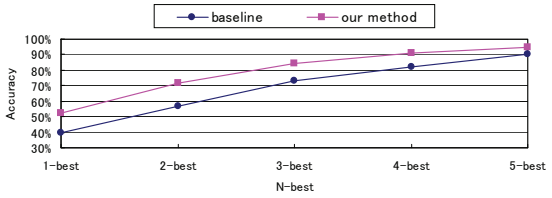


Figure 5: Accuracy when N candidates were provided in sightseeing domain ($1 \leq N \leq 5$)

jects. The numbers were 400, 100, and 500 for the restaurant, hotel, and sightseeing domains.

5.2 Evaluation Criterion

We manually gave five help messages corresponding to grammar rules as reference labels per utterance in the order of having a strong relation to the utterance. The numbers of candidate help messages were 28, 27, and 23 for the restaurant, hotel and sightseeing domains, respectively.

We evaluated our ranking method as the accuracy where at least one of the reference labels was contained in its top N candidates. This corresponds to a probability where at least one appropriate help message was contained in a list of N candidates. The accuracy was calculated by 5-fold cross validation. In the baseline method we set, help messages were provided only by using the grammar verification scores.

5.3 Results

Results in the sightseeing domain are plotted in Figure 5. We can see that our method outperformed the baseline in the accuracies for all N values. All these differences were statistically significant ($p < 0.05$) by the McNemar test. The accuracies were also better in the other two domains for all N values, and the average differences for the three domains were 11.7 points for N=1, 9.7 points for N=2, and 6.7 points for N=3. The differences were large especially for small N values. This result indicates that we can successfully reduce the number of help messages when providing several ones for users. The improvements were derived from the features we incorporated such as the estimated user knowledge in addition to grammar verification results. The baseline method was only based on grammar verification results for single utterances, which contained insufficient information because OOG utterances were often misrecognized or misunderstood.

Table 2: Sum of absolute values of weight α for each feature

H7	H17 (H7*H8)	H19 (H2*H9)	H2	H6
9.58	6.91	6.61	6.02	6.01

We also investigated dominant features by calculating the sum of absolute values of final weight α for each feature in RankBoost. Five dominant features based on the sums are shown in Table 2. These five features include a feature obtained from grammar verification result (H2), a feature about the user's utterance history (H6), a feature representing estimated user knowledge (H7), and features representing properties of the utterances. The most dominant feature was H7, which appeared twice in this table. This was because user utterances were not likely to be OOG utterances again after the user had already known an expression corresponding to the grammar rule, which can be detected when user utterances for it were correctly accepted, that is, its grammar verification score was high. The second dominant feature was H2, which showed that grammar verification results worked effectively.

6 Conclusion

We addressed an issue of OOG utterances in spoken dialogue systems by generating help messages. To manage situations when a user utterance could not be accepted, we robustly estimated a user's intention as a grammar rule that the user intends to use. We furthermore integrated various information as well as the grammar verification results for complementing missing information in single utterances, and then ranked help message candidates corresponding to the grammar rules for efficiently providing them.

Our future work includes the following. The evaluation in this paper was taken place only on the basis of utterances collected beforehand. Providing help messages itself should be evaluated by another experiment through dialogues. Furthermore, we assumed that language expressions of help messages to show an example language expression were fixed. We also need to investigate what kind of expression is more helpful to novice users.

References

Yoav Freund, Raj D. Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Yuichiro Fukubayashi, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. 2006. Dynamic help generation by estimating user's mental model in spoken dialogue systems. In *Proc. Int'l Conf. Spoken Language Processing (INTERSPEECH)*, pages 1946–1949.

Genevieve Gorrell, Ian Lewin, and Manny Rayner. 2002. Adding intelligent help to mixed-initiative spoken dialogue systems. In *Proc. Int'l Conf. Spoken Language Processing (ICSLP)*, pages 2065–2068.

Alexander Gruenstein and Stephanie Seneff. 2007. Releasing a multimodal dialogue system into the wild: User support mechanisms. In *Proc. 8th SIGdial Workshop on Discourse and Dialogue*, pages 111–119.

Melanie Hartmann and Daniel Schreiber. 2008. Proactively adapting interfaces to individual users for mobile devices. In *Adaptive Hypermedia and Adaptive Web-Based Systems, 5th International Conference (AH 2008)*, volume 5149 of *Lecture Notes in Computer Science*, pages 300–303. Springer.

Beth A. Hockey, Oliver Lemon, Ellen Campana, Laura Hiatt, Gregory Aist, James Hieronymus, Alexander Gruenstein, and John Dowding. 2003. Targeted help for spoken dialogue systems: intelligent feedback improves naive users' performance. In *Proc. 10th Conf. of the European Chapter of the ACL (EACL2003)*, pages 147–154.

Kazunori Komatani, Tatsuya Kawahara, and Hiroshi G. Okuno. 2007. Analyzing temporal transition of real user's behaviors in a spoken dialogue system. In *Proc. INTERSPEECH*, pages 142–145.

Kazunori Komatani, Satoshi Ikeda, Tetsuya Ogata, and Hiroshi G. Okuno. 2008. Managing out-of-grammar utterances by topic estimation with domain extensibility in multi-domain spoken dialogue systems. *Speech Communication*, 50(10):863–870.

Akinobu Lee, Kiyohiro Shikano, and Tatsuya Kawahara. 2004. Real-time word confidence scoring using local posterior probabilities on tree trellis search. In *IEEE Int'l Conf. Acoust., Speech & Signal Processing (ICASSP)*, volume 1, pages 793–796.

Cheongjae Lee, Sangkeun Jung, Donghyeon Lee, and Gary Guenbae Lee. 2007. Example-based error recovery strategy for spoken dialog system. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 538–543.

Ryuichi Nisimura, Akinobu Lee, Masashi Yamada, and Kiyohiro Shikano. 2005. Operating a public spoken guidance system in real environment. In *Proc. European Conf. Speech Commun. & Tech. (EUROSPEECH)*, pages 845–848.

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. INTERSPEECH*.

Dialogue behaviour under high cognitive load

Jessica Villing
Graduate School of Language Technology and
Department of Philosophy, Linguistics and Theory of Science
jessica@ling.gu.se

Abstract

For safety reasons, in-vehicle dialogue systems should be able to take the cognitive load of the driver into consideration. However, it is important to distinguish between two types of cognitive load, namely if the cognitive load is affecting the driving behaviour or not. We will present findings from a data collection carried out in the DICO project[1], where the dialogue behaviour under high cognitive load is analysed, and present a novel theory of how to distinguish between different types of workload.

1 Introduction

In-vehicle dialogue systems demand dialogue management that takes the cognitive workload of the driver into consideration. The primary task is the driving, and therefore it is necessary to develop a dialogue system that interferes as little as possible with the driving task. However, the driver's cognitive workload might increase for various reasons, and it is important to distinguish between workload that is driving-induced (i.e. due to, for example, a heavy traffic situation) and workload that is dialogue-induced (i.e. due to a complicated dialogue). If the workload is driving-induced it is probably necessary to pause the dialogue to enable for the driver to concentrate on the driving task, whereas if the workload is dialogue-induced it is instead necessary to facilitate the dialogue task, for example by reformulating a question.

[1] www.dicoproject.org

2 Data collection

DICO is a project that aims to develop a proof-of-concept demo system, showing how a spoken dialogue system can be an aid for drivers. To study how an additional distraction or increase in the cognitive load would affect a driver's dialogue behaviour, a data collection has been made. The goal was to elicit a natural dialogue (as opposed to giving the driver a constructed task such as for example a math task) and make the participants engage in the conversation.

The participants (two female and six male) between the ages of 25 and 36, drove a car in pairs while interviewing each other. The interview questions and the driving instructions were given to the passenger, hence the driver knew neither what questions to discuss nor the route in advance. Therefore, the driver had to signal, implicit or explicit, when she wanted driving instructions and when she wanted a new question to discuss. The passenger too had to have a strategy for when to change topic. The reasons for this setup was to elicit a natural and fairly intense dialogue and to force the participants to change topic and/or domain (e.g. to get driving instructions).

The participants changed roles after 30 minutes, which meant that each participant acted both as driver and as passenger. The cognitive load of the driver was measured in two ways. The driver performed a Tactile Detection Task (TDT (van Winsum et al., 1999)) [2], and workload was also measured by using an IDIS system [3].

[2] When using a TDT, a summer is attached to the driver's wrist. The driver is told to push a button each time the summer is activated. Cognitive load is determined by measuring hit-rate and reaction time.

[3] IDIS determines workload based on the driver's behaviour (for example steering wheel

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 322–325,
Queen Mary University of London, September 2009.

The participants were audio- and videotaped, and then transcribed with the transcription tool ELAN[4], using an orthographic transcription. The annotation scheme was designed to enable analysis of utterances with respect to topic change for each domain.

Domain and topic was defined as:

- *interview* domain: discussions about the interview questions where each interview question was defined as a topic
- *navigation* domain: navigation-related discussions where each navigation instruction belonging to the same row in the given route was defined as a topic
- *traffic* domain: discussions about the traffic situation and fellow road-users where each comment not belonging to a previous event was defined as a topic
- *other* domain: anything that does not fit within the above domains where each comment not belonging to a previous event was defined as a topic

Topics change has been coded as follows:

- *begin-topic*: whatever → topic A (new)
- *end-topic*: topic A (finished) → whatever
- *interrupt-topic*: topic A (unfinished) → whatever
- *resume-topic*: whatever → topic A (unfinished)
- *reraise-topic*: whatever → topic A (finished)

Cognitive load has been annotated as:

- **reliable workload**: annotated when workload is reliably high according to the TDT (reliability was low if response button was pressed more than 2 times after the event).
- **high**: high workload according to IDIS
- **low**: low workload according to IDIS

movements or driver applying the brake). See http://www.roadsafe.com/news/article.aspx?article=210

[4]http://www.lat-mpi.eu/tools/elan/

Silence, regardless of length, has been coded as a pause.

The annotation schema has not been tested for inter-coder reliability. While full reliability testing would have further strengthened the results, we believe that our results are still useful as a basis for future implementation and experimental work.

2.1 High workload

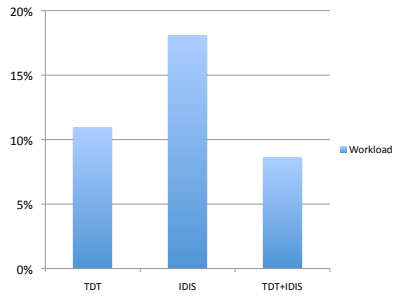


Figure 1: Percentage of annotated workload time.

Figure 1 shows workload measured uniqely by IDIS, uniqely by the TDT and annotations made by IDIS and TDT jointly.

The difference in annotation time can be explained by the fact that IDIS analyses driving behaviour while the TDT measures the driver's reaction time. IDIS is developed to decide when it is suitable to show alarms that are non-critical (such as the indicator for low level of wind screen washer fluid). Since showing the alarm is not time critical, IDIS does not measure the individual driver's workload directly. Taking this into consideration, IDIS measurements alone might be too general and approximate when it comes to adapting a dialogue system to the driver's cognitive load. However, neither IDIS nor TDT in isolation say anything about *what* is causing the high cognitive load, only that something makes the driver unable to pay full attention to the task at hand. These differences can be used to decide what type of workload the driver is experiencing, which will be explained next.

3 Workload management

To determine *type* of workload, the dialoge manager could be extended with a *Cognitive Load Manager (CLM)* which has access to two workload detectors, a *Speech Analyser (SA)* and a *Vehicle State Analyser (VSA)*, see figure 2.

Since the driver is talking to a dialogue system the most convenient method for determining

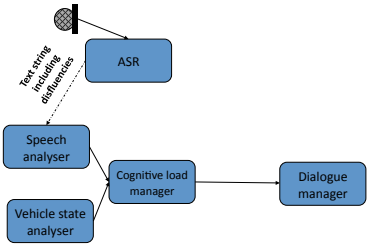


Figure 2: Architecture of the cognitive load manager.

workload level would be to analyse the speech. Studies have for example shown that an increased number of disfluencies such as deletions can indicate increased workload (Shriberg, 2001; Lindstrom et al., 2008). The driver might also make sudden changes of domain, e.g. talk as if addressing fellow road-users, to indicate that she is busy sorting out a difficult traffic situation (Villing et al., 2008). There are no commercial SA systems present today, however research has shown that it is possible to detect workload by analysing the speech signal (Yin et al., 2008).

The VSA analyses the driving behaviour to find signs of increased workload. Variants of VSA-like modules are a reality in the vehicle industry today. For example, if the driver puts the brake on, makes a left turn or manages the radio or the fan, it is assumed that the workload is high.

The CLM collects data from the detectors and determines type of workload based on the combined signals from the SA and the VSA. Type of workload can be set to *driving-induced* (workload that is affecting the driving performance, detected by the VSA) or *dialogue-induced* (workload that is not affecting the driving performance, detected by the SA) based on four assumptions, shown in Table 1.

4 Results

4.1 High workload annotations

Figure 3 shows the number of instances of high workload detected by IDIS alone (possibly driving-induced), by TDT alone (dialogue-induced) and by both IDIS and TDT jointly (driving-induced) for each domain. The TDT makes most annotations in the *other* and *interview* domains and fewest in the *traffic* domain, while the TDT and IDIS jointly makes most annotations in the *traffic* and *other* domains and fewest in the

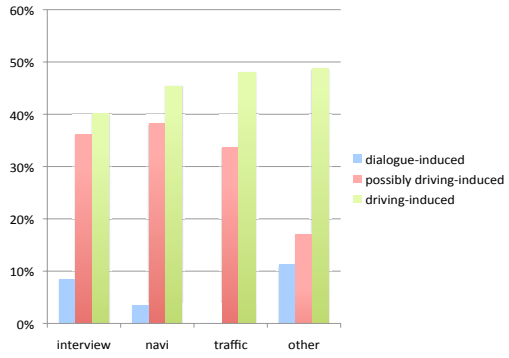


Figure 3: High workload measured for each domain.

interview domain.

To make the SA more powerful, we wanted to investigate if an analysis of dialogue behaviour might improve the possibility to determine workload level. The most frequent topic changes are shown in Figure 4. Most interview related topics are discussed during dialogue-induced workload, while traffic related topics are discussed during driving-induced workload. During possibly driving-induced workload the topics are fairly equally spread. These results are further discussed in (Villing, 2009).

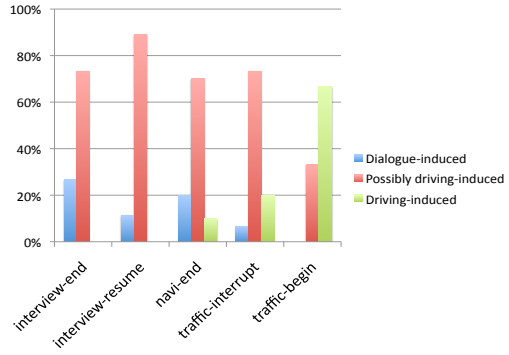


Figure 4: Topic shift during high workload.

Figure 5 shows the average duration of the driver's pauses.

Figure 6 shows that the majority of driver utterances are produced during low workload.

5 Discussion

Figure 5 and 6 shows that an analysis of the speech can give clues about workload level. The duration of the pauses is increasing during high workload, and especially during driving-induced workload. This supports our hypothesis that the dialogue system should pause when the driver needs to concentrate on the driving task. This trend can also be derived from Figure 6, since the number

	SA + VSA	SA	VSA
driver speaking	driving-induced	dialogue-induced	false alarm
driver not speaking	-	-	possibly driving-induced

Table 1: CLM output based on information from the SA and the VSA.

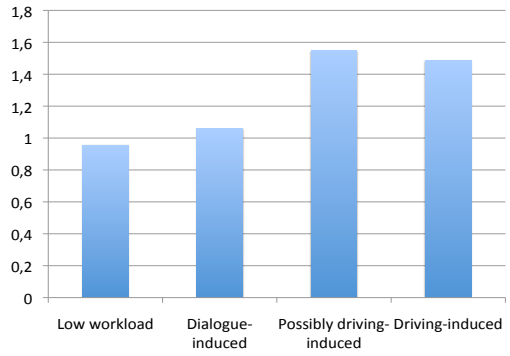


Figure 5: Average pause duration for the driver (in seconds).

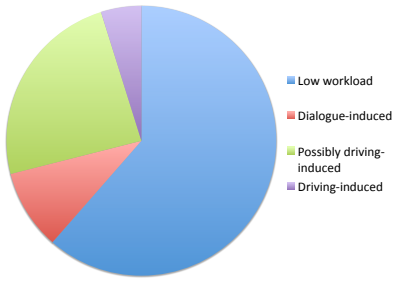


Figure 6: Distribution of driver utterances during low and high workload.

of utterances are decreasing dramatically under high workload when comparing with low workload. The driver seems to make fewest utterances during driving-induced workload.

Looking at Figure 3, it seems like the VSA-like systems present today would benefit from cooperating with a system that is able to make a deeper analyse of the cognitive load of the driver. For example, the *traffic* domain holds almost no dialogue-induced workload annotations but second most driving-induced, supporting the theory that people often make comments about the traffic situation to signal that they have to concentrate on the driving task. The results, although tentative, can be seen as an indication that it is possible to distinguish between different types of cognitive load by analysing both driving behaviour and speech, and that different types of workload demand different dialogue strategies.

6 Future work

Next we will analyse the DICO material regarding interruptions, to find a relevant *interruption place* in the dialogue, i.e. a place where it is most suitable to pause in order to disturb the driver as little as possible.

The resumption behaviour will also be analysed to see who takes the initiative to resume the dialogue and how it is done. The findings will form a basis for a theory of in-vehicle dialogue management.

References

Anders Lindstrom, Jessica Villing, Staffan Larsson, Alexander Seward, Nina Aberg, and Cecilia Holtelius. 2008. The effect of cognitive load on disfluencies during in-vehicle spoken dialogue. In *Proceedings of Interspeech 2008*.

Elisabeth Shriberg. 2001. To "errrr" is human: ecology and acoustics of speech and disfluencies. *Journal of the International Phonetic Association*, 31:153–169.

W van Winsum, M Martens, and L Herland. 1999. The effect of speech versus tactile driver support messages on workload, driver behaviour and user acceptance. tno-report tm-99-c043. Technical report, Soesterberg, Netherlands.

Jessica Villing, Cecilia Holtelius, Staffan Larsson, Ander Lindstrom, Alexander Seward, and Nina Aberg. 2008. Interruption, resumption and domain switching in in-vehicle dialogue. In Bengt Nordstrom and Aarne Ranta, editors, *Proceedings of GoTAL, 6th International Conference of Advances in Natural Language Processing*, volume 5221, pages 488–499, August.

Jessica Villing. 2009. In-vehicle dialogue management - towards distinguishing between different types of workload. In *Proceedings of SiMPE, Fourth Workshop on Speech in Mobile and Pervasive Environments (to appear)*.

Bo Yin, N. Ruiz, Fang Chen, and E. Embikairajah. 2008. Investigating speech features and automatic measurement of cognitive load. In *Proceedings of 2008 IEEE 10th Workshop on Multimedia Signal Processing*, pages 988–993.

A Comparison between Dialog Corpora Acquired with Real and Simulated Users

David Griol
Departamento de Informática
Universidad Carlos III de Madrid
dgriol@inf.uc3m.es

Zoraida Callejas, Ramón López-Cózar
Dpto. Lenguajes y Sistemas Informáticos
Universidad de Granada
{zoraida, rlopezc}@ugr.es

Abstract

In this paper, we test the applicability of a stochastic user simulation technique to generate dialogs which are similar to real human-machine spoken interactions. To do so, we present a comparison between two corpora employing a comprehensive set of evaluation measures. The first corpus was acquired from real interactions of users with a spoken dialog system, whereas the second was generated by means of the simulation technique, which decides the next user answer taking into account the previous user turns, the last system answer and the objective of the dialog.

1 Introduction

During the last decade, there has been a growing interest in learning corpus-based approaches for the different components of spoken dialog systems (Minker, 1999), (Young, 2002), (Esteve et al., 2003), (He and Young, 2003), (Torres et al., 2005), (Georgila et al., 2006), (Williams and Young, 2007). One of the most relevant areas of study has been the automatic generation of dialogs between the dialog manager and an additional module, called the user simulator, which generates automatic interactions with the dialog system.

