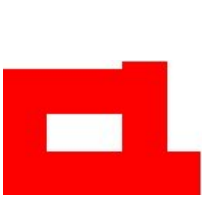
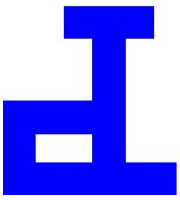
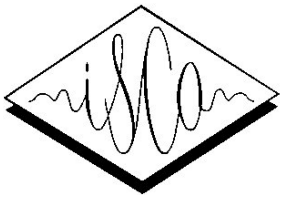
Proceedings of the

8th SIGdial Workshop on Discourse and Dialogue

Edited by Simon Keizer, Harry Bunt, and Tim Paek







1–2 September 2007
Antwerp, Belgium

Tilburg University, Department of Communication and Information Sciences
P.O. Box 90153
5000 LE Tilburg
The Netherlands

We thank our sponsors:

Order copies of this and other ACL proceedings from:

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

ISBN 9789074029322

Introduction

It is with great pleasure that we introduce the Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue. In putting together the accepted papers for the workshop, we encountered an uncommon but very welcome turn of events: we received many more papers with high recommendations from our Program Committee than we had anticipated. If we had kept the number of papers accepted to the maximum of previous SIGdial Workshops, several papers that had received all recommendations of accept would have been rejected. Given the abundance of favorably rated papers, we felt strongly that they should be given a chance for presentation. Besides the 18 full papers that were accepted for long presentation (out of 46 submitted), we therefore also accepted a number of papers for short presentation and created two poster sessions, to accommodate the extra girth. In addition, 5 out of 18 submitted short papers and demo descriptions were accepted as such. This was all in keeping with the tradition and purpose of the SIGdial venue to showcase promising new research approaches in discourse and dialogue, as well as state-of-the-art implementations.

Readers may notice that the workshop program has papers grouped into four topics: Multi-Party Dialogue, Spoken and Multimodal Dialogue Systems, Conversation Modeling and Dialogue Management. This organization was purely for presentation convenience, and quite often papers that were put under one rubric could be easily put under another.

We wish to thank the members of our illustrious Program Committee members for their advice in selecting papers for the workshop. The review process was facilitated by the ACL START system, which we received access to with the help of Antal van den Bosch and Claire Cardie. In preparing for the workshop we received very helpful advice from David Traum, Wolfgang Minker, Laila Dybkjær, and Kristiina Jokinen.

The actual Workshop could not have happened if not for the generous support of many people. Tilburg University staff managed the online and onsite registration, the production of the proceedings, and the local arrangements at the conference site in Antwerp. In particular, we wish to thank Jeroen Geertzen, Volha Petukhova, Femke Wieme and Lauraine Sinay. Torben Madsen, the SIGdial webmaster, helped put up our website and Priscilla Rasmussen of ACL and Christian Wellekens of ISCA advertised the event in their respective mailing lists.

We also thank the distinguished Prof. Herbert H. Clark of Stanford University for giving the SIGdial 2007 keynote address on “Rationality and Conversation.”

Finally, we wish to thank you, the SIGdial audience, for making our event a premiere forum for dialogue and discourse researchers. We hope you enjoy the collection of papers before you.

Harry Bunt (Co-Chair) Simon Keizer (Local Chair) Tim Paek (Co-Chair)

Organizers:

Harry Bunt, Tilburg University, Netherlands (co-chair)
Tim Paek, Microsoft Research, USA (co-chair)
Simon Keizer, Tilburg University, Netherlands (local chair)

Program Committee:

Arne Jönsson, Linköping University, Sweden
Alex Rudnicky, CMU, USA
Andrei Popescu-Belis, University of Geneva, Switzerland
Bonnie Webber, University of Edinburgh, UK
Candy Sidner, Bae Systems, USA
Claudia Soria, CNR, Italy
Dan Bohus, CMU, USA
David Traum, USC/ICT, USA
Emiel Krahmer, Tilburg University, Netherlands
Gokhan Tur, SRI, USA
Ingrid Zukerman, Monash University, Australia
Jan Alexandersson, DFKI GmbH, Germany
Jason Williams, AT&T Labs, USA
Jean Carletta, University of Edinburgh, UK
Jens Allwood, University of Göteborg, Sweden
Julia Hirschberg, Columbia University, USA
Justine Cassell, Northwestern University, USA
Kallirroi Georgila, University of Edinburgh, UK
Kristiina Jokinen, University of Helsinki, Finland
Laila Dybkjær, University of Southern Denmark, Denmark
Marc Swerts, Tilburg University, Netherlands
Marilyn Walker, Sheffield University, UK
Mark Core, USC/ICT, USA
Masato Ishizaki, University of Tokyo, Japan
Massimo Poesio, University of Essex, UK
Matthew Stone, Rutgers University, USA
Michael Johnston, AT&T Labs, USA
Michael McTear, University of Ulster, UK
Oliver Lemon, University of Edinburgh, UK
Patrick Paroubek, LIMSI-CNRS, France
Paul Piwek, Open University, UK
Robbert-Jan Beun, Universiteit Utrecht, Netherlands
Roberto Pieraccini, Speech Cycle, USA
Rolf Carlson, KTH, Sweden
Sadaoki Furui, Tokyo Institute of Technology, Japan
Srinivas Bangalore, AT&T Labs, USA
Stephanie Seneff, MIT, USA
Steve Young, Cambridge University, UK
Wolfgang Minker, University of Ulm, Germany