A considerable effort is necessary to acquire and label a corpus with the data necessary to train good models. User simulators make it possible to generate a large number of dialogs in a very simple way, reducing the time and effort needed for the evaluation of a dialog system each time the system is modified.

The construction of user models based on statistical methods has provided interesting and well-founded results in recent years and is currently a growing research area. A probabilistic user model can be trained from a corpus of human-computer dialogs to simulate user answers. Therefore, it can be used to learn a dialog strategy by means of its interaction with the dialog manager. In the literature, there are several corpus-based approaches for developing user simulators, learning optimal management strategies, and evaluating the dialog system (Scheffler and Young, 2001) (Pietquin and Dutoit, 2005) (Georgila et al., 2006) (Cuayáhuitl et al., 2006) (López-Cózar et al., 2006). A summary of user simulation techniques for reinforcement learning of the dialog strategy can be found in (Schatzmann et al., 2006). In this paper, we propose a statistical approach to acquire a labeled dialog corpus from the interaction of a user simulator and a dialog manager. In our methodology, the new user turn is selected using the probability distribution provided by a neural network. By means of the interaction of the dialog manager and the user simulator, an initial dialog corpus can be extended by increasing its variability and detecting dialog situations in which the dialog manager does not provide an appropriate answer. We propose the use of this corpus for evaluating both our user simulation technique and our dialog system performance.

Different studies have been carried out to compare corpora acquired by means of different techniques and to define the most suitable measures to carry out this evaluation (Schatzmann et al., 2005), (Turunen et al., 2006), (Ai et al., 2007b), (Ai and Litman, 2006), (Ai and Litman, 2007), (Ai et al., 2007a). In this work, we have applied our dialog simulation technique to acquire a corpus in the academic domain, and compared it with a corpus recorded from real users interactions with a spo-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 326–332,
Queen Mary University of London, September 2009.

ken dialog system

The results of this comparison show that the simulated corpus obtained is very similar to the corpus recorded from real user interactions in terms of number of turns, confirmations and dialog acts among other evaluation measures.

The rest of the paper is organized as follows. Section 2 summarizes the main characteristics of the UAH system. Section 3 describes our statistical methodology for user simulation. Section 4 describes the set of measures used to compare the corpus acquired with real users and the simulated corpus. Section 5 presents the results of this evaluation, and finally, the conclusions are presented in Section 6.

2 The UAH System

Universidad Al Habla (UAH - University on the Line) is a spoken dialog system developed to provide spoken access to academic information about our department (Callejas and López-Cózar, 2005) (Callejas and López-Cózar, 2008). The system is comprised of the five typical modules of current spoken dialog systems, concerned with automatic speech recognition (ASR), dialog management (DM), database access (DB Access), data storage (DB) and oral response generation (RG). In addition, we implemented a module called GAG to automatically create ASR grammars (Callejas and López-Cózar, 2007).

The dialog manager of the UAH system was implemented using VoiceXML documents that are dynamically created using PHP. This way, it can adapt the system responses to the context of the conversation and the dialog state, which improves the naturalness of the interaction. For example, the help messages provided by the system take into account the topic that the user and the system are addressing at a particular moment. The context is used as well to decide the confirmation strategy to use.

The information that the system provides can be classified in four main groups: subjects, professors, doctoral studies and registration, as shown in Table 1. As can be observed, the system must ask the user for different pieces of information before producing a response.

A set of 100 dialogs was acquired with the UAH system by means of its interaction with real users. The acquisition process resulted in a spontaneous Spanish speech dialog corpus with 60 different speakers. The total number of user turns was 422 and the recorded material has a duration of 150 minutes. Figure 1 shows an example of a dialog extracted from this corpus. Turns tagged with an 'S' refer to system turns, and turns tagged with 'U' refer to user turns.

S1: Welcome to the UAH system. How can I help you?
U1: I want to know information about subjects in Computer Science
S2: Which subject do you want to consult?
U2: Language Processors I
S3: What would you like to know?
U3: Can you inform me about the schedules of Group D?
S4: The schedule of Group D is: Mondays from 9 to 10 and Wednesdays from 9 to 11 and from 11 to 13. Anything else?
U4: Can you tell me the name of the lecturer?
S5: The name of the lecturer of the Group D of Language Processors I is David Griol. Anything else?
U5: No, thank you.
S6: Thank you for using the UAH system.

Figure 1: An example of a dialog acquired with the interaction of the UAH system with real users

3 The Statistical User Simulation Technique

In the UAH spoken dialogue system, the user simulator developed replaces the functions performed by the ASR and the NLU modules.

The methodology that we have developed for user simulation extends our work for developing a statistical methodology for dialog management (Griol et al., 2008). The user answers are generated taking into account the information provided by the simulator throughout the history of the dialog, the last system turn, and the objective(s) predefined for the dialog. A labeled corpus of dialogs is used to estimate the user model. The formal description of the proposed model is as follows:

Let A_i be the output of the dialog system (the system answer) at time i, expressed in terms of dialog acts. Let U_i be the semantic representation of the user turn. We represent a dialog as a sequence of pairs (*system-turn, user-turn*):

Category	Information provided by the user (including examples)		Information provided by the system
Subject	*Name*	Compilers	Degree, lecturers, responsible lecturer, semester, credits, web page
	Degree, in case that there are several subjects with the same name	Computer science	
	Group name and optionally *type*, in case he asks for information about a specific group	A Theory A	Timetable, lecturer
Lecturers	Any combination of *name* and *surnames*	Zoraida Zoraida Callejas Ms. Callejas	Office location, contact information (phone, fax, email), groups and subjects, doctoral courses
	Optionally *semester*, in case he asks for the tutoring hours	First semester Second semester	Tutoring timetable
Doctoral studies	Name of a doctoral program	Software development	Department, responsible
	Name of a course if he asks for information about a specific course	Object-oriented programming	Type, credits
Registration	Name of the deadline	Provisional registration confirmation	Initial time, final time, description

Table 1: Information provided by the UAH system

$$(A_1, U_1), \cdots, (A_i, U_i), \cdots, (A_n, U_n)$$

where A_1 is the greeting turn of the system (the first turn of the dialog), and U_n is the last user turn. We refer to a pair (A_i, U_i) as S_i, the state of the dialog sequence at time i.

Given this representation, the objective of the user simulator at time i is to find an appropriate user answer U_i. This selection, which is a local process for each time i, takes into account the sequence of dialog states that precede time i, the system answer at time i, and the objective of the dialog $\mathcal{O}$. If the most probable user answer U_i is selected at each time i, the selection is made using the following maximization:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | S_1, \cdots, S_{i-1}, A_i, \mathcal{O})$$

where set $\mathcal{U}$ contains all the possible user answers.

As the number of possible sequences of states is very large, we establish a partition in this space (i.e., in the history of the dialog preceding time i).

Let UR_i be the user register at time i. The user register is defined as a data structure that contains the information provided by the user throughout the previous history of the dialog.The partition that we establish in this space is based on the assumption that *two different sequences of states are equivalent if they lead to the same* UR. After applying the above considerations and establishing the equivalence relations in the histories of the dialogs, the selection of the best U_i is given by:

$$\hat{U}_i = \underset{U_i \in \mathcal{U}}{\operatorname{argmax}} P(U_i | UR_{i-1}, A_i, \mathcal{O}) \qquad (1)$$

We propose the use of a multilayer perceptron (MLP) to make the assignation of a user turn. The input layer receives the current situation of the dialog, which is represented by the term $(UR_{i-1}, A_i, \mathcal{O})$ in Equation 1. The values of the output layer can be viewed as the a posteriori probability of selecting the different user answers defined for the simulator given the current situation of the dialog. The choice of the most probable user answer of this probability distribution leads to Equation 1. In this case, the user simulator will always generate the same answer for the same situation of the dialog. Since we want to provide the user simulator with a richer variability of behaviors, we base our choice on the probability distribution supplied by the MLP on all the feasible user answers.

For the UAH task, the variable $\mathcal{O}$ is modeled taking into account the different types of scenarios defined for the acquisition of the original corpus with real users (33).

The corpus acquired with real users includes information about the errors that were introduced by

the ASR and the NLU modules during this acquisition. This information also includes confidence measures, which are used by the DM to evaluate the reliability of the concepts and attributes generated by the NLU module.

An error simulator module has been designed to perform error generation. The error simulator modifies the frames generated by the user simulator once the UR is updated. In addition, the error simulator adds a confidence score to each concept and attribute in the frames. Experimentally, we have detected 2.3 errors per dialog in our initial corpus. This value can be modified to adapt the error simulator module to the operation of any ASR and NLU modules.

A maximum number of twelve user turns per dialog was defined for acquiring a corpus using our user simulator. A user request for closing the dialog is selected once the system has provided the information defined in the objective(s) of the dialog. The dialogs that fulfill this condition before the maximum number of turns are considered successful. The dialog manager considers that the dialog is unsuccessful and decides to abort it when the following conditions take place:

- The dialog exceeds the maximum number of user turns.
- The answer selected by the dialog manager corresponds with a query not required by the user simulator.
- The database query module provides an error warning because the user simulator has not provided the mandatory information needed to carry out the query.
- The answer generator provides a error warning when the selected answer involves the use of a data not provided by the user simulator.

The semantics selected for the dialog manager is represented through the 27 possible system answers defined for the UAH task. The selection of the possible user answers is carried out using the semantics defined for the user in the NLU module. A dialog corpus of 3000 successful dialogs was acquired using the user simulation technique following a set of scenarios covering all the possible types of queries regarding the information sought.

Figure 2 shows an example of a dialog corresponding to a scenario with two objectives (to know timetables of a given lecturer and subject, and information about the registration in its specific degree). The confidence scores generated by the error simulator are shown between brackets. Sentences in natural language, which are equivalent to the labeled system and user turns, have been incorporated to clarify the explanation.

S1: Opening
Welcome to the UAH system. How may I help you?
U1: Lecturers [0.1]
I want to know information about lecturers.
S2: (Confirmation:Lecturers)
Do you want to know information about lecturers?
U2: Affirmation [0.9]
Yes.
S3: (Question:Lecturer_Name)
Tell me the name of the lecturer.
U3: Lecturer_Name: Ramón López-Cózar [0.8]
Ramón López-Cózar.
S4: (Question:Lecturer_Information)
What would you like to know about Mr. López-Cózar?
U4: (Question:Office) [0.9]
The location of his office please.
S5: (Answer:Office) (New-Query)
{*The office of Ramón López-Cózar is located in the third floor of ...*} *Anything else?*
U5: (Question:Registration) [0.9]
Degree: Computer Science[0.9]
How to register in Computer Science.
S6: (Answer:Registration) (New-Query)
{*Registration information*} *Anything else?*
U6: Negation [0.9] *No.*
S7: (Closing)
Thank you for using the UAH system.

Figure 2: An example of a dialog acquired by means of the simulation technique

4 Evaluation of the Corpora

We used a set of measures to carry out the evaluation of the acquired corpora based on prior work in the dialog literature. (Schatzmann et al., 2005) proposed a comprehensive set of quantitative evaluation measures to compare two dialog corpora. These measures were adapted for our purpose and can be divided into three types:

High-level dialog features
Average number of turns per dialog
Percentage of different dialogs
Number of repetitions of the most seen dialog
Number of turns of the most seen dialog
Number of turns of the shortest dialog
Number of turns of the longest dialog
Dialog style/cooperativeness measures
System dialog acts: Confirmation of concepts and attributes, Questions to require information, and Answers generated after a database query.
User dialog acts: Request to the system, Provide information, Confirmation, Yes/No answers, and Other answers.

Figure 3: Evaluation measures used to compare the acquired corpora

- High-level dialog features: These features evaluate the duration of the dialogs, the amount of information transmitted in the individual turns, and how active the dialog participants are.

- Dialog style/cooperativeness measures: These measures analyze the frequency of the different speech acts and study, for example, the proportion of actions which are goal-directed vs. dialog formalities.

- Task success/efficiency measures: These are computations of the goal achievement rates and goal completion times.

We have defined six high-level dialog features for the evaluation of the dialogs: the average number of turns per dialog, the percentage of different dialogs without considering the attribute values, the number of repetitions of the most seen dialog, the number of turns of the most seen dialog, the number of turns of the shortest dialog, and the number of turns of the longest dialog. Using these measures, we tried to evaluate the success of the simulated dialogs as well as their efficiency and variability with regard to the different objectives.

For dialog style features, we have defined a set of system/user dialog acts. On the system side, we have measured the frequency of confirmations, questions that require information, and system answers generated after a database query. We have not taken into account the opening and closing system turns. On the user side, we have measured the percentage of turns in which the user carries out a request to the system, provide information, confirms a concept or attribute, Yes/No answers, and other answers not included in the previous categories.

We have not considered task success/efficiency measures in our evaluation, since only the dialogs that fulfill the objectives predefined in the scenarios have been incorporated into our corpora. We have considered successful dialogs those that fulfill the complete list of objectives defined in the corresponding scenario. Figure 3 summarizes the complete set of measures used in the evaluation.

5 Evaluation Results

To compare the two corpora, we have computed the mean value for each corpus with respect to each of the evaluation measures shown in the previous section. Then two-tailed t-tests have been employed to compare the means across the two corpora as described in (Ai et al., 2007a). All differences reported as statistically significant have p-values less than 0.05 after Bonferroni corrections.

5.1 High-level Dialog Features

As stated in the previous section, the first group of experiments covers the following statistical properties: i) Dialog length in terms of the average number of turns per dialog, number of turns of the shortest dialog, number of turns of the longest dialog, and number of turns of the most seen dialog; ii) Number of different dialogs in each corpus in terms of the percentage of different dialogs and the number of repetitions of the most seen dialog; iii) Turn length in terms of actions per turn; iv) Participant activity as a ratio of system and user actions per dialog.

	Initial Corpus	Simulated Corpus
Average number of user turns per dialog	4.99	3.75
Percentage of different dialogs	85.71%	77.42%
Number of repetitions of the most seen dialog	5	27
Number of turns of the most seen dialog	2	2
Number of turns of the shortest dialog	2	2
Number of turns of the longest dialog	14	12

Table 2: Results of the high-level dialog features defined for the comparison of the three corpora

Table 2 shows the results of the comparison of the high-level dialog features. It can be observed that all measures have similar values in both corpora. The more significant difference is the average number of user turns. In the four types of scenarios, the dialogs acquired using the simulation technique were shorter than the dialogs acquired with real users. This can be explained by the fact that there was a number of dialogs acquired with real users in which the user asked for additional information not included in the definition of the corresponding scenario once the dialog objectives had been achieved.

5.2 Dialog Style and Cooperativeness

Tables 3 and 4 respectively show the frequency of the most dominant user and system dialog acts. Table 3 shows the results of this comparison for the system dialog acts. It can be observed that there are also only slight differences between the values obtained for both corpora. There is a higher percentage of confirmations and questions in the corpus acquired with real users due to its higher average number of turns per dialog.

Table 4 shows the results of this comparison for the user dialog acts. The most significant difference between both corpora is the percentage of turns in which the user makes a request to the system, which is lower in the corpus acquired with real users. This is possibly because it is less probable that simulated users provide useless information, as it is shown in the lower percentage of the users turns classified as Other answers.

6 Conclusions

In this paper, we have presented a comparison between two corpora acquired using two different techniques. Firstly, we gathered an initial dialog corpus from real user-system interactions. Secondly, we have employed a statistical user simulation technique based on a classification process to automatically obtain a corpus of simulated dialogs. Our results show that it is feasible to acquire a realistic corpus by means of the simulation technique. The experimental results reported indicate that the simulated and real interactions corpora are very similar in terms of number of user turns, user and system dialog style and cooperativeness, and most frequent dialogs statistics. As future work, we plan to employ the simulated dialogs for evaluation purposes and for extracting valuable information to optimize the current dialog strategy.

References

H. Ai and D. Litman. 2006. Comparing Real-Real, Simulated-Simulated, and Simulated-Real Spoken Dialogue Corpora. In *Procs. of AAAI Workshop Statistical and Empirical Approaches for Spoken Dialogue Systems*, Boston, USA.

H. Ai and D.J. Litman. 2007. Knowledge Consistent User Simulations for Dialog Systems. In *Proc. of Interspeech'07*, pages 2697–2700, Antwerp, Belgium.

H. Ai, A. Raux, D. Bohus, M. Eskenazi, and D. Litman. 2007a. Comparing Spoken Dialog Corpora Collected with Recruited Subjects versus Real Users. In *Proc. of the SIGdial'07*, pages 124–131, Antwerp, Belgium.

H. Ai, J.R. Tetreault, and D.J. Litman. 2007b. Comparing User Simulation Models For Dialog Strategy Learning. In *Proc. of NAACL HLT'07*, pages 1–4, Rochester, NY, USA.

Z. Callejas and R. López-Cózar. 2005. Implementing modular dialogue systems: a case study. In *Proc. of Applied Spoken Language Interaction in Distributed Environments (ASIDE'05)*, Aalborg, Denmark.

Z. Callejas and R. López-Cózar. 2007. Automatic creation of ASR grammar rules for unknown vocabulary applications. In *Proc. of the 8th International workshop on Electronics, Control, Modelling, Measurement and Signals (ECMS'07)*, pages 50–55, Liberec, Czech Republic.

Z. Callejas and R. López-Cózar. 2008. Relations between de-facto criteria in the evaluation of a spoken

	Initial Corpus	Simulated Corpus
Confirmation of concepts and attributes	13.51%	12.23%
Questions to require information	18.44%	16.57%
Answers generated after a database query	68.05%	71.20%

Table 3: Percentages of the different types of system dialog acts in both corpora

	Initial Corpus	Simulated Corpus
Request to the system	31.74%	35.43%
Provide information	21.72%	20.98%
Confirmation	10.81%	9.34%
Yes/No answers	33.47%	32.77%
Other answers	2.26%	1.48%

Table 4: Percentages of the different types of user dialog acts in both corpora

dialogue system. *Speech Communication*, 50(8–9):646–665.

H. Cuayáhuitl, S. Renals, O. Lemon, and H. Shimodaira. 2006. Learning Multi-Goal Dialogue Strategies Using Reinforcement Learning with Reduced State-Action Spaces. In *Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 469–472, Pittsburgh (USA).

Y. Esteve, C. Raymond, F. Bechet, and R. De Mori. 2003. Conceptual Decoding for Spoken Dialog systems. In *Proc. of European Conference on Speech Communications and Technology (Eurospeech'03)*, volume 1, pages 617–620, Geneva (Switzerland).

K. Georgila, J. Henderson, and O. Lemon. 2006. User Simulation for Spoken Dialogue Systems: Learning and Evaluation. In *Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1065–1068, Pittsburgh (USA).

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.

Y. He and S. Young. 2003. A data-driven spoken language understanding system. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU'03)*, pages 583–588, St. Thomas (U.S. Virgin Islands).

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*, 26:291–323.

W. Minker. 1999. Stocastically-based semantic analysis. In *Kluwer Academic Publishers*, Boston (USA).

O. Pietquin and T. Dutoit. 2005. A probabilistic framework for dialog simulation and optimal strategy learning. In *IEEE Transactions on Speech and Audio Processing, Special Issue on Data Mining of Speech, Audio and Dialog*, volume 14, pages 589–599.

J. Schatzmann, K. Georgila, and S. Young. 2005. Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems. In *Proc. of SIGdial'05*, pages 45–54, Lisbon (Portugal).

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.

K. Scheffler and S. Young. 2001. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proc. of HLT'02*, pages 12–18, San Diego (USA).

F. Torres, L.F. Hurtado, F. García, E. Sanchis, and E. Segarra. 2005. Error handling in a stochastic dialog system through confidence measures. In *Speech Communication*, pages (45):211–229.

M. Turunen, J. Hakulinen, and A. Kainulainen. 2006. Evaluation of a Spoken Dialogue System with Usability Tests and Long-term Pilot Studies: Similarities and Differences. In *Proc. of the 9th International Conference on Spoken Language Processing (Interspeech/ICSLP)*, pages 1057–1060, Pittsburgh, USA.

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.