Table of Contents

Conference Program

Saturday, September 1, 2007

8:00–9:00 Conference Registration

9:00–9:15 Opening Remarks

9:15–10:15 Keynote Address by Prof. Herbert H. Clark

Rationality and Conversation
Herbert H. Clark

10:15–10:40 Coffee Break

Session: Multi-Party Dialogue
Area Chair: David Traum

10:40–11:05 *Collective States of Understanding*
Arash Eshghi and Patrick G.T. Healey

11:05–11:30 *Contrasting the Automatic Identification of Two Discourse Markers in Multiparty Dialogues*
Andrei Popescu-Belis and Sandrine Zufferey

11:30–11:55 *Detecting and Summarizing Action Items in Multi-Party Dialogue*
Matthew Purver, John Dowding, John Niekrasz, Patrick Ehlen, Sharareh Noorbaloochi and Stanley Peters

11:55–12:20 *Detecting Arguing and Sentiment in Meetings*
Swapna Somasundaran, Josef Ruppenhofer and Janyce Wiebe

12:20–13:30 Poster Session / Walking Lunches

A Model of Compliance and Emotion for Potentially Adversarial Dialogue Agents
Antonio Roque and David Traum

Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique
David Griol, Lluis F. Hurtado, Emilio Sanchis and Encarna Segarra

An Empirical View on IQA Follow-up Questions
Manuel Kirschner and Raffaella Bernardi

An Implemented Method for Distributed Collection and Assessment of Speech Data
Alexander Siebert, David Schlangen and Raquel Fernández

Saturday, September 1, 2007 (continued)

Beyond Repair Testing the Limits of the Conversational Repair System
David Schlangen and Raquel Fernández

Dialogue Policy Learning for Combinations of Noise and User Simulation: Transfer Results
Oliver Lemon and Xingkun Liu

Dynamic n-best Selection and Its Application in Dialog Act Detection
Junling Hu, Fabrizio Morbini, Fuliang Weng and Xue Liu

Emergent Conversational Recommendations: A Dialogue Behavior Approach
Pontus Wärnestål, Lars Degerstedt and Arne Jönsson

Exploiting Semantic and Pragmatic Information for the Automatic Resolution of Spatial Linguistic Expressions
Andrea Corradini

Hassan: A Virtual Human for Tactical Questioning
David Traum, Antonio Roque, Anton Leuski, Panayiotis Georgiou, Jillian Gerten, Bilyana Martinovski, Shrikanth Narayanan, Susan Robinson and Ashish Vaswani

Identifying Formal and Functional Zones in Film Reviews
Heike Bieler, Stefanie Dipper and Manfred Stede

Session: Spoken and Multimodal Dialogue Systems
Area Chair: Marilyn Walker

13:30–13:55 *CHAT to Your Destination*
Fuliang Weng, Baoshi Yan, Zhe Feng, Florin Ratiu, Madhuri Raya, Brian Lathrop, Annie Lien, Sebastian Varges, Rohit Mishra, Feng Lin, Matthew Purver, Harry Bratt, Yao Meng, Stanley Peters, Tobias Scheideck, Badri Raghunathan and Zhaoxia Zhang

13:55–14:20 *Commute UX: Telephone Dialog System for Location-based Services*
Ivan Tashev, Michael Seltzer, Yun-Cheng Ju, Dong Yu and Alex Acero

14:20–14:45 *Corpus-Based Training of Action-Specific Language Models*
Lars Schillingmann, Sven Wachsmuth and Britta Wrede

14:45–15:10 Tea Break

Saturday, September 1, 2007 (continued)

Session: Spoken and Multimodal Dialogue Systems (Continued)

15:10–15:35 *Negotiating Spatial Goals with a Wheelchair*
Thora Tenbrink and Hui Shi

15:35–16:00 *Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms*
Alexander Gruenstein and Stephanie Seneff

Short Paper Session: Spoken Dialogue Systems

16:00–16:15 *Analysis of User Reactions to Turn-Taking Failures in Spoken Dialogue Systems*
Mikio Nakano, Yuka Nagano, Kotaro Funakoshi, Toshihiko Ito, Kenji Araki, Yuji Hasegawa and Hiroshi Tsujino

16:15–16:30 *Comparing Spoken Dialog Corpora Collected with Recruited Subjects versus Real Users*
Hua Ai, Antoine Raux, Dan Bohus, Maxine Eskenazi and Diane Litman

16:30–16:45 *Dealing with DEAL: A Dialogue System for Conversation Training*
Anna Hjalmarsson, Preben Wik and Jenny Brusk

16:45–17:00 *Referring under Restricted Interactivity Conditions*
Raquel Fernández, Tatjana Lucht and David Schlangen