S. Young. 2002. The Statistical Approach to the Design of Spoken Dialogue Systems. Technical report, CUED/F-INFENG/TR.433, Cambridge University Engineering Department, Cambridge (UK).

Simultaneous dialogue act segmentation and labelling using lexical and syntactic features

Ramon Granell, Stephen Pulman
Oxford University Computing Laboratory,
Wolfson Building, Parks Road,
Oxford, OX1 3QD, England
ramg@comlab.ox.ac.uk
sgp@clg.ox.ac.uk

Carlos-D. Martínez-Hinarejos
Instituto Tecnológico de Informática,
Universidad Politécnica de Valencia,
Camino de Vera, s/n, 46022, Valencia, Spain
cmartine@dsic.upv.es

Abstract

Segmentation of utterances and annotation as dialogue acts can be helpful for several modules of dialogue systems. In this work, we study a statistical machine learning model to perform these tasks simultaneously using lexical features and incorporating deterministic syntactic restrictions. There is a slight improvement in both segmentation and labelling due to these restrictions.

1 Introduction

Dialogue acts (DA) are linguistic abstractions that are commonly accepted and employed by the the dialogue community. In the framework of dialogue systems, they can be helpful to identify and model user intentions and system answers by the dialogue manager. Furthermore, in other dialogue modules such as the automatic speech recognizer or speech synthesiser, DA information may be also used to increase their performance.

Many researchers have studied automatic DA labelling using different techniques. However, in most of this work it is common to assume that the dialogue turns are already segmented into separate utterances, where each utterance corresponds to just one DA label, as in (Stolcke et al (2000); Ji and Bilmes (2005); Webb et al (2005)). This is not a realistic situation because the segmentation of turns into utterances is not a trivial problem.

There have been many previous approaches to segmentation of turns prior to DA labelling, beginning with (Stolcke and Shriberg (1996)). Typically some combination of words and part of speech (POS) tags is used to predict segmentation boundaries. In this work we make use of a statistical model to solve both the DA labelling task and the segmentation task simultaneously, following (Ang et al (2005); Martínez-Hinarejos et al (2006)). Our aim is to see whether going beyond the word n-gram models can improve accuracy, using syntactic information (constituent structure) obtained from the dialogue transcriptions. We examine whether this information can improve the segmentation of the dialogue turns into DA segments. Intuitively, it seems logical to believe that most of these segments must coincide with particular syntactic structures, and that segment boundaries would respect constituent boundaries.

2 Dialogue data

The dialogue corpus used to perform the experiments is the Switchboard database (SWBD). It consists of human-human conversations by telephone about generic topics. There are 1155 5-minute conversations, comprising approximately 205000 utterances and 1.4 million words. The size of the vocabulary is approximately 22000 words.

All this corpus has been manually annotated at the dialogue act level using the SWBD-DAMSL scheme, (Jurafsky et al (1997)), consisting of 42 different labels. Every dialogue turn was manually segmented into utterances. The average number of segments (utterances) per dialogue turn is 1.78 with a standard deviation of 1.41. Each utterance was assigned one SWBD-DAMSL label (see Figure 1).

3 Syntactic analysis of DA segments

An initial analysis of the syntactic structures of the dialogue data was performed to study their possible relevance for DA segmentation.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 333–336,
Queen Mary University of London, September 2009.

- $LAUGH he waits until it gets about seventeen below up here $SEG and then he calls us , $SEG
sd sd
- he waits until it gets about seventeen below up here and then he calls us .

Figure 1: The first row is an original segmented dialogue turn, where the $SEG label indicates the end of a DA segment. The second row contains the corresponding DA label for each segment, where "sd" corresponds to the SWBD-DAMSL label of Statement non-opinion. The third row is the input for the parser.

3.1 Parsing of spontaneous dialogues

One of the main problems we face when we try to syntactically analyse a corpus transcribed from spontaneous speech by different people such as SWBD corpus, is the inconsistency of annotation conventions for spontaneous speech phenomena and punctuation marks. This can be problematic for parsers, as they work at the sentence level. Some of the dialogue turns of the SWBD corpus are not transcribed using consistent punctuation conventions. We therefore carried out some preprocessing so that all turns end with proper punctuation marks. Additionally, the non-verbal labels (e.g. $LAUGH, $OVERLAP, $SEG, ...) are removed. In Figure 1 there is an example of this process.

The Stanford Parser, (Klein and Manning (2003)) was used for the syntactic analysis of the transcriptions of SWBD dialogues. The English grammar used to train the parser is based on the standard LDC Penn Treebank WSJ training sections 2-21. Is is important to remark that the nature of the training corpus (journalistic style reports) is different from the transcriptions of spontaneous speech conversations. We would therefore expect a decrease in accuracy. As output of the parsing process, a tree that contains syntactic structures was provided (e.g. see Figure 2).

```
(ROOT
 (S (: -)
  (S
   (NP (PRP he))
    (VP (VBZ waits)
     (SBAR (IN until)
      (S
       (NP (PRP it))
       (VP (VBZ gets)
        (PP (IN about)
         (NP (NN seventeen)))
        (PP (IN below)
         (ADVP (RB up) (RB here))))))))
   (CC and)
   (S
    (ADVP (RB then))
    (NP (PRP he))
    (VP (VBZ calls)
     (NP (PRP us))))
   (. .)))
```

Figure 2: Example of the syntactic analysis of the dialogue turn that appears in Figure 1.

3.2 Syntactic features and segmentation

As we are interested in studying the coincidence of syntactic structures with DA segments, we will select two general features for each word (see Figure 3):

- Most general syntactic category that starts with a word, **(MGSS)**, i.e., the root of the current subtree of the syntactic analysis, (e.g. in Figure 2, "CC" is the MGSS of the first word of the second segment, "and").
- Most general syntactic category that ends with a word, **(MGSE)**, i.e., the root of the subtree of the syntactic analysis that ends with that word, (e.g. in Figure 2, "S" is the MGSE of last word of the first segment, "here").

Using these features, we have analysed the syntactic categories of boundary words of segments. Particularly, it seems interesting to study MGSE of last word of the segment and MGSS of first word of the segment, because it indicates which syntactic structure ends before the segment boundary and which one starts after it. As there is always the beginning of a segment with the first word of the turn and the end of a segment with the last word of the turn, we are ignoring these for the analysis, because we are looking for intra-turn segments. Results of this analysis can be seen in Table 1.

4 The model

The statistical model used to DA label and segment the dialogues is extensively explained in (Martínez-Hinarejos (2008)). Basically, it is

ROOT+**-+**: **$LAUGH** S+**he**+NP VP+**waits**+VBZ SBAR+**until**+IN S+**it**+NP VP+**gets**+VBZ PP+**about**+IN NP+**seventeen**+PP PP+**below**+IN ADVP+**up**+RB RB+**here**+S **$SEG** CC+**and**+CC S+**then**+ADVP NP+**he**+NP VP+**calls**+VBZ NP+**us**+S .+**.**+ROOT **$SEG**

Figure 3: For each word of the example turn of Figure 1, MGSS (item before the word) and MGSE (item after the word) are obtained from the tree of Figure 2. Non-verbal labels were reincorporated.

MGSE			MGSS		
Occ	%	Cat	Occ	%	Cat
33516	37.1	,	30318	33.5	ROOT
30640	33.9	ROOT	19988	22.1	CC
7801	8.6	:	13275	14.7	NP
7134	7.9	S	10187	11.3	S
2687	3.0	NP	3508	3.9	SBAR
2319	2.6	PRN	3421	3.8	ADVP
750	0.8	VP	2034	2.2	VP
531	0.6	ADVP	1957	2.2	INTJ
478	0.5	PP	1300	1.4	UH
465	0.5	RB	972	1.1	PP
4078	4.5	Other	3481	3.8	Other

Table 1: Occurrences and percentage of the syntactic categories that correspond with the most frequent MGSE of the last segment word (except last segment) and MGSS of the first segment word (except first segment).

based on a combination of a Hidden Markov Model at lexical level and a Language Model (n-gram) at DA level. The Viterbi algorithm is used to find the most likely sequence of DA labels according to the trained models. The segmentation is obtained from the jumps between DAs of this sequence.

The previous section has shown that the MGSE and MGSS for the segments boundary words are concentrated in a small set of categories (see Table 1). Therefore, one quick and easy way to incorporate this information to the existing model is to add some restrictions during the decoding process, giving the model:

$$\widehat{U} = \arg\max_{U} \max_{r,s_1^r} \prod_{k=1}^{r} \Pr(u_k|u_{k-n-1}^{k-1}) \cdot \cdot \Pr(W_{s_{k-1}+1}^{s_k}|u_k)\sigma(x_{s_k})$$

where $\widehat{U}$ is the sequence of DAs that we will get from the annotation/segmentation process. The search process produces a segmentation $s = (s_0, s_1, \ldots, s_r)$, that divides the word sequence W into the segments $W_{s_0+1}^{s_1} W_{s_1+1}^{s_2} \ldots W_{s_{r-1}+1}^{s_r}$. Each segment is assigned to a DA u_i that forms the DA sequence $U = u_1 \ldots u_r$. x_i corresponds to the syntactic features of the i word that can be MGSE, MGSS or both of them, and

$$\sigma(x_i) = \begin{cases} 1 & \text{if } x_i \in X \\ 0 & \text{otherwise} \end{cases}$$

where X can be a subset of all the possible syntactic categories that correspond to:

1. the most frequent MGSE of last segment word, if x is MGSE.
2. the most frequent MGSS of first segment word, if x is MGSS
3. the most frequent combinations of both previous sets.

It means that we will only allow a segment ending when the MGSE of a word is in this set, or a start of a segment when the MGSS of the following word is in the corresponding set or both conditions at the same time.

5 Experiments and results

Ten cross-validation experiments were performed for each model using, in each experiment a training partition composed of 1136 dialogues and a test set of 19 dialogues, as in (Stolcke et al (2000); Webb et al (2005); Martínez-Hinarejos et al (2006)). The N-grams were obtained using the SLM toolkit (Rosenfeld (1998)) with Good-Turing discounting and the HMMs were trained using the Baum-Welch algorithm. We use the following evaluation measures:

- To evaluate the labelling, we use the DA Error Rate (equivalent to Word Error Rate) and the percentage of error labelling of whole turns.
- For the segment evaluation, we only check where the segments bounds are produced (word position in the segment), making use of F-score obtained from precision and recall.

The results from using different sizes for the set X are shown for labelling performance in Tables 2 and 3, and F-score of the segmentation in Table 4.

Model/SizeX	5	10	20	All
MGSE	53.31	54.76	54.60	54.76
MGSS	53.35	**52.76**	54.92	54.76
Both	53.58	52.84	54.76	54.76

Table 2: DAER for models using MGSE, MGSS and both features. SizeX indicates the size of the set of most frequent categories accepted. Without syntactic categories (baseline) we obtain a DAER of 54.41.

Model/SizeX	5	10	20	All
MGSE	53.61	55.41	55.34	55.77
MGSS	53.61	53.32	55.63	55.77
Both	53.46	**53.10**	55.19	55.77

Table 3: Percentage of error of labelling of complete turns for all the possible models. The baseline value is 55.41.

Model / SizeX	5	10	20	All
MGSE	73.08	71.18	71.44	71.17
MGSS	73.60	73.72	71.44	71.17
Both	**74.36**	74.08	71.75	71.16

Table 4: F-score of segmentation. The baseline value is 71.17.

6 Discussion and future work

In this work, we have used lexical and syntactic features for labelling and segmenting DAs simultaneously. Syntactic features obtained automatically were deterministically applied during the statistical decoding process. There is a slight improvement using syntactic information, obtaining better results than reported in other work such as (Martínez-Hinarejos et al (2006)). The F-score of the segmentation improves 3% using the syntactic features, however values are slightly worse (2%) than results in (Stolcke and Shriberg (1996)).

As future work, we think that incorporating the syntactic information in a non-deterministic way might further improve the annotation and segmentation scores. Furthermore, it is possible to make use of additional information from the syntactic structure, rather than just the boundary information we are currently using. Finally, an evaluation over different corpora must be done to check both the performance of the proposed model and the reusability of the syntactic sets.

Acknowledgments

This work was partially funded by the Companions project (http://www.companions-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434.

References

Ang J., Liu Y., Shriberg E. 2005. Automatic Dialog Act Segmentation and Classification in Multiparty Meetings. Proc. ICASSP, Philadelphia, USA, pp. 1061-1064

Ji, G and Bilmes, J. 2005. Dialog act tagging using graphical models. Proc. ICASSP, Philadelphia, USA

Jurafsky, D. Shriberg, E., Biasca, D. 1997. Switchboard swbd-damsl shallow- discourse-function annotation coders manual. Tech. Rep. 97-01, University of Colorado Institute of Cognitive Science

Klein D. and Manning, C. D. 2003. Accurate Unlexicalized Parsing. Proc. ACL, Sapporo, Japan, pp. 423-430

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

Martínez-Hinarejos, C. D., Benedí, J. M., Granell, R. 2008. Statistical framework for a spanish spoken dialogue corpus. Speech Communication, vol. 50, number 11-12, pp. 992-1008

Rosenfeld, R. 1998. The cmu-cambridge statistical language modelling toolkit v2. Technical report, Carnegie Mellon University

Stolcke, A. and Shriberg, E. 1996. Automatic linguistic segmentation of conversational speech. Proc. of ICSLP, Philadelphia, USA

Stolcke, A., Coccaro, N., Bates, R., Taylor, P., van Ess-Dykema, C., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., Meteer, M. 2000. Dialogue act modelling for automatic tagging and recognition of conversational speech. Computational Linguistics 26 (3), 1-34

Webb, N., Hepple, M., Wilks, Y. 2005. Dialogue act classification using intra-utterance features. Proc. of the AAAI Workshop on Spoken Language Understanding. Pittsburgh, USA

The Spoken Dialogue Challenge

Alan W Black and Maxine Eskenazi
Language Technologies Institute
Carnegie Mellon University, Pittsburgh, PA, USA
{awb,max}@cs.cmu.edu [name]

Abstract

In the field of speech and language processing the introduction of "Challenges" has helped focus the focus a field, allowing detailed comparisons of systems and techniques, bringing new members into the field, and facilitating advancements in core research. Although the idea of a spoken dialogue challenge has been discussed for some time, this is the first attempt to bring these discussions together and take concrete action.

1 Motivation for a centralized Challenge

The idea of a centralized challenge in the fields of speech and language where different systems and techniques are applied to the same data has been with us for some time. Probably the most lengthy and successful challenge is the DARPA-funded speech recognition set of challenges [3] which started in the 1980s and came to a peak in the 1990s. They still continue through different programs, and in spite of being criticized for focusing Automatic Speech Processing (ASR) on simple measures of success, it is clear that they have helped make ASR fundamentally better.

The important points in a successful challenge are: a well-defined task that the community considers to be challenging for the current state of the art; sufficient number of participants with varied systems so that the performance of different techniques can be evaluated; generally accepted evaluation/success criteria; and a sense of collaboration between participants so that details of their entries may be reasonably shared. It is important that the challenge does not become an outright competition, but striving to produce the "best" system encourages exploring new ideas.

2 Introduction

Within the field of spoken dialogue systems perhaps the most similar coordinated evaluation of multiple systems was part of the DARPA-funded Communicator program. This program gave spoken dialogue system access to a flight information and booking system. In 2000-2001 a number of systems developed as part of the program were centrally evaluated by NIST. A number of external callers accessed each system and the results were published in [9]. Since spoken dialogue systems do not have as clear a measure of success as ASRs, a number of different measures were designed, both objective and subjective. And other analyses of the evaluation were made by a number of groups including [1].

Other competitions have been developed including the Loebner Prize [5], which addresses the issues of (at least originally) text-based conversations that attempt to pass the Turing Test. This year, however, the Loebner Prize is being held in conjunction with Interspeech 2009 and will have a speech component.

3 Proposed Spoken Dialogue Challenge

Although it is clear there are many possible choices for this challenge we feel the first round of the challenge should be simple and clear, allowing for the types of tasks to grow in variety over the years. Our initial proposal builds on our own experience in building deployed spoken dialogue systems. Our system has already collected a large number of example dialogues that can be used for training and made available in the first year to participants.

The target domain is bus information. This was chosen because it is a popular domain, used by several different dialogue systems. It is useful to the public and doesn't require dealing with sensitive information. The system is simple enough that it can be re-implemented without

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 337–340,
Queen Mary University of London, September 2009.

extensive additional work. We have already collected over 75,000 deidentified dialogues in the domain that we can distribute for training purposes.

The second important detail to define is what we expect of the participants (the task). There is a wide variety of research interests in the current spoken dialogue field. It is impossible for a single task to provide suitable challenges for everyone in its first iteration, so it is important to find a task in which a substantial subset can compete.

Given the bus information task as the most wide-reaching for the first year, we see three levels of participation:

1. Build a bus information system with your spoken dialogue architecture
2. Build a bus information system with free software tools such as Olympus II.
3. Take the existing Let's Go Bus Information system and adapt it with your components.

These three levels offer a certain amount of freedom to the participants. If participants are only interested in one small part of the dialogue task they can added new components to the Let's Go Bus Information system [7], such as a new recognizer, a new synthesizer, or address core dialog components like error recovery etc. The initial language would be (US) English but we would like to extend this in later years.

3.1 Evaluation

Evaluation of spoken dialogue systems is still very much a research issue and we see the Spoken Dialogue Challenge as a mechanism to aid that area of research. Following structures in the Speech Synthesis Blizzard Challenge [2], we propose to have a number of different groups of users call the submitted systems.

Group 1 - Dialog Researchers: each participant group will provided a number of callers who will be given scenarios and asked to call some of the participating systems. Dialogue Researchers are probably the best users of a system and are also the most personally interested in completing the task.

Group 2 – Native Speaker Undergraduates: this group will be paid and intended to be the most homogeneous group of callers.

Group 3 - Volunteers: by requesting for volunteers through mailing lists and the web we will collect the third set of callers.

Having three sets of callers will enable us to perform correlation between the groups and therefore try to find reliable statistics.

We also propose to run the Challenge on two levels. All participating systems will take part in these initial evaluations. The best (or most stable) systems can then be deployed on the Let's Go Live system which provides bus information to the people of Pittsburgh. Thus real users, who are interested in the time of the next bus rather than the success of the dialogue system, will provide an additional set of caller statistics.

The first groups of users will be given scenarios (they are unlikely to be familiar with Pittsburgh busses), and will fill in a web questionnaire after each call. For the live test with real users no questionnaire will be possible. Real users are uninterested in answering any further questions, and a design that gets some subset to fill in questionnaires would probably yield a different result than standard real users.

Evaluation will be through objective and subjective measures. Although evaluation of dialogue systems is still a research issue, we will offer the conventional techniques including task completion and number of turns as well as questionnaires about user satisfaction. As we do have some support for labeling we would also consider calculating word error and other dialogue state level labeling to find accuracy of systems.

3.2 Where to base the first iteration of the Challenge

Why should this challenge be based at Carnegie Mellon? First CMU has been prominent in spoken dialogue research for many years and importantly has provided free software systems to jumpstart others' system building efforts. Olympus II [6] contains everything needed from recognition, synthesis, parses, generators and a dialogue system that allows others to build complete systems. We have provided tutorials describing how to use each of the components and the whole systems.

We also have the experience of running Let's Go Lab [8], which offers a real-time platform with real users that spoken dialogue studies from other institutions have used. We have already gained experience in dialogue evaluation and in providing support for other dialogue teams. This makes the first challenge a clear extension of our existing spoken dialogue evaluation services. After this first iteration, other institutions and applications will be chosen under similar criteria.

3.3 Challenges for the Challenge

Challenges must be supported by the community. The committee governing this challenge therefore consists of leading experts representing a variety of areas in spoken dialogue research: Dr Jason Williams, (AT&T), Dr Dan Bohus, (Microsoft), Prof David Traum (USC), Prof Kristiina Jokinen (Helsinki) and Prof Helen Meng (CU Hong Kong).

Evaluation of spoken dialogue systems is still very much a research issue. Thus the Spoken Dialogue Challenge will become a mechanism that serves that area of research too. The initial metrics that will be produced will include objective measures such as task completion, number of turns etc, and subjective measures (from questionaires) such as user satisfaction. The dialogue data collected during the Challenges will be made available to researchers to further investigate evaluation metrics.

There are a number of non-trivial issues in bringing a successful challenge like this together. Although we do not yet have all of the solutions we are aware of many of the issues.

To be successful we need to have participants. Since there is no funding for participants it is hard for them to devote resources to a task for which they are not being funded. Building a dialogue system is a considerable amount of effort and it may be that only a few groups have the additional resources to participate. We will make every effort to make participation as easy as possible but we know from experience that getting, especially initial, participants is hard. As the Challenge matures over time, we expect to see more participants as people see it as a worthwhile endeavor, and modify their research agenda to include it.

A second issue is computing infrastructure, for an anticipated international set of participants we have to address how callers access systems. International telephone charges notwithstanding, international lines are often more prone to noise than national lines (and often have a delay). Alternatives such as voice over IP (including Skype) have been suggested but they bring in their own issues, ASR of VoIP encoded calls is not the same as ASR over cell phones or landlines. We may have versions of the systems made available at centers on each continent where there are participants.

3.4 Future extensions

We see The Spoken Dialogue Challenge as a long term progressive event. There are a number of current research areas that will be hard to include in the initial instantiation but are certainly worth including in future years. A couple of specific areas are worth considering. It would be good if the systems could be automatically tested. This implies some sort of user simulator, such research has been considered by a number of research groups and it is certainly an interesting direction to take. The second area that is currently active in dialogue research is reinforcement learning or any technique that requires significant numbers of interactions and dynamically learns from them. Again we would like to see this added to future Challenges.

4 Timeline

After consultation with participants at SigDial 2009, we will put out a call for participation in mid-October 2009 in a Challenge that will start in early 2010. We will allow at least 6 months for groups to develop their systems, and allow 3 months for evaluation. The intention is to then present the results and descriptions of the participating systems at a dedicated workshop or special session of a conference.

References

[1] Bennett, C., "A Comparative Analysis of DARPA Communicator Systems," Presentation at DARPA Communicator PI Meeting, New Orleans, Louisiana, July 2001

[2] Black, A., and Tokuda, K., (2005) Blizzard Challenge -- 2005: Evaluating corpus-based speech synthesis on common datasets Interspeech 2005, Lisbon, Portugal.

[3] DARPA, "The DARPA speech recognition evaluation workshops," http://www.nist.gov/speech/publications/index.htm.

[4] Eskenazi, M., Black, A., Raux, A. and Langner, B. "Let's Go Lab: a platform for evaluation of spoken dialog systems with real world users", Interspeech 2008, Brisbane, Australia.

[5] Loebner Prize

http://www.loebner.net/Prizef/loebner-prize.html

[6] Olympus II Dialog System Framework http://wiki.speech.cs.cmu.edu/olympus/index.php/Olympus

[7] Raux, A., Bohus, D., Langner, B., Black, A., and Eskenazi, M. (2006) Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience, Interspeech 2006 - ICSLP, Pittsburgh, PA.

[8] Raux, A., Langner, B., Black, A. and Eskenazi, M. "Building Practical Spoken Dialog Systems" ACL/HLT 2008 Tutorial, Columbus, Ohio.

[9] Walker, M., Passonneau, R., and Boland, J. "Quantitative and qualitative evaluation of Darpa Communicator spoken dialogue systems" 39th ACL, pp 515-222, Toulouse, France, 2001.

Unsupervised Classification of Dialogue Acts using a Dirichlet Process Mixture Model

Nigel Crook, Ramon Granell, and Stephen Pulman
Oxford University Computing Laboratory
Wolfson Building
Parks Road, OXFORD, UK
nigc@comlab.ox.ac.uk
ramg@comlab.ox.ac.uk
sgp@clg.ox.ac.uk

Abstract

In recent years Dialogue Acts have become a popular means of modelling the communicative intentions of human and machine utterances in many modern dialogue systems. Many of these systems rely heavily on the availability of dialogue corpora that have been annotated with Dialogue Act labels. The manual annotation of dialogue corpora is both tedious and expensive. Consequently, there is a growing interest in unsupervised systems that are capable of automating the annotation process. This paper investigates the use of a Dirichlet Process Mixture Model as a means of clustering dialogue utterances in an unsupervised manner. These clusters can then be analysed in terms of the possible Dialogue Acts that they might represent. The results presented here are from the application of the Dirichlet Process Mixture Model to the Dihana corpus.

1 Introduction

Dialogue Acts (DAs) are an important contribution from discourse theory to the design of dialogue systems. These linguistics abstractions are based on the illocutionary force of speech acts (Austin, 1962) and try to capture and model the communicative intention of human or machine utterances. In recent years, several dialogue systems have made use of DAs for modelling discourse phenomena in either the Dialogue Manager (Keizer et al., 2008), Automatic Speech Recogniser (Stolcke et al., 2000) or the Automatic Speech Synthesiser (Zovato and Romportl, 2008). Additionally, they have been used also in other tasks such as summarisation, (Murray et al., 2006). Therefore, a correct DA classification of dialogue turns can bring benefits to the performance of these modules and tasks.

Many machine learning approaches have been used to automatically label DAs. They are usually based on Supervised Learning techniques involving combinations of Ngrams and Hidden Markov Models (Stolcke et al., 2000; Martínez-Hinarejos et al., 2008), Neural Networks (Garfield and Wermter, 2006) or Graphical Models (Ji and Bilmes, 2005). Relatively few approaches to DA classification have been based on unsupervised learning methods. Some promising results were reported by Anderach et al (Andernach et al., 1997; Andernach, 1996) who applied Kohonen Self Organising Maps (SOMs) to the problem of DA classification. Although the SOM is *nonparametric* in the sense that it doesn't require that the number of clusters to be found in the data be a parameter of the SOM that is specified before clustering begins, it's capacity to detect clusters is limited to the size of the two-dimensional lattice onto which the clusters are projected, and the size of this lattice *is* determined prior to clustering. This paper investigates the use of an unsupervised, nonparametric Bayesian approach to automatic DA labelling: namely the Dirichlet Process Mixture Model (DPMM). Specifically, the paper reports results from applying the Chinese Restaurant Process (CRP), a popular approach to DPMMs, to the automatic labelling of DAs in the Dihana corpus. The Dihana corpus (J.M.Benedí et al., 2006) has previously been used for the same task but with a supervised learning approach (Martínez-Hinarejos et al., 2008). The results reported here indicate that, treating each utterance as a *bag of words*, the CRP is capable of automatically clus-

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 341–348,
Queen Mary University of London, September 2009.

tering most utterances according to speaker, level 1 and in some cases level 2 DA annotations (see below).

2 The Dihana corpus

The Dihana corpus consists of human-computer spoken dialogues in Spanish about queuing information of train fares and timetables. The acquisition was performed using the Wizard of Oz (WoZ) technique, where a human simulates the system following a prefixed strategy. User and system utterances are different in nature, user utterances are completely spontaneous speech whereas system utterances are based on pre-written patterns that the WoZ selected according to what the user said in the previous turn, the current dialogue state and the WoZ strategy. There is a total of 900 dialogues with a vocabulary of 823 words. However, after applying a process of name entity recognition (cities, times, number, ...) and making the distinction between system and user words there are 964 different words. The same process of name entity recognition was also used by Martinez Hinarejos (Martínez-Hinarejos et al., 2008)

2.1 Annotation scheme

Dialogues were manually annotated using a dialogue act annotation scheme based on three levels (see Table 1). The first level corresponds to the general intention of the speaker (speech act), the second level represents the implicit information that is referred to in the first level and the third level is the specific data provided in the utterance. Using these three levels and making the distinction between user and system labels, there are 248 different labels (153 for the user and 95 for the system). Combining only first and second level there are 72 labels (45 for user and 27 for system), and with only first level there are 16 labels (7 for user and 9 for system).

Annotation was done at utterance level. That is, each dialogue turn was divided (segmented) into utterances such that each one corresponds to a unique DA label. An example of the segmentation and annotation of two turns of a dialogue can be seen in Figure 1

Level	Labels
First	Opening, Closing, Confirmation, Undefined, Not-understood, Waiting, Consult, Acceptance, Rejection
Second	Departure-hour, Arrival-hour, Fare, Origin, Destination, Day, Train-Type, Service, Class, Trip-time
Third	Departure-hour, Arrival-hour, Fare, Origin, Destination, Day, Train-Type, Service, Class, Trip-time, Order-number, Number-trains, Trip-type

Table 1: Set of dialogue act labels used in the Dihana corpus

3 Dirichlet Process Mixture Models

This paper present a Dirichlet Process Mixture Model (DPMM) (Maceachern and Müller, 1998; Escobar and West, 1995; Antoniak, 1974) for the automatic, unsupervised clustering of the utterances in the Dihana corpus. This approach treats each utterance as a *bag of words* (i.e. an unordered collection of words) (Sebastiani, 2002). Utterances are clustered according to the relative counts of word occurrences that they contain so that utterances with similar histograms of word counts will, in general, appear in the same cluster.

Bayesian methods for unsupervised data clustering divide into parametric and nonparametric approaches. Parametric approaches to clustering such as Finite Bayesian Mixture Models (Mclachlan and Peel, 2000) require prior estimation of the number of clusters that are expected to be found in the data. However, it is not always possible to know this in advance and often it is necessary to repeat a modelling experiment many times over a range of choices of cluster numbers to find an optimal number of clusters. Sub-optimal choices for the number of clusters can lead to a degradation in the generalisation performance of the model. Nonparametric approaches to mixture modelling, on the other hand, do not require prior estimates of the number of clusters in the data; this is discovered automatically as the model clusters the data. Dirichlet Processes offer one approach to developing Bayesian nonparametric mixture models. The remainder of this section briefly introduces DPMMs, beginning with a brief look at finite Bayesian mixture models which will serve as useful background for presenting the Chinese Restaurant Process, the Dirichlet Process paradigm used in this paper.

Speaker	Utterance	Transcription		
		Level 1	Level 2	Level 3
S	S1	Welcome to the railway information system. How may I help you?		
		Opening	Nil	Nil
U	U1	Could you tell me the departure times from Valencia		
		Question	Departure-hour	Origin
	U2	to Madrid .		
		Question	Departure-hour	Destination

Figure 1: An example of some turns from an annotated dialogue of DIHANA corpus.

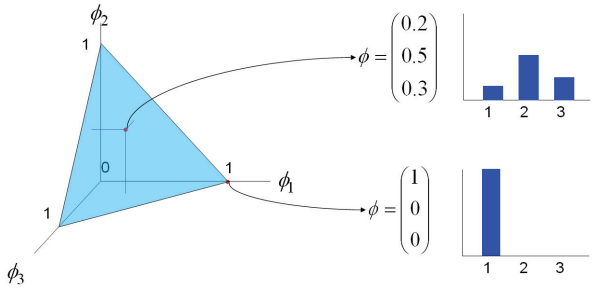


Figure 2: A 3-simplex with two examples points and the corresponding distributions

3.1 Finite Bayesian Mixture Models

A Dirichlet distribution is defined as a *measure on measures*. Specifically, a Dirichlet distribution defines a probability measure over the k-simplex. The k-simplex is a convex hull constructed so that each point on the surface of the simplex describes a probability distribution over k outcomes:

$$Q_k = \{(x_1, \ldots, x_k) : x_i \geq 0 \quad \forall i \in \{1 \ldots k\}, \sum_{i=1}^{k} x_i = 1\}$$

Figure 2 shows a 3-simplex with two example points and the corresponding distributions. The Dirichlet distribution places a probability measure over the k-simplex so that certain subsets of points on the simplex (i.e. certain distributions) have higher probabilities than others (Figure 3). The probability measure in the Dirichlet is parameterised by a set of positive, non-zero concentration constants $\boldsymbol{\alpha} = \{\alpha_1, \ldots \alpha_k : \alpha_i > 0\}$, written $Dirichlet_k(\alpha_1, \ldots \alpha_k)$. The effects of different values of $\boldsymbol{\alpha}$ for the 3-simplex are shown in Figure 3.

The probability density function of the Dirichlet

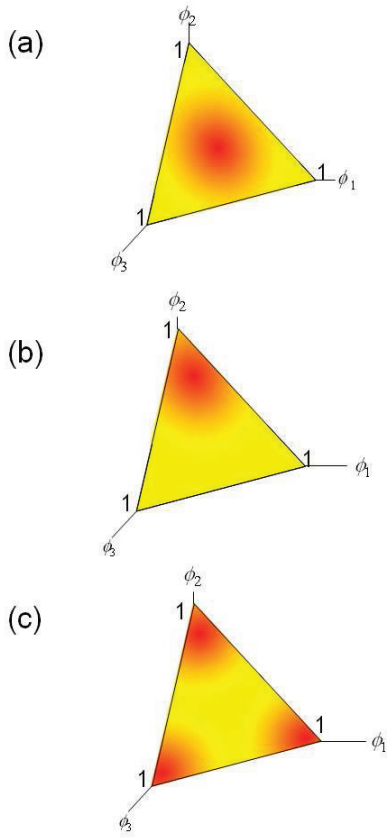


Figure 3: Three example Dirichlet Distributions over the 3-simplex with darker regions showing areas of high probability: (a) Dirichlet(5,5,5), (b) Dirichlet(0.2, 5, 0.2), (c) Dirichlet(0.5,0.5,0.5).

distribution is given by:

$$Dirichlet_k(\alpha_1, \ldots, \alpha_k) = f(x_1, \ldots, x_k; \alpha_1, \ldots, \alpha_k) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} x_i^{a_i - 1}$$

where $\Gamma(x)$ $(= \int_0^\infty t^{(x-1)} e^{-t} dt)$ extends the factorial function to the real numbers. Since a draw from a Dirichlet distribution (written $\beta \sim Dirichlet_k(\boldsymbol{\alpha})$) gives a distribution, a Dirichlet can be used as the prior for a Bayesian finite mixture model:

$$\beta \sim Dirichlet_k(\alpha_1, \ldots, \alpha_k)$$

β is a distribution over the k components $\boldsymbol{\phi}$ of the finite mixture model. Each component ϕ_{z_i} is drawn from a base measure G_0 ($\phi_{z_i} \sim G_0$). The choice of distribution G_0 depends on the nature of the data to be clustered; with data that is represented using the *bag of words* model, G_0 must generate distributions over the word vocabulary. Hence the Dirichlet distribution is an appropriate choice in this case:

$$\phi_{z_i} \sim Dirichlet_v(\alpha_1, \ldots, \alpha_v)$$

where v is the size of the vocabulary.

For each data point (utterance) x_i a component ϕ_{z_i} is selected by a draw z_i from the multinomial distribution β:

$$z_i \sim Multinomial_k(\beta)$$

A suitable distribution $F(\phi_{z_i})$ is then used to draw the data point (utterance). In the bag of words model, the multinomial distribution is used to draw the words for each data point x_i:

$$x_i \sim Multinomial_v(\phi_{z_i})$$

A small example will illustrate this generative process. Imagine that there are just two types of utterances with a vocabulary consisting simply of the words A, B and C. A finite Bayesian mixture model in this case would first draw β from a suitable Dirichlet distribution (e.g. $\beta \sim Dirichlet_2(0.5, 1)$) as, for example, is shown in Figure 4(a). Next the two components ϕ_{z_1} and ϕ_{z_2} would be drawn from a suitable base distribution G_0 (e.g. $\phi_{z_1} \sim Dirichlet_3(1, 0.5, 0.5)$ and $\phi_{z_2} \sim Dirichlet_3(0.5, 0.5, 1)$, see Figure 4(b) and 4(c)). In this case, ϕ_{z_1} will tend to generate

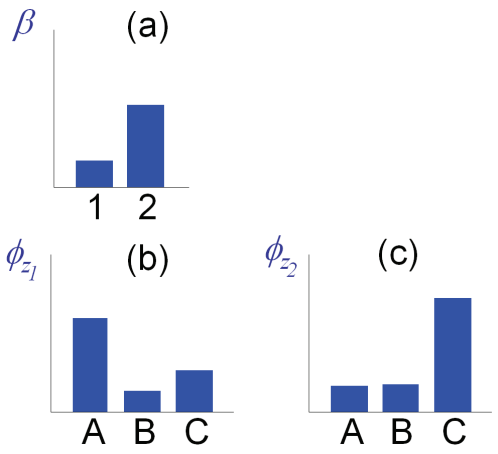


Figure 4: An example finite Bayesian mixture model. (a) The prior distribution over components ϕ_{z_1} (b) and ϕ_{z_2} (c)

utterances containing more occurrences of word A than B or C, whilst ϕ_{z_2} will tend to generate utterances with more C's than A's or B's. A component z_i is then selected for each utterance ($z_i \sim Multinomial_k(\beta)$). Note that in this example, the distribution β would lead to more utterances generated by ϕ_{z_2} than by ϕ_{z_1}. Suppose that five utterances are to be generated by this model and that the components for each utterance are $z_1 = 1$, $z_2 = 2$, $z_3 = 2$, $z_4 = 1$ and $z_5 = 2$. The words in each utterance are then generated by repeated draws from the corresponding component (e.g. $x_1 = ACAAB$, $x_2 = ACCBCC$, $x_3 = CCC$, $x_4 = CABAAC$ and $x_5 = ACC$).

3.2 Dirichlet Processes

A Dirichlet Process can be thought of as an extension of a Dirichlet distribution where the dimensions of the distribution are infinite. The problem with the infinite dimension Dirichlet distribution, though, is that its probability mass would be distributed across the whole of the distribution. However, in most practical applications of mixture modelling there will be a finite number of clusters. The solution is to have a process which will tend to place most of the probability mass at the beginning of the infinite distribution, thereby making it possible to assign probabilities to clusters without restricting the number of clusters available. The GEM *stick breaking* construction (the name comes from the first letters of Griffiths, Engen and McCloskey (Pitman, 2002)) achieves precisely this (Pitman and Yor, 1997). Starting with

a stick of unit length, random portions β'_k are repeatedly broken off the stick, with each part that is broken off representing the proportion of probability assigned to a component:

$$\beta'_k \sim Beta(1, \alpha) \qquad \beta_k = \prod_{i+1}^{k-1} (1 - \beta'_i) \cdot \beta'_k$$

The Dirichlet Process mixture model can now be specified as:

$$\beta \sim GEM(\boldsymbol{\alpha}) \qquad \phi_{z_i} \sim G_0 \quad z_i \in (1 \ldots \infty)$$
$$z_i \sim Multinomial(\beta) \quad x_i \sim F(\phi_{z_i})$$

3.3 Chinese Restaurant Process

The Chinese Restaurant Process (CRP) is a popular Dirichlet Process paradigm that has been successfully applied to many clustering problems. In the CRP, one is asked to imagine a Chinese restaurant with an infinite number of tables. The customers enter the restaurant and select, according to a given distribution, a table at which to sit. All the customers on the same table share the same dish. In this paradigm, the tables represent data clusters, the customers represent data points (x_i) and the dishes represent components (ϕ_z). As each customer (data point) enters the restaurant the choice of which table (cluster) and therefore which dish (component) is determined by a draw from the following distribution:

$$\phi_i | \phi_1, \ldots, \phi_{i-1} \sim \frac{1}{(\alpha + i - 1)} \left(\sum_{j=1}^{i-1} \delta_{\phi_j} + \alpha G_0 \right)$$

where α is the concentration parameter for the CRP. The summation over the δ_{ϕ_j}'s counts the number of customers sat at each of the occupied tables. The probability of sitting at an already occupied table, therefore, is proportional to the number of customers already sat at the table, whilst the probability of starting a new table is proportional to αG_0. Figure 5 illustrates four iterations of this initial clustering process.