18:00–20:00 Reception

Sunday, September 2, 2007

Session: Conversation Modeling
Area Chair: Jan Alexandersson

9:00–9:25 *A Multidimensional Approach to Utterance Segmentation and Dialogue Act Classification*
Jeroen Geertzen, Volha Petukhova and Harry Bunt

9:25–9:50 *Accented Pronouns and Unusual Antecedents: A Corpus Study*
Anubha Kothari

9:50–10:15 *Evaluating Combinations of Dialogue Acts for Generation*
Simon Keizer and Harry Bunt

10:15–10:40 Coffee Break

Session: Conversation Modeling (Continued)

10:40–11:05 *Measuring Adaptation Between Dialogs*
Svetlana Stenchikova and Amanda Stent

11:05–11:30 *Token-based Chunking of Turn-internal Dialogue Act Sequences*
Piroska Lendvai and Jeroen Geertzen

Short Paper Session: Conversation Modeling

11:30–11:45 *A Comprehensive Disfluency Model for Multi-Party Interaction*
Jana Besser and Jan Alexandersson

11:45–12:00 *Experimental Modeling of Human-human Multi-threaded Dialogues in the Presence of a Manual-visual Task*
Alexander Shyrokov, Andrew Kun and Peter Heeman

12:00–12:15 *Modeling Vocal Interaction for Text-Independent Classification of Conversation Type*
Kornel Laskowski, Mari Ostendorf and Tanja Schultz

12:15–13:25 Poster Session / Walking Lunches

Introducing Utterance Verification in Spoken Dialogue System to Improve Dynamic Help Generation for Novice Users
Kazunori Komatani, Yuichiro Fukubayashi, Tetsuya Ogata and Hiroshi G. Okuno

Sunday, September 2, 2007 (continued)

Making Grounding Decisions: Data-driven Estimation of Dialogue Costs and Confidence Thresholds
Gabriel Skantze

On the Training Data Requirements for an Automatic Dialogue Annotation Technique
Carlos D. Martínez-Hinarejos

Practical Dialogue Manager Development using POMDPs
Trung H. Bui, Boris van Schooten and Dennis Hofs

Problem-Sensitive Response Generation in Human-Robot Dialogs
Petra Gieselmann and Mari Ostendorf

Rapid Development of Dialogue Systems by Grammar Compilation
Björn Bringert

Resolving "You" in Multi-Party Dialog
Surabhi Gupta, John Niekrasz, Matthew Purver and Dan Jurafsky

SIDGRID: A Framework for Distributed, Integrated Multimodal Annotation, Archiving, and Analysis
Gina-Anne Levow, Bennett Bertenthal, Mark Hereld, Sarah Kenny, David McNeill, Michael Papka and Sonjia Waxmonsky

ScIML: Model-based Design of Voice User Interfaces
Jörn Kreutel

Tutoring in a Spoken Language Dialogue System
Jaakko Hakulinen, Markku Turunen and Kari-Jouko Räihä

Using Speech Acts in Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue
Vladimir Popescu, Jean Caelen and Corneliu Burileanu

Sunday, September 2, 2007 (continued)

Session: Dialogue Management
Area Chair: Oliver Lemon

13:25–13:50 *Dialogue Management for Automatic Troubleshooting and other Problem-solving Applications*
Johan Boye

13:50–14:15 *Implicitly-supervised Learning in Spoken Language Interfaces: an Application to the Confidence Annotation Problem*
Dan Bohus and Alexander Rudnicky

14:15–14:40 *Planning Dialog Actions*
Mark Steedman and Ronald Petrick

14:40–15:05 Tea Break

Session: Dialogue Management (Continued)

15:05–15:30 *Statistical User Simulation with a Hidden Agenda*
Jost Schatzmann, Blaise Thomson and Steve Young

Short Paper Session: Dialogue Management

15:30–15:45 *An Empirically Based Computational Model of Grounding in Dialogue*
Harry Bunt, Roser Morante and Simon Keizer

15:45–16:00 *Pragmatic Usage of Linear Regression Models for the Prediction of User Judgments*
Klaus-Peter Engelbrecht and Sebastian Möller

16:00–16:15 Closing Remarks

16:15–17:15 SIGDIAL Business Meeting

KEYNOTE ADDRESS

Rationality and Conversation

Herbert H. Clark
Department of Psychology
Building 420, Jordan Hall
Stanford University
Stanford CA, USA 94305–2130
email: clark@stanford.edu

Abstract

In the model of language use proposed by philosopher H. Paul Grice, people in conversation recognize "a common purpose or set of purposes, or at least a mutually accepted direction," and they cooperate in contributing to those purposes. Grice went on to argue, "Talking [is] a special case or variety of purposive, indeed rational, behavior." But Grice tacitly assumed a type of omniscient rationality: People in conversation have perfect knowledge of the language and the current common ground, and they have an unlimited processing capacity in choosing what to say. In reality, people's rationality is bounded, and that leads to quite a different view of language use. I take up some of the consequences of bounded rationality in language use.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, page 1,
Antwerp, September 2007.