Once all the customers (data points) have been placed at tables (clusters), the inference process begins. The posterior $p(\boldsymbol{\beta}, \boldsymbol{\phi}, \boldsymbol{z} | \boldsymbol{x})$ cannot be calculated exactly, but Gibbs sampling can be used. Gibbs sampling for the CRP involves iteratively removing a randomly selected customer from their table, calculating the posterior probability distribution across all the occupied tables together with a potential new table (with a randomly drawn dish,

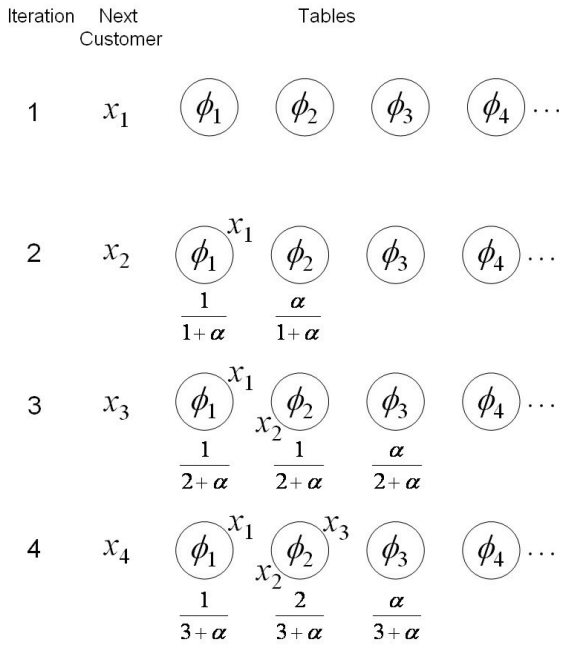


Figure 5: The first four steps of the initial clustering process of the CRP. The probability distribution over the tables is also shown in each case.

i.e. component), and making a draw from that distribution to determine the new table for that customer. The posterior distribution across the tables is calculated as follows:

$$\phi_i | \phi_1, \ldots, \phi_{i-1}, \boldsymbol{x} \sim \frac{1}{B} \left(\sum_{j=1}^{i-1} \delta_{\phi_j} p(x_i | \phi_j) + \alpha G_0 p(x_i | \phi_i) \right)$$

where $B = \alpha p(x_k) + \sum_{j=1}^{i-1} p(x_i | \phi_i)$ is the normalising constant. After a predetermined number of samples, the dish (component) of each occupied table is updated to further resemble the customers (data points) sitting around it. In the *bag of words* approach used here, this involves converting the histogram of word counts in each customer (utterance) sitting at the table into an empirical distribution $\mathcal{H}(x_i)$, taking the average of these empirical distributions and modifying the dish (component) to further resemble this distribution:

$$\phi_i = \phi_i + \frac{\mu}{m_i} \sum_{j=1}^{m_i} \mathcal{H}(x_j)$$

where μ $(0 \leq \mu < 1)$ is the learning constant and m_i is the number of customers around

table i. The inference process continues to iterate between Gibbs sampling and updating the table dishes (components) until the process converges. Convergence can be estimated by observing n consecutive samples in which the customer was returned to the same table they were taken from.

4 Results

The CRP with Gibbs sampling was used to cluster both user and system utterances from the 900 dialogues in the Dihana corpus. Each utterance is treated as an independent bag of words where all information about the dialogue that it came from and the context in which it was uttered is ignored during training. Intra-cluster and inter-cluster similarity measures were used to evaluate the resulting clusters. Intra-cluster similarity S'_i is calculated by averaging the Euclidean distance between every pair of data points in the cluster i:

$$S'_i = \frac{1}{2m_i} \sum_{i=1;j=1}^{m_i} |x_i - x_j|$$

Inter-cluster similarity S'' is calculated by summing the Euclidean distance between the centroids of all pairs of clusters:

$$S'' = \sum_{i=1;j=1}^{n} |C_i - C_j|$$

where C_i is the centroid of cluster i and n is the number of clusters.

Two classification error measures were also used, one from the cluster (table) perspective E', and the other from the perspective of the Dialogue Act (DA) annotations (first level) of the Dihana corpus E''. The cluster classification error of table i is calculated by summing up the occurrences of each DA on the table, finding the DA with the largest total and allocating that DA as the correct classification for that table D_i. The number of false positives f_i^p for that table is the count of all customers (utterances) with DA annotations not in D_i. The number of false negatives f_i^n is the count of utterances with label D_i that occur on other tables. The cluster classification error for table i is therefore:

$$E'_i = \frac{1}{n}(f_i^p + f_i^n)$$

The DA classification error E''_i measures how well DA i has been clustered, using the size of the DA class N_i^c, the size of the largest cluster of utterances from that DA class M_i^c, and the total number of utterances n in the corpus:

$$E''_i = \frac{1}{n}(N_i^c - M_i^c)$$

Cluster No.	Ans	Ask	Clo	Not	Rej	Und
1	1					5
4	2	91				2
9		2	1			9
12	7	161	1			1
13	273	26				8
14	**382**	12	1			5
15	6	1	**909**	1	**327**	22
17	47	39			1	1
18	73				1	3
19		1				4
20	131	115	1		3	1
22	270	29	3			3
23	135	8			2	2
25	83	31	1			4
28	247	16	1			4
29	349	6	1			12
33	13	3	5	1	4	**25**
41	202	45	1		2	3
46		4				1
49	6	251	1	**2**		4
51	124	**896**	1			12
53	45	477				10

Table 2: Clusters of user utterances, with the counts for each level 1 speech act. The largest cluster for each speech act is in bold. The abbreviations are: Und = Undefined, Ans = Answering, Ask = Asking, Clo = Closing, Rej = Rejection, Not = Not-understood.

Table 6 summarises the results from three separate runs of the CRP, each increasing in number of epochs. It should be noted here that the Dihana corpus has 72 DA categories, so the ideal number of clusters discovered by the CRP would be 72. It should also be noted that given an initial random clustering, a good clustering algorithm will reduce intra-cluster similarity ($\bar{S}'$), increase inter-cluster similarity (S'') and reduce the classification errors ($\bar{E}'$ and $\bar{E}''$).

Epochs (K)	No. Clusters	$\bar{S}'$	S''	$\bar{E}'$	$\bar{E}''$
0	70	99703.6	243.74	0.05303	0.00979
1000	44	14975.4	217.56	0.01711	0.00385
1500	54	10093.7	336.15	0.01751	0.00435

Figure 6: The results from three separate runs of the CRP on utterances from the Dihana corpus. Cluster similarity measures and classification error values are shown after 0 (i.e. random clustering), 1000K, and 1500K epochs. $\bar{S}'$, $\bar{E}'$ and $\bar{E}''$ are averaged values.

Level 1	Level 2	Cluster No.
Answering	Day	14
	Destination	22
	Fare	29
	Departure-hour	28, 41
Asking	Departure-hour,Fare	4
	Train-type	12
	Fare	49
	Departure-hour	51, 53

Table 3: Clusters that have specialised on level 1 and level 2 annotations.

5 Discussion

The first row of the table in Figure 6 shows the cluster similarity measures and classification errors after 0 epochs of the inference procedure (i.e. for a random clustering of utterances). This gives a baseline for the measures and error values used in subsequent runs. The second row of values shows the results after a run of 1000K epochs of the inference procedure. This run finds only 44 clusters but has a much lower value for $\bar{S}'$ than was found in the random clustering, showing a significant increase in the similarity between utterances within each cluster. Surprisingly, the value for S'' is also reduced, showing that the differentiation between the clusters formed at this stage is even lower than there was with the random clustering. $\bar{E}'$ and $\bar{E}''$ show suitable reductions indicating that the classification errors are being reduced by the inference process. The third row of values show that after 1500K epochs 54 clusters have been found, intra-cluster similarity is increased beyond that for the random clustering, but the classification errors remain essentially the same as for the 1500K run.

Although the 1500K epoch run found only 54 clusters, it was able to clearly distinguish between system and user utterances: with 30 clusters containing system utterances only, 22 clusters containing user utterances only and 2 clusters containing instances of both. Given that the system utterances in the Dihana corpus are generated from a restricted set of sentences, it is not surprising that these were easy to cluster and differentiate from user utterances. However, the CRP was also able to cluster user utterances well, which is more of a challenge. Table 2 shows the clusters that have specialised on user utterances, with the counts of the level 1 annotations in each case. The largest cluster for each level 1 annotation is shown in bold typeface. From here it can be seen that cluster 15 has specialised on both *Closing* and *Rejection*. It is not surprising that these fall within the same cluster since the words used in each are often the same (e.g. "No thank you" can act as either a closing statement or a rejection statement). Clusters 14, 22, 29, 28 and 41 have specialised to the Answering annotation, whilst clusters 4, 12 49, 51 and 53 have specialised to Asking. Table 3 shows how each of these clusters have specialised to level 2 annotations. Cluster 14, for example, specialises on the Answering:Day pair, whilst 22 specialises on Answering:Destination pair.

These initial results show that, at least for the Dihana corpus, the DPMM can successfully cluster utterances into Speaker, Level 1, and Level2 classes. Whilst this looks promising, it must be acknowledged that the Dihana corpus is restricted to train service inquiries and it remains unclear whether this approach will generalise to other dialogue corpora with a broader range of topics and wider vocabularies. Future work will include investigating the use of ngrams of words, syntactic features, the DAs of previous utterances and experimentation with other corpora such as Switchboard (Godfrey et al., 1992).

Acknowledgments

This work was funded by the Companions project (www.companions-project.org) sponsored by the

European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-034434. We thank Jeff Bilmes (University of Washington) for many very helpful discussions about Dirichlet processes and their application.

References

Toine Andernach, Mannes Poel, and Etto Salomons. 1997. Finding classes of dialogue utterances with kohonen networks. In *In Daelemans*, pages 85–94.

J.A. Andernach. 1996. A machine learning approach to the classification and prediction of dialogue utterances. In *Proceedings of the 2nd International Conference on New Methods in Language Processing*, pages 98–109.

Charles E. Antoniak. 1974. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174.

J.L. Austin. 1962. *How to do things with words*. Oxford: Clarendon Press.

Michael D. Escobar and Mike West. 1995. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

Sheila Garfield and Stefan Wermter. 2006. Call classification using recurrent neural networks, support vector machines and finite state automata. *Knowl. Inf. Syst.*, 9(2):131–156.

J. J. Godfrey, E. C. Holliman, and J. Mcdaniel. 1992. SWITCHBOARD: telephone speech corpus for research and development. In *Proc. ICASSP*, volume 1, pages 517–520 vol.1.

Gang Ji and J. Bilmes. 2005. Dialog act tagging using graphical models. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 1, pages 33–36.

J.M.Benedí, E.Lleida, A. Varona, M.J.Castro, I.Galiano, R.Justo, I. López, and A. Miguel. 2006. Design and acquisition of a telephone spontaneous speech dialogue corpus in spanish: Dihana. In *Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1636–1639, Genova, Italy, May.

S. Keizer, M. Gasic, F. Mairesse, B. Thomson, K. Yu, and S. Young. 2008. Modelling user behaviour in the his-pomdp dialogue manager. In *IEEE SLT*, pages 121–124, Dec.

Steven N. Maceachern and Peter Müller. 1998. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):223–238.

C. D. Martínez-Hinarejos, J. M. Benedí, and R. Granell. 2008. Statistical framework for a spanish spoken dialogue corpus. *Speech Communication*, 50:992–1008.

Geoffrey Mclachlan and David Peel. 2000. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience, October.

Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore. 2006. Incorporating speaker and discourse features into speech summarization. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 367–374, Morristown, NJ, USA. Association for Computational Linguistics.

J. Pitman and M. Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900.

J. Pitman. 2002. Combinatorial stochastic processes.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, March.

Andreas Stolcke, Noah Coccaro, Rebecca Bates, Paul Taylor, Carol Van Ess-Dykema, Klaus Ries, Elizabeth Shriberg, Daniel Jurafsky, Rachel Martin, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Comput. Linguist.*, 26(3):339–373.

E. Zovato and J. Romportl. 2008. Speech synthesis and emotions: a compromise between flexibility and believability. In *Proceedings of Fourth International Workshop on Human-Computer Conversation*, Bellagio, Italy.

A Handsome Set of Metrics to Measure Utterance Classification Performance in Spoken Dialog Systems

David Suendermann, Jackson Liscombe, Krishna Dayanidhi, Roberto Pieraccini*

SpeechCycle Labs, New York, USA

{david, jackson, krishna, roberto}@speechcycle.com

Abstract

We present a set of metrics describing classification performance for individual contexts of a spoken dialog system as well as for the entire system. We show how these metrics can be used to train and tune system components and how they are related to Caller Experience, a subjective measure describing how well a caller was treated by the dialog system.

1 Introduction

Most of the speech recognition contexts in commercial spoken dialog systems aim at mapping the caller input to one out of a set of context-specific semantic classes (Knight et al., 2001). This is done by providing a *grammar* to the speech recognizer at a given recognition context. A grammar serves two purposes:

- It constraints the lexical content the recognizer is able to recognize in this context (the language model) and
- It assigns one out of a set of possible classes to the recognition hypothesis (the classifier).

This basic concept is independent of the nature of a grammar: it can be a rule-based one, manually or automatically generated; it can comprise a statistical language model and a classifier; it can consist of sets of grammars, language models, or classifiers; or it can be a holistic grammar, i.e., a statistical model combining a language model and a classification model in one large search tree.

Most commercial dialog systems utilize grammars that return a semantic parse in one of these contexts:

- directed dialogs (e.g., yes/no contexts, menus with several choices, collection of information out of a restricted set [*Which type of modem do you have?*]—usually, less than 50 classes)
- open-ended prompts (e.g. for call routing, problem capture; likewise to collect information out of a restricted set [*Tell me what you are calling about today*]—possibly several hundred classes (Gorin et al., 1997; Boye and Wiren, 2007))
- information collection out of a huge (or infinite) set of classes (e.g., collection of phone numbers, dates, names, etc.)

When the performance of spoken dialog systems is to be measured, there is a multitude of objective metrics to do so, many of which feature major disadvantages. Examples include

- **Completion rate** is calculated as the number of completed calls divided by the total number of calls. The main disadvantage of this metric is that it is influenced by many factors out of the system's control, such as caller hang-ups, opt-outs, or call reasons that fall out of the system's scope. Furthermore, there are several system characteristics that impact this metric, such as recognition performance, dialog design, technical stability, availability of back-end integration, etc. As experience shows, all of these factors can have unpredictable influence on the completion rate. On the one hand, a simple wording change in the introduction prompt of a system can make this rate improve significantly, whereas, on the other hand, major improvement of the open-ended speech recognition grammar following this very prompt may not have any impact.
- **Average holding time** is a common term for the average call duration. This metric is often considered to be quite controversial since it is unclear whether longer calls are preferred or dispreferred. Consider the following two incongruous behaviors resulting in longer call duration:
  - The system fails to appropriately treat callers, asking too many questions, performing redundant operations, acting unintelligently because of missing back-end integration, or letting the caller wait in never-ending wait music loops.
  - The system is so well-designed that it engages callers to interact with the system longer.

*Patent pending.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 349–356,
Queen Mary University of London, September 2009.

- **Hang-up and opt-out rates.** These metrics try to encapsulate how many callers choose not to use the dialog system, either because they hang up or because they request to speak with a human operator. However, it is unclear how such events are related to dialog system performance. Certainly, many callers may have a prejudice against speaking with automated systems and may hang up or request a human regardless of how well-performing the dialog system is with cooperative users. Furthermore, callers who hang up may do so because they are unable to get their problem solved or they may hang up precisely because their problem was solved (instead of waiting for the more felicitous post-problem-solving dialog modules).

- **Retry rate** is calculated as the average number of times that the system has to re-prompt for caller input because the caller's previous utterance was determined to be Out-of-Grammar. The intuition behind this metric is that the lower the retry rate, the better the system. However, this metric is problematic because it is tied to grammar performance itself. Consider a well-performing grammar that correctly accepts In-Grammar utterances and rejects Out-of-Grammar utterances. This grammar will cause the system to produce retries for all Out-of-Grammar utterances. Consider a poorly designed grammar that accepts everything (incorrectly), even background noise. This grammar would decrease the retry rate but would not be indicative of a well-performing dialog system.

As opposed to these objective measures, there is a subjective measure directly related to the system performance as perceived by the user:

- **Caller Experience**. This metric is used to describe how well the caller is treated by the system according to its design. Caller Experience is measured on a scale between 1 (bad) and 5 (excellent). This is the only subjective measure in this list and is usually estimated based on averaging scores given by multiple voice user interface experts which listen to multiple full calls. Although this metric directly represents the ultimate design goal for spoken dialog systems—i.e., to achieve highest possible user experience—it is very expensive to be repeatedly produced and not suitable to be generated on-the-fly.

  Our former research has suggested, however, that it may be possible to automatically estimate Caller Experience based on several objective measures (Evanini et al., 2008). These measures include the overall number of no-matches and substitutions in a call, operator requests, hang-ups, non-heard speech, the fact whether the call reason could be successfully captured and whether the call reason was finally satisfied. Initial experiments showed a near-human accuracy of the automatic predictor trained on several hundred calls with available manual Caller Experience scores. The most powerful objective metric turned out to be the overall number of no-matches and substitutions, indicating a high correlation between the latter and Caller Experience.

No-matches and substitutions are objective metrics defined in the scope of semantic classification of caller utterances. They are part of a larger set of semantic classification metrics which we systematically demonstrate in Section 2. The remainder of the paper examines three case studies exploring the usefulness and interplay of different evaluation metrics, including:

- the correlation between True Total (one of the introduced metrics) and Caller Experience in Section 3,

- the estimation of speech recognition and classification parameters based on True Total and True Confirm Total (another metric) in Section 4, and

- the tuning of large-scale spoken dialog systems to maximize True Total and its effect on Caller Experience in Section 5.

2 Metrics for Utterance Classification

Acoustic events processed by spoken dialog systems are usually split into two main categories: In-Grammar and Out-of-Grammar. In-Grammar utterances are all those that belong to one of the semantic classes processable by the system logic in the given context. Out-of-Grammar utterances comprise all remaining events, such as utterances whose meanings are not handled by the grammar or when the input is non-speech noise.

Spoken dialog systems usually respond to acoustic events after being processed by the grammar in one of three ways:

- The event gets *rejected*. This is when the system either assumes that the event was Out-of-Grammar, or it is so uncertain about its (In-Grammar) finding that it rejects the utterance. Most often, the callers get re-prompted for their input.

Table 1: *Event Acronyms*

I	In-Grammar
O	Out-of-Grammar
A	Accept
R	Reject
C	Correct
W	Wrong
Y	Confirm
N	Not-Confirm
TA	True Accept
FA	False Accept
TR	True Reject
FR	False Reject
TAC	True Accept Correct
TAW	True Accept Wrong
FRC	False Reject Correct
FRW	False Reject Wrong
FAC	False Accept Confirm
FAA	False Accept Accept
TACC	True Accept Correct Confirm
TACA	True Accept Correct Accept
TAWC	True Accept Wrong Confirm
TAWA	True Accept Wrong Accept
TT	True Total
TCT	True Confirm Total

- The event gets *accepted*. This is when the system is certain to have correctly detected an In-Grammar semantic class.
- The event gets *confirmed*. This is when the system assumes to have correctly detected an In-Grammar class but still is not absolutely certain about it. Consequently, the caller is asked to verify the class. Historically, confirmations are not used in many contexts where they would sound confusing or distracting, for instance in yes/no contexts ("*I am sorry. Did you say NO?*"—"*No!*"—"*This was NO, yes?*"—"*No!!!*").

Based on these categories, an acoustic event and how the system responds to it can be described by four binary questions:

1. Is the event In-Grammar?
2. Is the event accepted?
3. Is the event correctly classified?
4. Is the event confirmed?

Now, we can draw a diagram containing the first two questions as in Table 2. See Table 1 for all

Table 2: *In-Grammar? Accepted?*

	A	R
I	TA	FR
O	FA	TR

Table 3: *In-Grammar? Accepted? Correct?*

	A		R	
	C	W	C	W
I	TAC	TAW	FRC	FRW
O	FA		TR	

acoustic event classification types used in the remainder of this paper.

Extending the diagram to include the third question is only applicable to In-Grammar events since Out-of-Grammar is a single class and, therefore, can only be either falsely accepted or correctly rejected as shown in Table 3.

Further extending the diagram to accomodate the fourth question on whether a recognized class was confirmed is similarly only applicable if an event was accepted, as rejections are never confirmed; see Table 4. Table 5 gives one example for each of the above introduced events for a yes/no grammar.