Collective States of Understanding

Arash Eshghi
Department of Computer Science
Queen Mary University of London
Mile End Road, London, E1 4NS
arash@dcs.qmul.ac.uk

Patrick G. T. Healey
Department of Computer Science
Queen Mary University of London
Mile End Road, London, E1 4NS
ph@dcs.qmul.ac.uk

Abstract

This paper uses an analysis of ellipsis in multi-party interaction to investigate the relative accessibility of dialogue context/common ground to direct addressees and side participants. The results show that side-participants frequently make direct use of the common ground established between a speaker and addressee despite the fact that, by definition, they did not directly collaborate with the speaker on constructing it. Different individuals can thus reach the same level of grounding through different levels of feedback. We conclude that mutliparty dialogue leads to distinct collective states of understanding that are not reducible to the component dyadic interactions.

1 Introduction

Goffman (1981) introduced a distinction between *ratified participants* and *overhearers* in a conversation. The former category is further decomposed into *direct addressees* (DA) and *side participants* (SP) of an utterance. The ratified participants are those who hold certain responsibilities towards each other for ensuring mutual-understanding (Clark and Schaefer, 1992):

Principle of Responsibility: In a conversation, the parties to it are each responsible for keeping track of what is said, and for enabling the other parties to keep track of what is said.

In dyadic interactions, mutual-understanding or 'grounding' is achieved through direct collaboration between the speaker and addressee. The speaker expects the addressee to provide evidence that he is understanding the speaker's utterance "to criteria sufficient for current purposes" (Clark and Brennan, 1991). In multi-party conversations the situation is more complex.

For example, if A makes an anaphoric reference to some entity, while addressing B with C present as a side-participant, he intends both B AND C to resolve the reference. However, by definition, the speaker does not collaborate as actively with side-participants. They "have to be satisfied with clearing up misunderstandings in natural breaks in their talk" (Clark and Schaefer, 1992). A SP will normally wait until speaker and addressee have carried out their *presentation* and *acceptance* phases, before attempting to rectify any possible misalignment with the speaker. On this account grounding between speaker and direct addressee always takes precedence.

By definition, SPs and DAs give different evidence of grounding of a speaker's utterances; DA's respond overtly and directly but SPs provide weaker evidence of grounding – primarily continued attention and withholding of repair. Consequently, if we understand level of grounding as being directly dependent on the level of 'evidence of acceptance' provided then we expect differences in the relative accessibility of the common ground for the different pairs of participants; roughly, Speaker & $DA >$ Speaker and $SP > SP$ and DA.

In a review article Branigan (2006) points out that

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 2–9,
Antwerp, September 2007.

there isn't yet any empirical evidence that DAs and SPs differ in the accumulation of common ground. In this paper we provide evidence that, in at least one case, the common ground is, in fact, equally accessible to SPs and DAs. We argue that this is evidence of *collective states of understanding* that are not reducible to the component dyadic interactions. It appears that in modelling multilogue we need to account for the possibility that one participant can stand proxy, in terms of grounding, for another (ratified) participant.

1.1 Side Participants in the Tangram Task

The key empirical evidence relating to grounding by SPs comes from the tangram experiments carried out by Clark and co-workers. The essence of these tasks is that on each trial one person, the 'Director' (D), describes a series of tangram figures so that another, the 'Matcher' (M) can identify them. If the same figure recurs on a number of trials the Director and Matcher quickly converge on some concise definite or nominal description for the figure. For example, they go from "Okay the next one is ... resembles someone that looks like they're trying to climb stairs. There's two feet, one is way above the other and–" on trial one to "Um, stair climber" on trial 6 ((Wilkes-Gibbs and Clark, 1992), p.184).

Wilkes-Gibbs and Clark (1992) carried out a two phase variation on this basic task. The first phase has two conditions. In one an additional 'silent' SP sits next to the director. In another an 'omniscient overhearer' (OO) follows all of phase 1 on video but is not co-present in the room. In both conditions the D is aware of the additional participants and they are both able to see each figure as the D describes it. In the second phase the SP or OO take on the role of matcher for another six trails. The D and former SP pairs are quicker, use fewer words and produce more definite/nominal description types than the D and former OO pairs; despite the fact that the SP and OO ostensibly have the same prior information.

Although this is clear evidence that side-participants reach a higher level of understanding than overhearers it is inconclusive about the SP - DA contrast. The SP - DA distinction relates to participant status with respect to utterances in the same conversation (Goffman, 1981). The experimental device of two task phases effectively breaks the interaction into two successive conversations where a direct comparison of SP and DA with respect to the same interaction is not made. The closest approximation is the comparison of the last trial of phase one and the first trial of phase 2 but this is equivocal. The Director-SP pairs are slower and use more words than the original Director-Matcher pairs but do make the same number of definite/nominal references. The task situation is also unusual in that in phase one the SP is positioned beside the Director and opposite the Matcher. The participants mutually know that the SP has direct visual access to the actual referents of the referring expressions whereas the Matcher does not. Arguably this gives the SP an unusually high degree of access to the common ground.

In this paper we compare the relative accessibility of common ground to different participants in a **single** multi-party conversation. In order to improve the ecological validity of the analysis we focus on (relatively) naturalistic dialogues between three or four participants. To provide a more sensitive index of the kinds of information that are assumed to be in the common ground we focus on the use of different kinds of ellipsis. We argue that, in fact, SP's and DA's are in all relevant respects equivalent and that this is evidence for distinct collective states of understanding that are not reducible to the component dyadic interactions. Like Branigan(2006) we argue that the ultimate difference between SP and DA grounding if any, is due to the goals of these participants in the conversation i.e. to what they individually judge to be 'sufficient for current purposes' in the context of the current activity.