When the performance of a given recognition context is to be measured, one can collect a certain number of utterances recorded in this context, look at the recognition and application logs to see whether these utterances where accepted or confirmed and which class they were assigned to, transcribe and annotate the utterances for their semantic class and finally count the events and divide them by the total number of utterances. If X is an event from the list in Table 1, we want to refer to x as this average score, e.g., tac is the fraction of total events correctly accepted. One characteristic of these scores is that they sum up to 1 for each of the Diagrams 2 to 4 as for example

$$a + r = 1, \tag{1}$$

$$i + o = 1, \tag{2}$$

$$ta + fr + fa + tr = 1. \tag{3}$$

In order to enable system tuning and to report system performance at-a-glance, the multitude of metrics must be consolidated into a single powerful metric. In the industry, one often uses weights to combine metrics since they are assumed to have different importance. For instance, a False Accept is considered worse than a False Reject since the latter allows for correction in the first retry whereas the former may lead the caller down the

Table 5: *Examples for utterance classification metrics.* This table shows the transcription of an utterance, the semantic class it maps to (if In-Grammar), a binary flag for whether the utterance is In-Grammar, the recognized class (i.e. the grammar output), a flag for whether the recognized class was accepted, a flag for whether the recognized class was correct (i.e. matched the transcription's semantic class), a flag for whether the recognized class was confirmed, and the acronym of the type of event the respective combination results in.

utterance	class	In-Grammar?	rec. class	accepted?	correct?	confirmed?	event
yeah	*YES*	1					I
what		0					O
			NO	1			A
			NO	0			R
no no no	*NO*	1	*NO*		1		C
yes ma'am	*YES*	1	*NO*		0		W
						1	Y
						0	N
i said no	*NO*	1	*YES*	1			TA
oh my god		0	*NO*	1			FA
i can't tell		0	*NO*	0			TR
yes always	*YES*	1	*YES*	0			FR
yes i guess so	*YES*	1	*YES*	1	1		TAC
no i don't think so	*NO*	1	*YES*	1	0		TAW
definitely yes	*YES*	1	*YES*	0	1		FRC
no man	*NO*	1	*YES*	0	0		FRW
sunshine		0	*YES*	1		1	FAC
choices		0	*NO*	1		0	FAA
right	*YES*	1	*YES*	1	1	1	TACC
yup	*YES*	1	*YES*	1	1	0	TACA
this is true	*YES*	1	*NO*	1	0	1	TAWC
no nothing	*NO*	1	*YES*	1	0	0	TAWA

Table 4: *In-Grammar? Accepted? Correct? Confirmed?*

		A		R	
		C	W	C	W
I	Y	TACC	TAWC	FRC	FRW
	N	TACA	TAWA		
O	Y	FAC		TR	
	N	FAA			

wrong path. However, these weights are heavily negotiable and depend on customer, application, and even the recognition context, making it impossible to produce a comprehensive and widely applicable consolidated metric. This is why we propose to split the set of metrics into two groups: good and bad. The sought-for consolidated metric is the sum of all good metrics (hence, an overall accuracy) or, alternatively, the sum of all bad events (overall error rate). In Tables 3 and 4, good metrics are highlighted. Accordingly, we define two consolidated metrics *True Total* and *True Confirm Total* as follows:

$$tt = tac + tr, \tag{4}$$

$$tct = taca + tawc + fac + tr. \tag{5}$$

In the aforementioned special case that a recognition context never confirms, Equation 5 equals Equation 4 since the confirmation terms $tawc$ and fac disappear.

The following sections report on three case studies on the applicability of True Total and True Confirm Total to the tuning of spoken dialog systems and how they relate to Caller Experience.

3 On the Correlation between True Total and Caller Experience

As motivated in Section 1, initial experiments on predicting Caller Experience based on objective metrics indicated that there is a considerable correlation between Caller Experience and semantic

Table 6: *Pearson correlation coefficient for several utterance classification metrics on the source data.*

	A		R
	C	W	
I	0.394	-0.160	-0.230
O	-0.242		-0.155

$$r(\mathrm{TT}) = 0.378$$

classification metrics such as those introduced in Section 2. In the first of our case studies, this effect is to be deeper analyzed and quantified. For this purpose, we selected 446 calls from four different spoken dialog systems of the customer service hotlines of three major cable service providers. The spoken dialog systems comprised

- a call routing application—cf. (Suendermann et al., 2008),
- a cable TV troubleshooting application,
- a broadband Internet troubleshooting application, and
- a Voice-over-IP troubleshooting application—see for instance (Acomb et al., 2007).

The calls were evaluated by voice user interface experts and Caller Experience was rated according to the scale introduced in Section 1. Furthermore, all speech recognition utterances (4480) were transcribed and annotated with their semantic classes. Thereafter, all utterance classification metrics introduced in Section 2 were computed for every call individually by averaging across all utterances of a call. Finally, we applied the Pearson correlation coefficient (Rodgers and Nicewander, 1988) to the source data points to correlate the Caller Experience score of a single call to the metrics of the same call. This was done in Table 6.

Looking at these numbers, whose magnitude is rather low, one may be suspect of the findings. E.g., $|r(\mathrm{FR})| > |r(\mathrm{TAW})|$ suggesting that False Reject has a more negative impact on Caller Experience than True Accept Wrong (aka *Substitution*) which is against common experience. Reasons for the messiness of the results are that

- Caller Experience is subjective and affected by inter- and intra-expert inconsistency. E.g., in a consistency cross-validation test, we observed identical calls rated by one subject as 1 and by another as 5.

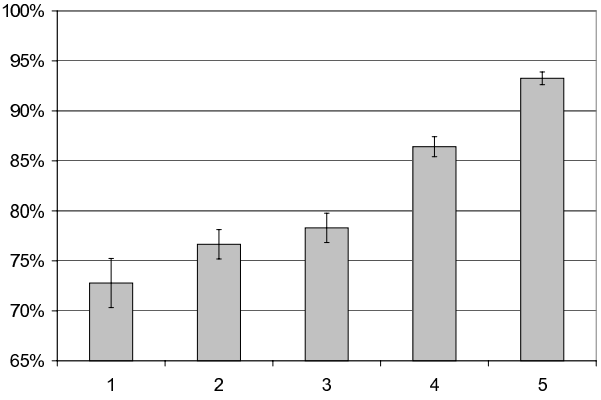


Figure 1: *Dependency between Caller Experience and True Total.*

- Caller Experience scores are discrete, and, hence, can vary by ± 1, even in case of strong consistency.
- Although utterance classification metrics are (almost) objective metrics measuring the percentage of how often certain events happen in average, this average generated for individual calls may not be very meaningful. For instance, a very brief call with a single yes/no utterance correctly classified results in the same True Total score like a series of 50 correct recognitions in a 20-minutes conversation. While the latter is virtually impossible, the former happens rather often and dominates the picture.
- The sample size of the experiment conducted in the present case study (446 calls) is perhaps too small for deep analyses on events rarely happening in the investigated calls.

Trying to overcome these problems, we computed all utterance classification metrics introduced in Section 2, grouping and averaging them for the five distinct values of Caller Experience. As an example, we show the almost linear graph expressing the relationship between True Total and Caller Experience in Figure 1. Applying the Pearson correlation coefficient to this five-point curve yields $r = 0.972$ confirming that what we see is pretty much a straight line. Comparing this value to the coefficients produced by the individual metrics TAC, TAW, FR, FA, and TR as done in Table 7, shows that no other line is as straight as the one produced by True Total supposing its maximization to produce spoken dialog systems with highest level of user experience.

Table 7: *Pearson correlation coefficient for several utterance classification metrics after grouping and averaging.*

	A		R
	C	W	
I	0.969	-0.917	-0.539
O	-0.953		-0.939

$$r(\mathrm{TT}) = \mathbf{0.972}$$

4 Estimating Speech Parameters by Maximizing True Total or True Confirm Total

The previous section tried to shed some light on the relationship between some of the utterance classification metrics and Caller Experience. We saw that, on average, increasing Caller Experience comes with increasing True Total as the almost linear curve of Figure 1 supposes. As a consequence, much of our effort was dedicated to maximizing True Total in diverse scenarios. Speech recognition as well as semantic classification with all their components (such as acoustic, language, and classification models) and parameters (such as acoustic and semantic rejection and confirmation confidence thresholds, time-outs, etc.) was set up and tuned to produce highest possible scores. This section gives two examples of how parameter settings influence True Total.

4.1 Acoustic Confirmation Threshold

When a speech recognizer produces a hypothesis of what has been said, it also returns an acoustic confidence score which the application can utilize to decide whether to reject the utterance, confirm it, or accept it right away. The setting of these thresholds has obviously a large impact on Caller Experience since the application is to reject as few valid utterances as possible, not confirm every single input, but, at the same time, not falsely accept wrong hypotheses. It is also known that these settings can strongly vary from context to context. E.g., in announcements, where no caller input is expected, but, nonetheless utterances like 'agent' or 'help' are supposed to be recognized, rejection must be used much more aggressively than in collection contexts. True Total or True Confirm Total are suitable measures to detect the optimum tradeoff. Figure 2 shows the True Confirm Total graph for a collection context with 30 distinguishable classes. At a confidence value of 0.12, there is a local and global maximum indicating the optimum setting for the confirmation threshold for this grammar context.

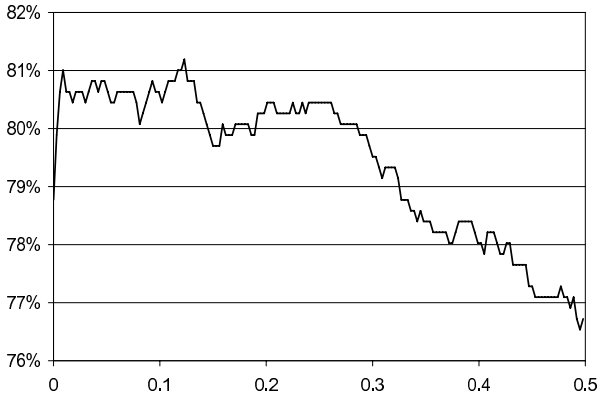


Figure 2: *Tuning the acoustic confirmation threshold.*

4.2 Maximum Speech Time-Out

This parameter influences the maximum time the speech recognizer keeps recognizing once speech has started until it gives up and discards the recognition hypothesis. Maximum speech time-out is primarily used to limit processor load on speech recognition servers and avoid situations in which line noise and other long-lasting events keep the recognizer busy for an unnecessarily long time. As it anecdotally happened to callers that they were interrupted by the dialog system, on the one hand, some voice user interface designers tend to chose rather large values for this time-out setting, e.g., 15 or 20 seconds. On the other hand, very long speech input tends to produce more likely a classification error than shorter ones. Might there be a setting which is optimum from the utterance classification point of view?

To investigate this behavior, we took 115,885 transcribed and annotated utterances collected in the main collection context of a call routing application and aligned them to their utterance dura-

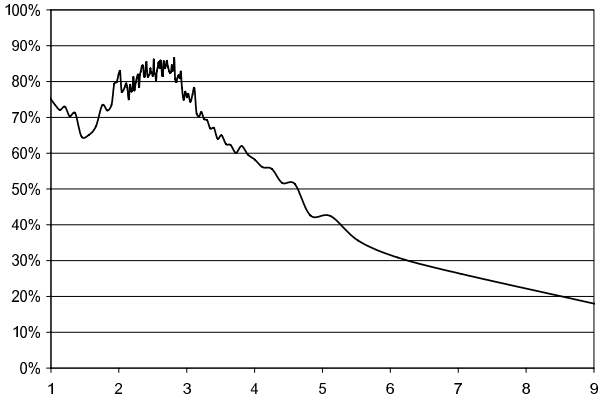


Figure 3: *Dependency between utterance duration and True Total.*

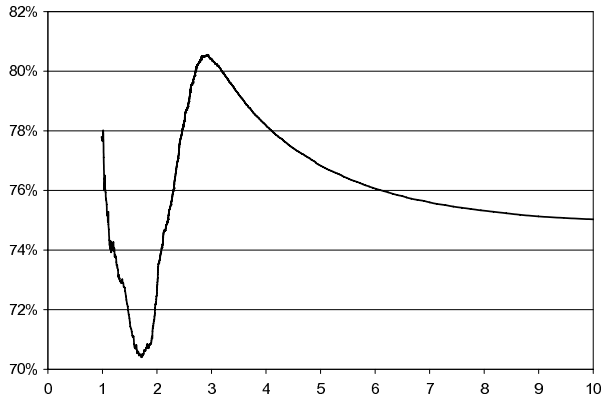


Figure 4: *Dependency between maximum speech time-out and True Total.*

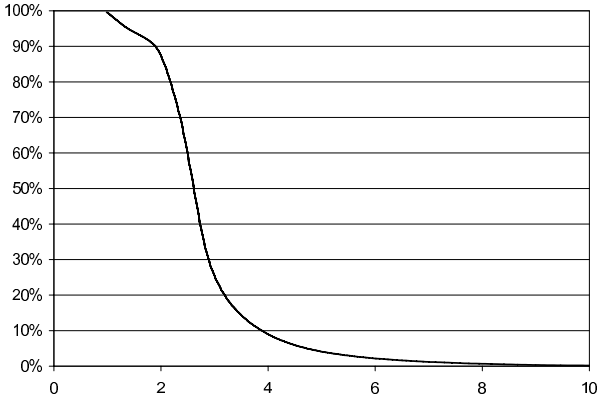


Figure 5: *Percentage of utterances interrupted by maximum speech time-out.*

tions. Then, we ordered the utterances in descending order of their duration, grouped always 1000 successive utterances together, and averaged over duration and True Total. This generated 116 data points showing the relationship between the duration of an utterance and its expected True Total, see Figure 3.

The figure shows a clear maximum somewhere around 2.5 seconds and then descends with increasing duration towards zero. Utterances with a duration of 9 seconds exhibited a very low True Total score (20%). Furthermore, it would appear that one should never allow utterances to exceed four second in this context. However, upon further evaluation of the situation, we also have to consider that long utterances occur much less frequently than short ones. To integrate the frequency distribution into this analysis, we produced another graph that shows the average True Total accumulated over all utterances *shorter than a certain duration*. This simulates the effect of using a different maximum speech time-out setting and is displayed in Figure 4. We also show a graph on how many of the utterances would have been interrupted in Figure 5.

The curve shows an interesting down-up-down trajection which can be explained as follows:

- Acoustic events shorter than 1.0 seconds are mostly noise events which are correctly identified since the speech recognizer could not even build a search tree and returns an empty hypothesis which the classifier, in turn, correctly rejects.

- Utterances with a duration around 1.5s are dominated by single words which cannot properly evaluated by the (trigram) language model. So, the acoustic model takes over the main work and, because of its imperfectness, lowers the True Total.

- Utterances with a moderate number of words are best covered by the language model, so we achieve highest accuracy for them (≈3s).

- The longer the utterances continues after 4 seconds, the less likely the language model and classfier are to have seen such utterances, and True Total declines.

Evaluating the case from the pure classifier performance perspective, the maximum speech time-out would have to be set to a very low value (around 3 seconds). However, at this point, about 20% of the callers would be interrupted. The decision whether this optimimum should be accepcted depends on how elegantly the interruption can be designed:

> "I'm so sorry to interrupt, but I'm having a little trouble getting that. So, let's try this a different way."

5 Continuous Tuning of a Spoken Dialog System to Maximize True Total and Its Effect on Caller Experience

In the last two sections, we investigated the correlation between True Total and Caller Experience and gave examples on how system parameters can be tuned by maximizing True Total. The present section gives a practical example of how rigorous improvement of utterance classification leads to real improvement of Caller Experience.

The application in question is a combination of the four systems listed in Section 3 which work in an interconnected fashion. When callers access the service hotline, they are first asked to briefly describe their call reason. After up to two follow-up questions to further disambiguate their reason, they are either connected to a human operator or one of the three automated troubleshooting systems. Escalation from one of them can connect

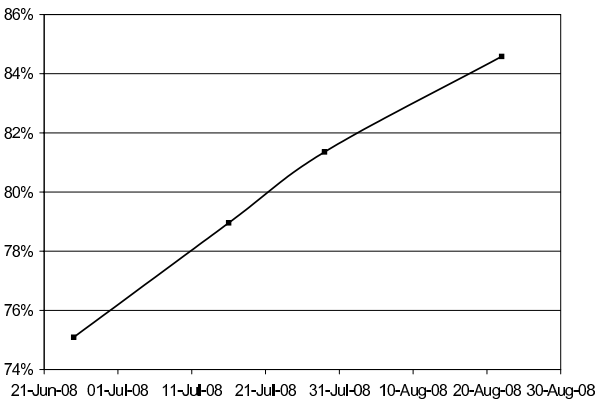


Figure 6: *Increase of the True Total of a large-vocabulary grammar with more than 250 classes over release time.*

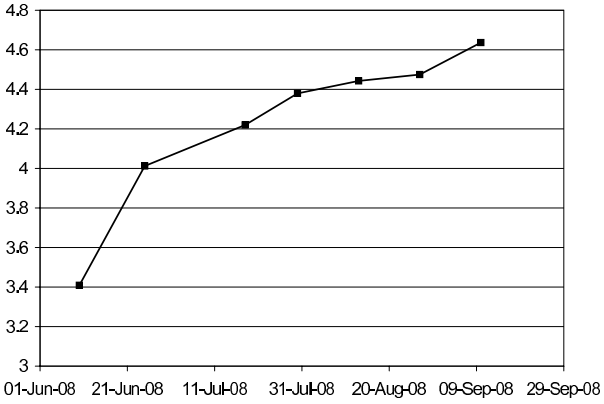


Figure 7: *Increase of Caller Experience over release time.*

the caller to an agent, transfer the caller back to the call router or to one of the other troubleshooting systems.

When the application was launched in June 2008, its True Total averaged 78%. During the following three months, almost 2.2 million utterances were collected, transcribed, and annotated for their semantic classes to train statistical update grammars in a continuously running process (Suendermann et al., 2009). Whenever a grammar significantly outperformed the most recent baseline, it was released and put into production leading to an incremental improvement of performance throughout the application. As an example, Figure 6 shows the True Total increase of the top-level large-vocabulary grammar that distinguishes more than 250 classes. The overall performance of the application went up to more than 90% True Total within three months of its launch.

Having witnessed a significant gain of a spoken dialog system's True Total, we would now like to know to what extent this improvement manifests itself in an increase of Caller Experience. Figure 7 shows that, indeed, Caller Experience was strongly positively affected. Over the same three month period, we achieved an iterative increase from an initial Caller Experience of 3.4 to 4.6.

6 Conclusion

Several of our investigations have suggested a considerable correlation between True Total, an objective utterance classification metric, and Caller Experience, a subjective score of overall system performance usually rated by expert listeners. This observation leads to our main conclusions:

- True Total and several of the other utterance classification metrics introduced in this paper can be used as input to a Caller Experience predictor—as tentative results in (Evanini et al., 2008) confirm.
- Efforts towards improvement of speech recognition in spoken dialog applications should be focused on increasing True Total since this will directly influence Caller Experience.

References

K. Acomb, J. Bloom, K. Dayanidhi, P. Hunter, P. Krogh, E. Levin, and R. Pieraccini. 2007. Technical Support Dialog Systems: Issues, Problems, and Solutions. In *Proc. of the HLT-NAACL*, Rochester, USA.

J. Boye and M. Wiren. 2007. Multi-Slot Semantics for Natural-Language Call Routing Systems. In *Proc. of the HLT-NAACL*, Rochester, USA.

K. Evanini, P. Hunter, J. Liscombe, D. Suendermann, K. Dayanidhi, and R. Pieraccini:. 2008. Caller Experience: A Method for Evaluating Dialog Systems and Its Automatic Prediction. In *Proc. of the SLT*, Goa, India.

A. Gorin, G. Riccardi, and J. Wright. 1997. How May I Help You? *Speech Communication*, 23(1/2).

S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. 2001. Comparing Grammar-Based and Robust Approaches to Speech Understanding: A Case Study. In *Proc. of the Eurospeech*, Aalborg, Denmark.

J. Rodgers and W. Nicewander. 1988. Thirteen Ways to Look at the Correlation Coefficient. *The American Statistician*, 42(1).