2 Method

Before describing the analysis in more detail we first introduce the corpus used.

2.1 The AMI Corpus

The AMI Meeting Corpus (Carletta, 2006) is a multi-modal (video, audio and text) set of 100 hours of meeting recordings. These consist of a set of naturally occurring and a set of scenario-based meetings. In this paper 10 of the naturally occurring meetings-roughly 9 hours of conversation- have been analysed. Only the video, audio and raw transcripts have

been used. For more information on AMI refer to http://www.idiap.ch/amicorpus.

2.2 Side Participants to Strips of Dyadic Talk

For reasons which will become clear, in order to make claims about a speaker's assumptions regarding SP understanding, we extracted all strips of dyadic talk from each meeting. These are segments during which there is no explicit feedback (except 'continued attention') , from participants other than speaker and addressee. This provides identifiable SP's and DA's for each dyadic segment (see below). Based on the turn taking model in (Sacks et al., 1974), these dyadic segments of talk end in one of two ways:

1. **Self-selected side participant (SP)**: a SP wins the floor by exploiting a gap in the dyadic talk, or she interrupts the talk mid-utterance.

2. **Nominated by Last Speaker (LS)**: Last speaker hands the floor over to a SP, by directly addressing her.

It is in general a current speaker's paralinguistic behaviour (gaze and body orientation) and/or the content of her utterance (e.g. use of personal pronouns accompanied by gaze) which together determine whom she is directly addressing. When a SP is directly nominated (addressed) at the end of a segment, it's the same information which signals a change in the speaker's set of DAs. Note that the DA is determined through reference exclusively to the speaker's behaviour. Also we take into account that the speaker might be 'addressing' the other participant in the dyad while making a SP the intended recipient as when the SP is the 'butt' of a speaker's joke (Levinson, 1988).

3 Analysis of Ellipsis

At the end of a dyadic segment the participants hold certain assumptions about each other's level of understanding. One way these assumptions are made manifest is in the *elliptical* expressions employed by the speaker.

Ellipsis is a mono/dialogical technique in producing expressions, whereby single or multiple sentence constituents are omitted. The 'complete' meaning of such elliptical expressions can be recovered (resolved) by reference to previous utterances/sentences the contents of which are immediately present in context.

Ellipsis is central to this analysis since it indexes the extent to which the meaning of an utterance depends directly on the context of the preceding dyadic exchange i.e. the extent to which participants assume the common ground established during the dyadic exchange is accessible to each other. More specifically, at the point when the dyadic exchange ends we have the opportunity to compare a) the pattern of use of ellipsis by the last speaker to the SP with b) the pattern used by the SP to the last speaker (LS).

If the LS addresses the SP elliptically they are demonstrating their assumption that the SP grounded the antecedent referents/propositions during the prior dyadic conversation. Conversely when the SP self-selects (interrupts), the use of ellipsis demonstrates the extent to which the SP's directly access the other participants' common ground.

Our first level classification distinguishes four categories:

- **CD (context-dependent)**: Utterance contains Syntactic Ellipsis, Anaphoric OR Definite reference.

- **CT (continuation of talk)**: In terms of semantic content, the utterance could intuitively be thought of as the continuation of the talk in the segment, i.e. utterance does not have a coherent meaning without the background of the dyadic talk.

- **BC (backchannel)**: Having been 'silent' throughout the dyadic segment, the SP merely starts to backchannel again.

- **NC (new context)**: Introduction of a new context/topic.

This scheme yields the following segment types: LS_{CD}, LS_{CT}, LS_{BC}, LS_{NC}, SP_{CD}, SP_{CT}, SP_{BC}, SP_{NC}.

For a second, more detailed level of analysis that takes the kind of ellipsis into account we further decomposed the CD category:

3.1 Ellipsis Taxonomy

1. **Non-Sentential Utterances (NSU)**: Fragmentary but intuitively complete utterances, exclusive to dialogue that are not sentential in their outward form. These utterances have been coded according to the typology developed in (Fernandez and Ginzburg, 2002). We have further collapsed these types according to their role/function in conversation, into the following more general categories:

   - Direct Answers (DA): Fragments used as answers to questions. Includes *Polar Answers* and *Short Answers*.
   - Clarification Requests (CR): Fragments in question form, used to request clarification or further elaboration of a previous utterance. Includes *Clarification Ellipses* and *Sluices*.
   - Modifiers (MOD): In their fully resolved form, these are statements somehow modifying a previous utterance in conversation. Includes *Propositional Modifiers*, *Factual Modifiers*, *Fillers* and *Fragments introduced by Connectives*.

2. **Sentential Ellipsis**: These are contained in utterances which are sentential, but semantically ambiguous as a result of either the full omission a syntactic constituent or its replacement by an auxiliary. In the case of stand-alone uses of propositional attitude verbs (know, see, believe ...), the whole of the antecedent utterance gets elided. Often the omitted/replaced syntactic constituent (not necessarily atomic/terminal) can be uniquely identified and recovered from context. Unlike NSU's these are not exclusive to dialogue. Here's an example:

   Verb Phrase (VP) Ellipsis:
   A: Will you please go to the market tomorrow?
   B: I already told you I will. [Resolved Content: "I already told you I will go to the market tomorrow"]

   We have developed an ad hoc taxonomy analogous to that for NSU's, based on the role/function of the utterance containing the ellipsis. Bear in mind that the taxonomy is being used merely to compare what SPs and DAs can 'do' elliptically.

   - Direct Answers (DA): Utterance containing the ellipsis is an answer to a question, like the above example.
   - Request for confirmation (RC): Partly redundant, these are tag questions used to request confirmation or initiate disputation. "A: I got an A in Biology. B: Did you? A: Yes. I got the results today."
   - Statement (ST): General category containing all statements, excluding Direct Answers.
   - Query: All elliptical questions excluding Requests for Confirmation.

3. **Anaphora (Anaph)**

4. **Definite/Nominal Reference (DR)**

To provide a baseline comparison of ellipsis types in ordinary dialogue we also coded 10 peoples conversations from the British National Corpus (BNC).

4 Results and Discussion

Table 1 shows dyadic segment type counts, for 10 AMI meetings (roughly 9 hours of conversation).

4.1 Segments of type LS_{CD}: Assumptions about SPs

All such segments indicate that the last speaker, in producing elliptical utterances addressing a SP, is tacitly making the assumption that the SP would be able to resolve the ellipses employed, which in turn depends directly on the SP having grounded the antecedent utterance(s) of the ellipsis contained within the segment, for which the SP did not produce any explicit feedback. Note that 'Continued attention' by the SP(s) is very frequently not monitored by any of the participants in the dyad. Eye contact is more or less exclusively maintained between the two and them alone. Nevertheless the SP is 'expected' by the last speaker to have grounded the antecedent utterance(s). Furthermore, none of these segments were followed by any form of Repair/Clarification by the SP. In all of them the SP seems to be coping perfectly well with the elliptical utterance, and the conversation goes on

	CD	CT	NC	BC
LS	20	4	3	4
SP	100	33	1	0

Table 1: Dyadic Segment Type Counts

smoothly.

This evidence seems to support the claim in (Branigan, 2006) that speakers have very similar and at times even higher expectations from SPs compared to those from DAs, concerning the participant's ability to resolve these ellipses/references. Nevertheless Branigan also proposes that these expectations from SPs should often be weaker.

The following are excerpts from AMI, showing the different kinds of ellipsis employed by the last speakers:

Anaphoric chains: distant antecedent recognised by SP

B: Yeah. But that still won't tell you. well howmany **tangrams** are there that they're using? Fifteen or something.
C: Uh no, not even that. They've of this relevant type . . . .
B: Uh-huh. So that's not gonna so that's not gonna tell you anything about **their** relative complexity. . . You still need some kinda scale for **these things**. Ca uh if you look at'**em**, do you just know?
C: Mm no. [laugh] Well I don't. I'm not .
B: No. I wouldn't either. What about him? I if Mister Geometry. I mean, you know. Can you tell just by looking at **these** how hard people find them?
A: No, I'm afraid not. I wouldn't know.

In the above excerpt, also note how similar C's (the DA) last utterance is to A's (the SP) : VP ellipsis in C's versus whole sentence ellipsis in A's utterance.

The whole segment as antecedent

B: [*. . 6 utterances so far exchanged between B and C*] Data I think we should keep in.
C: OK. [laugh]
B: Because it's would be the same as feature.
B: Or spec spectrum. I think data's the same as spectrum . . .
C: I do I still don't think that goes in. But .
C: yeah, I still don't like it. But
B: Final view, Bob?
A: I don't have passionate feelings.

Here, B's last utterance explicitly addressing A, is highly elliptical with no particular utterance as antecedent, i.e. the resolved content of the utterance depends on the whole segment between B and C. B expects A (Bob) to have grasped the issue under discussion. One would expect A here to initiate clarification if he really didn't know what B was asking.

We think that the speaker's assumptions about SPs are among other things, strongly mediated by the speaker's prior beliefs (before the conversation) about the SP and his relevant knowledge. In the meeting from which the above was extracted, A is a supervisor with whom the rest of the participants check their results as they go along. So, firstly if cooperative, he should be 'paying attention' to the dyadic interactions in which he is not directly involved (most of the meeting). Secondly, the rest of the participants believe to begin with, that he would understand such technical issues under discussion. So perhaps, it is some notion of the well known 'lab coat effect' that could justify such high expectations (e.g. see (Healey and Mills, 2006), page 5).

4.2 Segments of type SP_{CD}: Side participant access to 'communal' common ground

In this section we will argue that SPs have the same kind of access through the same techniques, to the 'communal' common ground, as the participants directly involved in collaboratively securing it (speaker and addressee). These segments which comprise the largest class in this analysis, end when a SP interjects producing an elliptical and hence context-dependent utterance. Again here, the antecedents of the ellipses, lie within the dyadic segments.