D. Suendermann, P. Hunter, and R. Pieraccini. 2008. Call Classification with Hundreds of Classes and Hundred Thousands of Training Utterances ... and No Target Domain Data. In *Proc. of the PIT*, Kloster Irsee, Germany.

D. Suendermann, J. Liscombe, K. Evanini, K. Dayanidhi, and R. Pieraccini. 2009. From Rule-Based to Statistical Grammars: Continuous Improvement of Large-Scale Spoken Dialog Systems. In *Proc. of the ICASSP*, Taipei, Taiwan.

Contrasting the Interaction Structure of an Email and a Telephone Corpus: A Machine Learning Approach to Annotation of Dialogue Function Units

Jun Hu
Department of Computer Science
Columbia University
New York, NY, USA
jh2740@columbia.edu

Rebecca J. Passonneau
CCLS
Columbia University
New York, NY, USA
becky@cs.columbia.edu

Owen Rambow
CCLS
Columbia University
New York, NY, USA
rambow@ccls.columbia.edu

Abstract

We present a dialogue annotation scheme for both spoken and written interaction, and use it in a telephone transaction corpus and an email corpus. We train classifiers, comparing regular SVM and structured SVM against a heuristic baseline. We provide a novel application of structured SVM to predicting relations between instance pairs.

1 Introduction

We present an annotation scheme for verbal interaction which can be applied to corpora that vary across many dimensions: modality of signal (oral, textual), medium (e.g., email, voice alone, voice over electronic channel), register (such as informal conversation versus formal legal interrogation), number of participants, immediacy (online versus offline), and so on.[1] We test it by annotating transcribed phone conversations and email threads. We then use three algorithms, two of which use machine learning (including a novel approach to using Structured SVM), to predict labels and links (a generalization of adjacency pairs) on unseen data. We conclude that we can indeed use a common annotation scheme, and that the email modality is easier to tag for dialogue acts, but that it is harder in email to find the links.

2 Related Work

Annotation for dialogue acts (DAs), inspired by Searle and Austin's work on speech acts, arose largely as a means to understand, evaluate and model human-human and human-machine communication. The need for the enterprise derives from the fact that the relationship between lexicogrammatical form (including mood, e.g., interrogative) and communicative actions cannot be enumerated; there are complex dependencies on the linguistic and situational contexts of use. Many DA schemes exist: they can be hierarchical or flat (Popescu-Belis, 2008), can comprise a large (Devillers et al., 2002; Hardy et al., 2003) or small repertoire (Komatani et al., 2005), or can be oriented towards human-human dialogue (Allen and Core, 1997; Devillers et al., 2002; Thompson et al., 1993; Traum and Heeman, 1996; Stolcke et al., 2000) or multi-party interactions (Galley et al., 2004), or human-computer interaction (Walker and Passonneau, 2001; Hardy et al., 2003), including multimodal ones (Thompson et al., 1993; Kruijff-Korbayová et al., 2006).

A major focus of the cited work is on how to recognize or generate speech acts for interactive systems, or how to classify speech acts for distributional analyses. The focus can be on a specific type of speech act (e.g., grounding and repairs (Traum and Heeman, 1996; Frampton and Lemon, 2008)), or on more general comparisons, such as the contrast between human-human and human-computer dialogues (Doran et al., 2001). While there is a large degree of overlap across schemes, the set of DA types will differ due to differences in the nature of the communicative goals; thus information-seeking versus task-oriented dialogues differ in the set of speech acts and their relative frequencies.

Our motivation in providing a new DA annotation scheme is that our focus differs from much of this prior work. We aim for a relatively abstract annotation scheme in order to make comparisons across interactions of widely differing properties. Our initial focus is less on speech act types and more on the patterns of local alternation be-

[1]This research was supported in part by the National Science Foundation under grants IIS-0745369 and IIS-0713548, and by the Human Language Technology Center of Excellence. 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 sponsors. We would like to thank three anonymous reviewers for their thoughtful comments.

Proceedings of SIGDIAL 2009: the 10th Annual Meeting of the Special Interest Group in Discourse and Dialogue, pages 357–366,
Queen Mary University of London, September 2009.

tween an initiating speech act and a responding one–the analog of adjacency pairs (Sacks et al., 1974). The most closely related effort is (Galley et al., 2004), which aims to automatically identify adjacency pairs in the ICSI Meeting corpus, a large corpus of 75 meetings, using a small tagset. Their maximum entropy ranking approach achieved 90% accuracy on the 4-way classification into agreement, disagreement, backchannel and other. Using the switchboard corpus, (Stolcke et al., 2000) achieved good dialogue act labeling accuracy (71% on manual transcriptions) for a set of 42 dialogue act types, and constructed probabilistic models of dialogue act sequencing in order to test the hypothesis that dialogue act sequence information could boost speech recognition performance.

There has been far less work on developing manual and automatic dialogue act annotation schemes for email. We summarize some salient recent work. Carvalho and Cohen (2006) use word n-grams (with extensive preprocssing) to classify entire emails into a complex ontology of speech acts. However, in their experiments, they concentrate on detecting only a subset of speech acts, which is comparable in size to ours. Speech acts are assigned for entire emails, but several speech acts can be assigned to one email. Apparently, they develop separate binary classifiers for each speech act. Corston-Oliver et al. (2004) are interested in identifying tasks in email. They label each sentence in email with tags from a set which describes the type of content of the sentence (describing a task, scheduling a meeting), but are less interested in the interactive aspect of email communication (creating an obligation to respond).

There has been some work which relates to finding links, but limited to finding question-answer pairs. Shrestha and McKeown (2004) first detect questions using lexical and part-of-speech features, and then find the paragraph that answers the question. They use features related to the structure of the email thread, as well as lexical features. As do we, they find that classifying is easier than linking.

Ding et al. (2008) argue that in order to do well at finding answers to questions, one must also find the context of the question, since it often contains the information needed to identify the answer. They use a corpus of online discussion forums, and use slip-CRFs and two-dimensional CRFs, models related to those we use. We will investigate their proposal to consider the question context in future work.

While they do not use dialogue act tagging to compare modalities, as we do, Murray and Carenini (2008) compare spoken conversation with email by comparing a common summarization architecture across both modalities. They get similar performance, but the features differ.

Table 1: DFU speech act labels

Request-Information (R-I)
Request-Action (R-A)
Inform (Inf)
Commit (Comm)
Conventional (Conv)
Perform (Perf)
Backchannel (Bch) (+/- Grounding)
Other

3 Annotation Scheme

Figure 1: Example DFU illustrating the relation of extent (segmentation) to speech act type

M1.2 I have completed the invoices for April, May and June
M1.3 and we owe Pasadena each month for a total of $3,615,910.62.
M1.4 I am waiting to hear back from Patti on May and June to make sure they are okay with her.
[**Inform(1.2-1.4)**: *status of Pasadena invoicing-completed & pending approval – versus amount due*]
Sflink(1.2-1.4)
M2.1 That's fine.
[**Inform(2.1)**: *acknowledgement of status of Pasadena invoicing*]
Blink(1.2-1.4)

The annotation scheme presented here consists of Dialogue Function Units (DFUs), which are intended to represent abstract units of interaction. The last two authors developed the annotation on three contrasting corpora: email threads, telephone conversations, and court transcripts. It builds on our previous work in intention-based segmentation (Passonneau and Litman, 1997), and on mixing a formal schema with natural language descriptions (Nenkova et al., 2007). In this

paper, we investigate the modalities of telephone two-person conversation in a library setting, and multi-party email in a workplace setting. Our initial focus is on the structure of turn-taking. By using a relatively abstract annotation scheme, we can compare and contrast this behavior across different types of interaction.

Our unit of annotation is the DFU. DFUs have an extent, a dialogue act (DA) label along with a description, and possibly one or more forward and/or backward links. We explain each component of the annotation in turn. We use the example in Figure 1; the example is drawn from actual messages, but has been modified to yield a more succinct example.

The extent of a DFU roughly corresponds to that portion of a turn (conversational turn; email message; etc.) that corresponds to a coherent communicative intention. Because we do not address automatic identification of the segmentation into DFU units in this paper, we do not discuss how annotators are instructed to identify extent.

As illustrated in Figure 1, the communicative function of a DFU is captured by a speech act type, and a natural language description. This is somewhat analogous to the natural language descriptions associated with Summary Content Units (SCUs) in pyramid annotation (Nenkova et al., 2007), or with the intention-based segmentation of (Passonneau and Litman, 1997). The purpose in all cases is to require annotators to articulate briefly but specifically the unifying intention (Passonneau and Litman, 1997), semantic content (Nenkova et al., 2007), or speech act. We use the eight dialogue act types listed in the upper left of Table 1. To accommodate discontinuous speech acts, due to the interruptions that are common to conversation, each speech act can have an operator affix such as "-Continue". We have previously shown (Passonneau and Litman, 1997) that intention-based segmentation can be done reliably by multiple annotators. For twenty narratives each segmented by the same seven annotators, using Cochran's $\mathcal{Q}$ (Cochran, 1950), we found the probabilities associated with the null hypothesis that the observed distributions could have arisen by chance to be at or below p=0.1 $\times 10^{-6}$. Partitioning $\mathcal{Q}$ by number of annotators gave significant results for all values of $\mathcal{A}$ ranging over the number of annotators apart from $\mathcal{A} = 2$. We would expect similar patterns of agreement on DFU segmentation, but have not collected segmentation data from multiple annotators on the two corpora presented here.

DFU Links, or simply Links, correspond to adjacency pairs, but need not be adjacent. A forward link (Flink) is the analog of a "first pair-part" of an adjacency pair (Sacks et al., 1974), and is similarly restricted to specific speech act types. All Request-Information and Request-Action DFUs are assigned Flinks. The responses to such requests are assigned a backward link (Blink). In principle, a response can be any of the speech act types, thus it can be an answer to a question (Inform), a rejection of a Request-Action or a commitment to take the requested action (Commit), a request for clarification (Request-Information), and so on. In most but not all cases, requests are responded to, thus most Flinks and Blinks come in pairs. We refer to Flinks with no matching Blink as dangling links. If an utterance can be interpreted as a response to a preceding DFU, it will get a Blink even where the preceding DFU has no Flink. The preceding DFU taken to be the "first pair-part" of the Link will be assigned a secondary forward link (Sflink). All links except dangling links are annotated with the address of the DFU from which they originate. Figure 1 illustrates an email message (M2) containing a single sentence ("That's fine") that is a response to a DFU in a prior email (M1), where the prior email had no Flink because it only contains Inform DAs; thus M1 gets an Sflink.

4 Corpora

The Loqui corpus consists of 82 transcribed dialogues from a larger set of 175 dialogues that were recorded at New York City's Andrew Heiskell Braille and Talking Book Library during the summer of 2005. All of the transcribed dialogues pertain to one or more book requests. Forty-eight dialogues were annotated; the annotators worked from a combination of the transcription and the audio. Three annotators were trained together, annotated up to a dozen dialogues independently, then discussed, adjudicated and merged ten of them. During this phase, the annotation guidelines were refined and revised. One of the three annotators subsequently annotated 38 additional dialogues.

We also annotated 122 email threads of the Enron email corpus, consisting of email messages in the inboxes and outboxes of Enron corporation

Table 2: Distributional Characteristics of Dialogue Acts in Enron and Loqui

	Loqui		Enron	
Words	21097		17924	
DFUs	3845		1400	
	Speech Act Labels			
Inform	1928	50%	853	61%
Request-Inf.	761	20%	149	11%
Request-Action	39	1%	37	3%
Commit	338	9%	3	0%
Conventional	254	7%	356	25%
Backchannel	507	13%	0	0
Other	18	0%	2	0%
Total	3845	100%	1400	100%
	Links			
Paired Links	1204	63%	193	28%
Flink/Blink	702	58%	83	43%
Sflink/Blink	502	42%	110	57%
Dangling Links	90	2%	97	7%
Mutliple Blinks	4	0%	4	0%
	Links by Speech Act Labels			
Inform	1003	83%	142	74%
Request-Inf.	170	14%	44	23%
Request-Action	1	0%	5	3%
Commit	13	1%	2	1%
Conventional	2	0%	0	0
Backchannel	15	1%	0	0
	1204	100%	193	100%

employees. Most of the emails are concerned with exchanging information, scheduling meetings, and solving problems, but there are also purely social emails. We used a version of the corpus with some missing messages restored from other emails in which they were quoted (Yeh and Harnly, 2006). The annotator of the majority of the Loqui corpus also annotated the Enron corpus. She received additional training and guidance based on our experience with a pilot annotator who helped us develop the initial guidelines.

Table 2 illustrates differences between the two corpora. The DFUs in the Loqui data are much shorter, with 5.5 words on average compared with 12.8 words in Enron. The distribution of DFU labels shows a similarly high proportion of Inform acts, comprising 50% of all Loqui DFUs and 61% of all Enron DFUs. Otherwise, the distributions are quite distinct. The Loqui interactions are all two party telephone dialogues where the callers (library patrons) tend to have limited goals (requesting books). The Enron threads consist of two or more parties, and exhibit a much broader range of communicative goals. In the Loqui data, backchannels are relatively frequent (13%) but do not occur in the email corpus for obvious reasons. There are some Commits (9%), typically reflecting cases where the librarian indicates she will send requested items to the caller by mail, or place them on reserve. There are no Commits in the Enron data. Neither corpus has many Request-Actions; the Loqui corpus has many more requests for information, which includes requests made by the librarian, e.g., for the patrons' identifying information, or by the caller.

The most striking differences between the two corpora pertain to the distribution of DFU Links. In Loqui, 63% of the DFUs are the first pair-part or the second pair-part of a Link compared with 28% in Enron. In Loqui, the majority of Links are initiated by overt requests (58% of Links are Flink/Blink pairs), whereas in Enron, the majority of Links involve SFlinks (57%). There are relatively few dangling Links in either dataset, with more than three times as many in Enron (7% versus 2% in Loqui). Most of the DFU types in the second pair-part of Links are Informs and Request-Information, with a different proportion in each dataset. In Loqui, 83% of DFUs that are second pair-part of a Link are Informs compared with 74% in Enron; correspondingly, only 14% of DFUs in Links are Request-Information in Loqui versus 23% in Enron.

5 Dialogue Act Tagging and Link Prediction

There are two machine learning tasks in our problem. The first is Dialogue Act (DA) Tagging, in which we assign DAs to every Dialogue Functional Unit (DFU). The second is Link prediction, in which we predict if two DFUs form a link pair. In this paper, we assume that the DFUs are given. We propose three systems to tackle the problem. The first system is a non-strawman Baseline Heuristics system, which uses the structural characteristics of dialogue. The second is Regular SVM. The third is Structured SVM. Structured SVM is a discriminative method that can predict complex structured output. Recently, discriminative Probabilistic Graphical Models have been widely applied in structural problems (Getoor and

Taskar, 2007) such as link prediction. However, Structured SVM (Taskar et al., 2003; Tsochantaridis et al., 2005) is also a compelling method which has the potential to handle the interdependence between labeling and sequencing, due to its ability to handle dependencies among features and prediction results within the structure. sequence labeling (Tsochantaridis et al., 2005). We have adapted Structured SVM to our problem, provided a novel method for link prediction, and shown that it is superior in some aspects to Regular SVM.

5.1 Features

We have two sets of features. DFU features are associated with a particular DFU, and link features describe the relationship between two DFUs. DFU features are used in both tasks. Link features are only used in link prediction. The feature vector of a link contains two sets of DFU features and the link features that are defined over the two DFUs. Table 3 gives the features we used, which are almost identical for both corpora, so we could compare the performance.

Because a lot of Flinks are questions, we chose some features that are tailored to Question-Answer detection, such as presence of a question mark. Dialogue fillers and acceptance words affect the accuracy of Part-Of-Speech tagging. On the other hand, they are helpful indicators of disfluency or confirmation. So we hand-picked a list of filler and acceptance words, removed them from the sentence, and added features counting their occurrences.

5.2 Baseline Heuristics

Dialogue Act Tagging We use the most frequent DA as the heuristic for prediction. In both Enron and Loqui, this DA is Inform.

Link Prediction In link prediction, the heuristics for Enron and Loqui corpora are different due to structural differences. In Loqui, whenever we see a DFU with a Forward Link (DA is Request-Information or Request-Action), we predict that the target of the link is the first following DFU that is available and acceptable. "Available" means that the second DFU has not been assigned a Backward Link yet. "Acceptable" means that the second DFU has a DA that is very frequent in a Backward Link and it is of a different speaker to the first DFU. We enforce similar constraints in Enron corpus for link prediction, except that the second DFU not only has to be from a different author, but also has to be in a message which is a direct descendant in the reply chain of the message that contains the first DFU. The baseline link prediction algorithm uses the DAs as predicted by the Regular SVM. If we used the baseline DA prediction, the result would be too low to make a valid comparison against other systems in terms of link prediction because all DAs would be identical.

Table 3: DFU features (E: Enron, L: Loqui)

Structural for DA prediction	
E,L	First three POS
E,L	Relative Position in the Dialogue
E	Existence of Question Mark
E,L	Does the first POS start with "w" or "v"
E,L	Length of the DFU
E	Head, body, tail of the Message
E,L	Dialogue Act (Only used in link prediction)
Lexical for DA prediction	
E,L	Bag of Words
E,L	Number of Content Words
L	Number of Filler Words, as "uh", "hmm"
E,L	Number of Acceptance Words, as "yes"
Structural for Link prediction	
E,L	The distance between two DFUs
Lexical for Link prediction	
E,L	Overlapping number of content words

5.3 Regular SVM

We have used the Yamcha support vector machine package (chasen.org/∼taku/software/yamcha/). The advantage of Yamcha is that it extends the traditional SVM by enabling using dynamically generated features such as preceding labels.

Dialogue Act Tagging We use the feature vector of the current DFU as well as the predicted DA of the preceding DFU as features to predict the DA of the current DFU.

Link Prediction First, in order to limit search space, we specify a certain window size to produce a space $\mathbf{S}$ of DFU pairs under consideration. For a particular DFU, we look at all succeeding DFUs and check if these two DFUs satisfy the following constraint: in Loqui, they must be of different speakers; in Email, one must be another's ancestor and they must be of different authors. We consider all valid pairs starting from the current DFU until

the number of considered valid pairs reaches the window size. Then we proceed to the next DFU and collect more DFU pairs into our consideration space.

Second, we train a link binary classifier with all DFU pairs in this consideration space along with a binary classification correct/not correct as training data. This classifier takes the feature vectors of the two DFUs as well as the link features such as the distance between these two DFUs as features.

Third, we apply a greedy algorithm to generate links in the test data with the binary classifier. The algorithm firstly uses the classifier to generate scores for all DFU pairs in the consideration space of the test data, then it scans the dialogue sequentially, checks all preceding DFUs that are allowed to link to the current DFU (i.e., the DFU pair is in the consideration space), and assigns corresponding links to the most likely DFU pair. We impose a restriction that there can be at most one Flink, one Sflink and one Blink for any given DFU.

5.4 Structured SVM

A Structured SVM is able to predict complex output instead of simply a binary result as in a regular SVM. There are several variants. We have followed the margin-rescaling approach (Tsochantaridis et al., 2005), and implemented our systems using SVM^{python}, which is a python interface to the SVM^{struct} package (svmlight.joachims.org/svm_struct.html). Generally, Structured SVM learns a discriminant function $F : \mathbf{X} \times \mathbf{Y} \rightarrow \mathbf{R}$, which estimates a score of how likely the output $\mathbf{y}$ is given the input $\mathbf{x}$. Crucially, $\mathbf{y}$ can be a complex structure. Section A in the appendix; here, we summarize the main intuitions.

Dialogue Act Tagging The input $\mathbf{x}$ is a sequence of DFUs, and $\mathbf{y}$ is the corresponding sequence of DAs to predict. Compared to Regular SVM, instead of predicting $\mathbf{y}^t$ one at a time, Structured SVM optimizes the sequence as a whole and predicts all labels simultaneously. Due to the similarity to HMM, the maximization problem is solved by the Viterbi algorithm (Tsochantaridis et al., 2005).