Table 2 below shows the ellipses identified in these SP utterances. They have been classified according to the taxonomy described in section 3.1. In order to assess whether there is a difference between the use of ellipsis types by SP's and the baseline

- typical frequency of use independent of both the number of participants in the conversation and the status of the participant upon employing the ellipsis - used in ordinary dialogue, we compared the frequency of ellipses of each type with that found in the BNC. Taking into consideration all categories in Table 2 (merging Sentential and non-Sentential DAs and ignoring DR since it wasn't coded for the BNC) there is a reliable difference (Chi^2_6 = 14.6, p = 0.02). However, as Table 2 indicates the main difference is in the relative frequency of direct answers which account for 26% of instances in the BNC but only 12% of instances in AMI. If this category is ignored we find no reliable difference between SP's and the baseline (Chi^2_5 = 4.33, p = 0.50).

The difference in frequency of use of direct answers is essentially an artefact of our coding scheme. As noted above the SP ellipses are those where they have nominated themselves as next speaker by interjecting. Consequently, direct answers by SPs to questions –which are by definition not directed at SPs– are much rarer. Subject to this caveat, we can conclude that the pattern of use of different ellipsis types by SP's is not in fact distinguishable from the pattern of use typical of participants in ordinary dialogue.

What now follows is a discussion over a set of examples from AMI comparing the kinds of access to context through various elliptical phenomena, possible by SPs to those by DAs.

4.2.1 Anaphora with distant whole utterance as antecedent

B: Um so this person didn't ha um the obviously didn't know about capitalisation. So just about every utterance needs to be capitalised and and needs the end punctuation. (1)

D: Mm-hmm. (2)

D: You know, when you get like um someone's talking and there's they sort of pause in the middle of a sentence that's long enough for it to put a break in, (3)

B: Yeah (4)

D: but they're actually sort of carrying on the sentence, do you have to capitalise each time you transcribe a bit of it's mid (5)

B: Um, no no no. (6)

D: No no no no. Yeah. (7)

B: Whatever um makes sense to you. (8)

D: Okay. (9)

B: Um [cough] but no, it it can continue into the next segment and that's perfectly fine. (10)

D: Yeah. Just okay. So it's put the hyphen and then. (11)

C: Yeah. (12)

C: I think that's actually the only case where you don't (13)

C: or where you're not supposed to capitalise, right? (15)

Utterances 1 to 12 above form a segment of type SP_{CD} which is terminated by C. The anaphora "that" in 13 can only be resolved with 3 as an antecedent. An issue is here raised initially by B to which D responds by asking a question (utterance 3). All the way down to utterance 12 the question is under discussion exclusively between B and D. C then produces utterance 13 which can intuitively be thought of as an answer to D's initial question. In other words it could have been produced by B (the DA) adjacently to the initial question. Note here that C has had to re-raise the context in order to make her contribution. I.e. a simple "No" (a Negative Polar Answer) like B's initial response, or even the less elliptical "I think that's actually the only case where you don't.", would most probably be infelicitous (the other party would be very likely to initiate clarification). But this seems to be the effect of antecedent distance alone, since all of the NSU classes in (Fernandez and Ginzburg, 2002) are possible by SPs at the end of the segments in question, but clearly not at such high antecedent distances. This will be a little further elaborated in section 4.4.

4.2.2 Non-Sentential Utterances (NSU)

Factual Modifier

C: When I did my masters um I took uh SP1 and SP2 with Simon King.

A: You survived SP1 and SP2.[laugh]

C: Yes. And actually I've done quite well in SP1, I've done it a bit worse in SP2 because it was a l a lot more challenging.

	Non-Sentential			Sentential				Other		
	DA(NSU)	**CR**	**MOD**	**ST**	**DA**	**Query**	**RC**	**DR**	**Anaph**	**Tot**
AMI	14	9	12	10	0	4	6	11	58	124
Baseline(BNC)	154	73	46	46	21	12	20	not coded	303	675

Table 2: Ellipses employed by SPs terminating SP_{CD} segments compared to the baseline (BNC)

A: We have two new teachers for SP2.
B: **Too many**. [laugh]

The excerpt above shows an instance of a SP Factual Modifier (boldfaced in the excerpt) produced by B adjacently to its antecendent. The same utterance "Too many." by the DA (B here) would have been perfectly felicitous (implies in this case, equivalence of SP and DA access to context).

Among the NSU classes in (Fernandez and Ginzburg, 2002), Clarification Ellipsis (CE) is of a special status, since it is known to be a common technique used in dyadic dialogue to ground utterances which weren't sufficiently understood by the recipient. There were very few CEs identified in this analysis. However, we do know that SPs can and do in fact initiate elliptical clarification, by exploiting gaps in dyadic talk:

Clarification Ellipsis (CE)

C: What does cutest spelling mean? (1)
B: oh, she spelled cutest um with an I_, (2)
C: oh, okay. (3)
B: so that that's just something I pointed out. (4)
D: oh yeah. (5)
A: Cutest? [*Gazing at D. Direct Addresee is D here.*](6)
D: E_S_T_ (7)
A: Thank you.[laugh] (8)

D and A above are both SPs to the dyadic segment between B and C. The CE produced by A is very interestingly addressed at D who is also a SP to what's being clarified, which shows that in multi-party dialogue all the ratified participants have obligations/responsibilities towards one another.