Link Prediction The input now contains the DFU sequence, a link consideration space, as well as a label sequence, which we get from the previous stage. The output structure chooses among the possible links in the link consideration space, such that there is at most one Flink/SFlink or Blink for any given DFU, and that there are no crossing links. (Note that all the constraints are only enforced in training and prediction; in testing, we compare results against the complete manual annotations which do not follow these constraints.) Then the maximization problem can be solved by a straightforward dynamic programming algorithm.

Table 4: Result of DA prediction

	Baseline	Regular	Struct
Loqui	50.14%	68.30%	70.26%
Enron	60.93%	88.34%	88.71%

Note: Structured SVM parameters for Loqui are $\mathbf{C} = 300,\ \alpha = 1$; Structured SVM parameters for Enron are $\mathbf{C} = 1000,\ \alpha = 1$.

6 Experiments

We have three hypotheses for our experiments:

Hypothesis 1 Link prediction is harder than Dialogue Act prediction.

Hypothesis 2 Enron is harder than Loqui.

Hypothesis 3 Structured SVM is better than Regular SVM, and Baseline is the worst.

We have applied the algorithm described in Section 5 to both the Enron and Loqui corpora. The data set is annotated with DFUs; we focus on the DA labels and Links. As discussed before, every system is a pipeline that would preprocess the data into separate DFUs, predict the Dialogue Acts, and then feed the Dialogue Acts into the link prediction algorithm. The size of the data set is shown in Table 2. We do five-fold cross-validation.

Table 4 shows the accuracy of three systems on Enron and Loqui. Structured SVM has a clear lead to Regular SVM in Loqui; but the advantage is less clear in Enron. Tables 6 and 7 give detailed results of DA prediction.We do not show DAs that do not exist in the corpora, or that were not predicted by the algorithms. Both Regular SVM and Structured SVM performed consistently for the two corpora.

Table 5 gives Link prediction results. Note that when we compute the combined result for both types of links, we are only concerned with the Link position. The seperate results for Flink/Blink and Sflink/Blink require us to identify the types of links first, so here we not only compare the position of predicted links against the gold, but also require predicted DAs to indicate the link type (e.g., the DA of the first DFU must be Request-

Table 5: Link Prediction for Enron and Loqui

	Baseline			Regular			Struct		
Enron	R	P	F	R	P	F	R	P	R
Paired Links	16.66%	40%	23.52%	18.75%	55.38%	28.01%	31.25%	39.47%	34.88%
Flink/Blink	32.53%	33.75%	33.13%	26.50%	61.11%	36.97%	34.93%	47.54%	40.27%
Sflink/Blink	0.0%	0.0%	0.0%	11.92%	44.82%	18.83%	22.93%	27.47%	25.00%
Loqui									
Paired Links	30%	56.15%	39.11%	43.59%	60.60%	50.71%	44.15%	56.02%	49.38%
Flink/Blink	43.30%	46.47%	44.83%	40.58%	57.73%	47.66%	43.55%	60.04%	50.48%
Sflink/Blink	0.0%	0.0%	0.0%	21.76%	29.36%	25.00%	22.88%	26.24%	24.45%

Note: Structured SVM parameters for Enron are $\mathbf{C} = 2000,\ \beta = 2.$, for Loqui $\mathbf{C} = 1000,\ \beta = 4$.

Information or Request-Action to qualify as a Flink/Blink).

Table 6: Recall/Precision/F-measure of DA prediction for Loqui (in %)

	Regular			Struct		
	P	R	F	P	R	F
R-A	50.0	51.7	50.9	43.3	43.3	43.3
R-I	51.3	61.1	55.8	52.3	71.2	60.3
Inf	73.9	73.0	73.5	76.9	74.1	75.5
Bch	65.3	51.7	57.7	65.1	53.6	58.8
Com	5.6	33.3	9.5	5.6	33.3	9.5
Conv	81.2	84.0	82.6	83.7	83.3	83.5

Table 7: Recall/Precision/F-measure of DA prediction for Enron (in %)

	Regular			Struct		
	R	P	F	R	P	F
R-A	27.8	55.6	37.0	25.0	75.0	37.5
R-I	77.9	82.3	80.0	77.2	83.3	80.1
Inf	92.5	90.6	91.5	92.1	91.2	91.7
Conv	90.5	87.3	88.9	93.4	85.6	89.3

7 Discussion

Hypothesis 1 The result of DA prediction is drastically better than link prediction. There are usually indicators of DA types such as "thank you" for Conventional, so learning algorithms could easily capture them. But in link prediction, we frequently need to handle deep semantic inference and sometimes useful information exists in the surrounding context rather than the DFU itself. Both of these scenarios imply that in order to predict links or relationships better, we need more sophisticated features.

Hypothesis 2 This hypothesis turns out to be half-correct. The DA prediction accuracy for Enron is better than that of Loqui. The higher percentage of Inform and less diversity of DAs in Enron (See Appendix for statistics) may be part of the reason. Another possible explanation is that as a set of spoken dialogue data, Loqui is inherently more difficult to process than written form, since some common tasks such Part-Of-Speech tagging have lower accuracy for spoken data. On the other hand, the result of link prediction did confirm our hypothesis. The first reason is that there are far fewer links in Enron than in Loqui, so we have less training data. The tree structure of the reply chain in the email threads also makes prediction more difficult. And the link distance is longer, because in email, people can respond to a very early message, while in a phone conversation, people tend to respond to immediate requests.

Hypothesis 3 Both SVM models perform better than the baseline. Generally, Structured SVM performs better than Regular SVM, especially in link prediction for Enron. This confirms the advantage of using Structured SVM for output involving inter-dependencies. The only exception is the Sflink prediction in Loqui, which in turn affects the overall accuracy of link prediction.

References

James Allen and Mark Core. 1997. Damsl: Dialogue act markup in several layers. http://www.cs.rochester.edu/research/cisd/resources/damsl.

Vitor Carvalho and William Cohen. 2006. Improving "email speech acts" analysis via n-gram selection. In *Proceedings of the Analyzing Conversations in Text and Speech*.

William G. Cochran. 1950. The comparison of percentages in matched samples. *Biometrika*, 37:256–266.

Simon Corston-Oliver, Eric Ringger, Michael Gamon, and Richard Campbell. 2004. Task-focused summarization of email. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Laurence Devillers, Sophie Rosset, Bonneau-Helene Maynard, and Lamel Lori. 2002. Annotations for dynamic diagnosis of the dialog state. In *LREC*.

Shilin Ding, Gao Cong, Chin-Yew Lin, and Xiaoyan Zhu. 2008. Using conditional random fields to extract contexts and answers of questions from online forums. In *Proceedings of ACL-08: HLT*, Columbus, Ohio.

Christine Doran, John Aberdeen, Laurie Damianos, and Lynette Hirschman. 2001. Comparing several aspects of human-computer and human-human dialogues. In *Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue*.

Matthew Frampton and Oliver Lemon. 2008. Using dialogue acts to learn better repair strategies for spoken dialogue systems. In *ICASSP*.

Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. 2004. Identifying agreement and disagreement in conversational speech: use of Bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 669–676.

Lise Getoor and Ben Taskar, editors. 2007. *Introduction to Statistical Relational Learning*. The MIT Press.

Hilda Hardy, Kirk Baker, Bonneau-Helene Maynard, Laurence Devillers, Sophie Rosset, and Tomek Strzalkowski. 2003. Semantic and dialogic annotation for automated multilingual customer service. In *Eurospeech/Interspeech*.

Kazunori Komatani, Nayouki Kanda, Tetsuya Ogata, and Hiroshi G. Okuno. 2005. Contextual constraints based on dialogue models in database search task for spoken dialogue systems. In *Eurospeech*.

Ivana Kruijff-Korbayová, Tilman Becker, Nate Blaylock, Ciprian Gerstenberger, Michael Kaisser, Peter Poller, Verena Rieser, and Jan Schehl. 2006. The Sammie corpus of multimodal dialogues with an mp3 player. In *LREC*.

Gabriel Murray and Giuseppe Carenini. 2008. Summarizing spoken and written conversations. In *EMNLP*.

Ani Nenkova, Rebecca J. Passonneau, and Kathleen McKeown. 2007. The pyramid method: incorporating human content selection variation in summarization evaluation. *ACM Transactions on Speech and Language Processing*, 4(2).

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1).

Andrei Popescu-Belis. 2008. Dimensionality of dialogue act tagsets: An empirical analysis of large corpora. *LREC*, 42(1).

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systemics for the organization of turn-taking for conversation. *Language*, 50(4).

Lokesh Shrestha and Kathleen McKeown. 2004. Detection of question-answer pairs in email conversations. In *COLING*.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykena, and Meteer Marie. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *International Journal of Computational Linguistics*, 26(3).

Ben Taskar, Crlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *NIPS*.

Henry S. Thompson, Anne H. Anderson, Ellen Gurman Bard, Gwyneth Doherty-Sneddon, Alison Newlands, and Cathy Sotillo. 1993. The HCRC map task corpus: Natural dialogue for speech recognition. In *Proceedings of the DARPA Human Language Technology Workshop*.

David Traum and Peter Heeman. 1996. Utterance units and grounding in spoken dialogue. In *Interspeech/ICSLP*.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *JMLR*, 6.

Marilyn A. Walker and Rebecca Passonneau. 2001. Date: A dialogue act tagging scheme for evaluation of spoken dialogue systems. In *HLT*.

Jen-Yuan Yeh and Aaron Harnly. 2006. Email thread reassembly using similarity matching. In *Conference on Email and Anti-Spam*.

A Appendix: Structured SVM

This section provides mathematical background for Secton 5.4. The hypothesis function is given by:

$$f(\mathbf{x}, \mathbf{w}) = argmax_{\mathbf{y} \in \mathbf{Y}} F(\mathbf{x}, \mathbf{y} : \mathbf{w})$$

And in addition, we assume F to be linear to a joint feature map $\Psi(\mathbf{x}, \mathbf{y})$.

$$F(\mathbf{x}, \mathbf{y} : \mathbf{w}) = \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}) \rangle$$

We also define a loss function $\Delta(\mathbf{y}, \overline{\mathbf{y}})$ which defines the deviation of the predicted output $\overline{\mathbf{y}}$ to the correct output.

As a result, given a sequence of training examples,$(\mathbf{x}_1, \mathbf{y}_1) \cdots (\mathbf{x}_n, \mathbf{y}_n) \in \mathbf{X} \times \mathbf{Y}$, the function we need to optimize becomes:

$$min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + \frac{C}{n} \sum_{i=1}^{n} \xi_i$$

$s.t. \; \forall i \forall \mathbf{y} \in \mathbf{Y} \backslash \mathbf{y}_{(i)} : \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}) \rangle > \Delta(\mathbf{y}_{(i)}, \mathbf{y}) - \xi_\mathbf{i}$ where,

$$\langle \mathbf{w}, \delta\Psi_i(\mathbf{y}) \rangle = \Big\langle \mathbf{w}, \Psi(\mathbf{x}_{(i)}, \mathbf{y}_{(i)}) - \Psi(\mathbf{x}_{(i)}, \mathbf{y}) \Big\rangle$$

$\mathbf{w}$ is optimized towards maximizing the margin between the true structured output $\mathbf{y}$ and any other suboptimal configurations for all training instances.

A cutting plane optimization algorithm is implemented in SVM^{struct}. However, for any problem, we need to implement the feature map $\Psi(\mathbf{x}, \mathbf{y})$, the loss function $\Delta(\mathbf{y}, \overline{\mathbf{y}})$, and a maximization problem which enables the cutting plane optimization, i.e.

$$\overline{\mathbf{y}} = argmax_{\mathbf{y} \in \mathbf{Y}} \Delta(\mathbf{y}_{(i)}, \mathbf{y}) - \langle \mathbf{w}, \delta\psi_i(\mathbf{y}) \rangle$$

Only certain feature maps that would make solving this maximization effectively, usually by dynamic programming, could be handled this way.

For **Dialogue Act Tagging**, let $\mathbf{x} = \left(\mathbf{x}^1, \mathbf{x}^2 \ldots \mathbf{x}^T\right)$ be the sequence of DFUs, and $\mathbf{y} = \left(\mathbf{y}^1, \mathbf{y}^2 \ldots \mathbf{y}^T\right)$ the corresponding sequence of dialogue acts. $\phi(\mathbf{x}^t)$ represents the DFU features and $\phi(\mathbf{x}^t) \in R^D$. $\mathbf{y}^t \in L = \{l_1, \ldots, l_K\}$ where L contains the set of available DAs. The feature map is (Tsochantaridis et al., 2005):

$$\mathbf{\Psi}(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} \sum_{t=1}^{T} \phi(\mathbf{x}^t) \otimes \mathbf{\Lambda}(\mathbf{y}^t) \\ \mathbf{\Lambda}(\mathbf{y}^{t-1}) \otimes \mathbf{\Lambda}(\mathbf{y}^t) \end{pmatrix}$$

where $\mathbf{\Lambda}(\mathbf{y}^t) = [\lambda(l_1, \mathbf{y}), \ldots, \lambda(l_k, \mathbf{y})]$ and λ is an indicator function that returns 1 if two parameters are equal. $\otimes$-operator is defined as:

$$\mathbf{R}^D \times \mathbf{R}^K \rightarrow \mathbf{R}^{D \cdot K}, \;\; (\mathbf{a} \otimes \mathbf{b})_{i+(j-1)D} \equiv \mathbf{a}_i \cdot \mathbf{b}_j$$

In analogy to an HMM, the lower part in $\mathbf{\Psi}(\mathbf{x}, \mathbf{y})$ encodes the histogram of adjacent DA transitions in $\mathbf{y}$; the upper part encodes the DA emissions from a specific label to one dimension in the DFU feature vector. Hence, the total number of dimensions in $\mathbf{\Psi}(\mathbf{x}, \mathbf{y})$ is $K^2 + DK$. As a result, $F(\mathbf{x}, \mathbf{y} : \mathbf{w}) = \langle \mathbf{w}, \mathbf{\Psi}(\mathbf{x}, \mathbf{y}) \rangle$ gives a global score based on all transitions and emissions in the sequence, which captures the dependecies among nearby labels and mimics the behaviour of an HMM. Figure 2 gives an example of how to compute the feature map.

The loss function is the sum of all zero-one losses across the sequence, i.e.

$$\Delta(\mathbf{y}, \overline{\mathbf{y}}) = \alpha \sum_{t=1}^{T} \lambda(\mathbf{y}^t, \overline{\mathbf{y}}^t)$$

α denotes a cost assigned to every DA loss.

For **Link Prediction**, the input contains the DFU sequence $\mathbf{x}$, a link consideration space $s = \{(i, j) :, \text{DFU } i \text{ and } j \text{ is being considered}\}$, as well as label sequence $\mathbf{y}$ which we get from the previous stage. $\varphi(\mathbf{x}^i, \mathbf{x}^j)$ is the link feature defined over two DFUs. Let the dimension of link feature be B. The output structure $\mathbf{u} = \left\{\mathbf{u}^1, \mathbf{u}^2 \ldots \mathbf{u}^T\right\}$ specifies the link plan. $\mathbf{u}^t$ denotes that there is a link from DFU $t - \mathbf{u}^t$ to t with the exception that $\mathbf{u}^t$ being zero denotes there is no link pointing to t. The setup of u constraints that there can be at most one Flink/SFlink or Blink for any given DFU. In addition $\mathbf{u}$ is also subject to the constraint that all specified links must be in the link consideration space.

The discriminant function becomes $F : \mathbf{X} \times \mathbf{Y} \times \mathbf{S} \times \mathbf{U} \rightarrow \mathbf{R}$. Similar to structured DA prediction, the discriminant function should give a global evaluation as to how likely is the link plan specified by $\mathbf{U}$ with respect to all the input vectors. Our solution is to decompose the score, and correspondingly, the feature representation into two components, link emission and no-link emission; the details can be found in Figure 3 in the appendix and an example is in Figure 2.

Similarly, we could define the loss function as the sum of all zero-one losses across the sequence , i.e.

$$\Delta(\mathbf{u}, \overline{\mathbf{u}}) = \beta \sum_{t=1}^{T} \lambda(\mathbf{u}^t, \overline{\mathbf{u}}^t)$$

β denotes a cost assigned to every Link loss.

Figure 2: A full example of feature map for Structured SVM

$\mathbf{x}^1$ = "are you you sure"
$\mathbf{x}^2$ = "sure"

$\mathbf{y}^1$ = "Req-Info"
$\mathbf{y}^2$ = "Inform"

$\mathbf{u}^1 = 0$
$\mathbf{u}^2 = 1$

$\phi(\mathbf{x}^1) = (1, 2, 1)$
$\phi(\mathbf{x}^2) = (0, 0, 1)$

$\varphi(\mathbf{x}^1, \mathbf{x}^2) = (1, 1)$

$$\mathbf{\Psi}_{da} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ \\ 0 \\ 0 \\ 1 \\ \\ 1 \\ 2 \\ 1 \end{pmatrix} \begin{matrix} \text{Inform to Inform} \\ \text{Inform to Req-Info} \\ \text{Req-Info to Inform} \\ \text{Req-Info to Inform} \\ \\ \text{Inform with "are"} \\ \text{Inform with "you"} \\ \text{Inform with "sure"} \\ \\ \text{Req-Info with "are"} \\ \text{Req-Info with "you"} \\ \text{Req-Info with "sure"} \end{matrix}$$

$$\mathbf{\Psi}_{link} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 0 \\ 1 \\ \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ \\ 1 \\ 1 \\ \\ 1 \\ 2 \\ 1 \\ 0 \\ 1 \end{pmatrix} \begin{matrix} \text{1st link pair-part with"are"} \\ \text{1st link pair-part with"you"} \\ \text{1st link pair-part with"sure"} \\ \text{1st link pair-part with Inform} \\ \text{1st link pair-part with Req-Info} \\ \\ \text{2nd link pair-part with"are"} \\ \text{2nd link pair-part with"you"} \\ \text{2nd link pair-part with"sure"} \\ \text{2nd link pair-part with Inform} \\ \text{2nd link pair-part with Req-Info} \\ \\ \text{distance of link} \\ \text{overlap of link} \\ \\ \text{No-Link with"are"} \\ \text{No-Link with"you"} \\ \text{No-Link with"sure"} \\ \text{No-Link with Inform} \\ \text{No-Link with Req-Info} \end{matrix}$$

Note: In this example, $\phi(\mathbf{x}^t)$ extracts the bag-of-words features from $\mathbf{x}^t$. "are","you","sure" are the 1st, 2nd and 3rd DFU feature respectively. $\varphi(\mathbf{x}^i, \mathbf{x}^j)$ extracts the distance and number of the overlap content, which are the link features, from the 1st and 2nd pair-part in a DFU link pair. There is a link from DFU 1 to DFU 2 as specified by $j - u^j = i$, but there is no link pointing to DFU 1.

Figure 3: The feature map of link prediction for the structured SVM

$$\mathbf{\Psi}_L = \begin{pmatrix} \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \phi(\mathbf{x}^i)\lambda(i, j - \mathbf{u}^j) \\ \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \mathbf{\Lambda}(\mathbf{y}^i)\lambda(i, j - \mathbf{u}^j) \\ \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \phi(\mathbf{x}^j)\lambda(i, j - \mathbf{u}^j) \\ \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \mathbf{\Lambda}(\mathbf{y}^j)\lambda(i, j - \mathbf{u}^j) \\ \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \varphi(\mathbf{x}^i, \mathbf{x}^j)\lambda(i, j - \mathbf{u}^j) \end{pmatrix}$$

$$\mathbf{\Psi}_{NL} = \begin{pmatrix} \sum_{i=1}^{T} \phi(\mathbf{x}^i)\lambda(0, \mathbf{u}^i) \\ \sum_{i=1}^{T} \mathbf{\Lambda}(\mathbf{y}^i)\lambda(0, \mathbf{u}^i) \end{pmatrix}$$

$$\mathbf{\Psi}(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{u}) = \begin{pmatrix} \mathbf{\Psi}_L \\ \mathbf{\Psi}_{NL} \end{pmatrix}$$

Note: $\mathbf{\Psi}_L$ and $\mathbf{\Psi}_{NL}$ correspond to the link and no-link emissions in the feature map $\mathbf{\Psi}(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{u})$ respectively as shown in the equations. The total dimension of the feature map is $3D + 3K + B$.

Author Index