This example also indicates clearly that there can be varying levels of understanding among the SPs themselves. However, note that we are not claiming by any means that in multi-party situations, the participants always reach a collective state of understanding. The claim is rather that such collective states do exist, and that they're often assumed by the parties involved.

Furthermore, it's interesting to see that had it been B (the DA of the antecedent utterance) who didn't understand, she would have produced the CE locally (as opposed to a distance of 5 here) which is what's generally expected in dyadic dialogue. This issue is further discussed in section 4.4 for future work on distance.

4.2.3 Sentential Ellipsis

These are the ellipses not covered by the NSU typology in (Fernandez and Ginzburg, 2002). The taxonomy described in section 3.1 has been used to classify these.

local VP ellipsis by SP

B: [*5 utterances exchanged between B and C so far in the segment*] but I I I do know the type of scenario you're describing. I just it's just hard to answer that without hearing something. Mm.
C: Mm-hmm. **The um** should be capitalised.
B: yeah, **they** should all. I stopped marking **them**, 'cause there are just too many.
C: yeah.
A: Should **it**? 'Cause the loose uh is continuing from one sentence isn't it?

Note also the chain of anaphora referring to the 'um', and how it carries on across to the SP's (A's) utterance. This phenomenon is very frequent in SP utterances terminating SP_{CD} segments.

4.3 Segments of type SP_{NC}: Implications for our claims

The analysis indicates that the introduction of new contexts/topics by SPs interrupting a dyadic segment, is extremely unlikely. Consequently if a SP is

to interrupt, she has to 'stick to the topic' already under discussion in the segment. This further supports our claims, in that even if a SP is not using ellipsis as direct access to the 'communal' common ground, she makes use of the information in there, to produce a relevant utterance. An utterance thus produced, semantically depends on and is incoherent without the background of the 'shared knowledge' established between the speaker and addressee in the dyad.

4.4 Future Work: *Antecedent Distance* of SP ellipses and *Context Re-raising*

The technique of re-raising context- avoiding highly elliptical expressions or in the case of anaphora, giving further descriptions of the discourse entities referred to- is frequently adopted by a SP in his attempt to produce a distant second pair part to an utterance far back within a dyadic exchange. This technique need not be exclusive to SPs, as DAs also in dyadic dialogue might do this to produce an utterance which isn't locally 'relevant', but counts as a second pair part to what's been discussed further back. However it is expected to be employed a lot more frequently by SPs in face-to-face multi-party dialogue.

This issue raises the following questions: What is the correct notion of antecedent distance here? What exactly is the threshold in terms of this notion for each ellipsis type, after which interjecting SPs need to avoid the ellipsis in order to prevent ambiguity/miscommunication? Or more formally with respect to interaction protocols, how does antecedent distance fit into ellipsis felicity conditions in multilogue?

5 Conclusion

The evidence from this analysis shows that with respect to common ground side participants in the AMI corpus do not appear to be different in any substantive respect from direct addressees. Speakers assume that SPs reach the same level of understanding as the addressees. Additionally, SPs were shown to use elliptical techniques to access the shared-context, in generally the same way as DAs do. All things being equal, this is strong evidence for collective states of understanding that could not be predicted from considering the component dyads alone since, prima facie, the SP's don't ground to the same level, don't go through the same grounding cycle as DA's with the speaker. Moreover it indicates that DAs can act as proxy for SPs in providing understanding evidence and, presumably, that they have obligations to each other. Finally, this all seems to make it simply a matter of winning the floor for SPs. Other than that there's no difference between the ratified participants in multi-party conversation.

References

H. Branigan. 2006. Perspectives on multi-party dialogue. *Special issue of Research on Language and Computation*.

J. Carletta. 2006. Announcing the ami meeting corpus. *The ELRA Newsletter*.

H.H. Clark and S.A. Brennan. 1991. Grounding in communication. *Perspectives on socially shared cognition*.

H.H. Clark and E.F. Schaefer. 1992. Dealing with overhearers.

R. Fernandez and J. Ginzburg. 2002. Non-sentential utterances: A corpus study. *Traitement automatique des languages*.

Erving Goffman. 1981. *Forms of Talk*.

P.G.T. Healey and G.J. Mills. 2006. Participation, precedence and co-ordination in dialogue. *Cognitive Science*.

S. Levinson. 1988. Putting linguistics on a proper footing: Explorations in goffman's concepts of participation.

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

D. Wilkes-Gibbs and H.H. Clark. 1992. Coordinating beliefs in conversation. *Memory and Language*.

Contrasting the Automatic Identification of Two Discourse Markers in Multiparty Dialogues

Andrei Popescu-Belis
ISSCO / TIM / ETI
University of Geneva
Bd. du Pont-d'Arve 40
1211 Geneva 4, Switzerland
andrei.popescu-belis@issco.unige.ch

Sandrine Zufferey
Department of Linguistics
University of Geneva
Rue de Candolle 2
1211 Geneva 4, Switzerland
sandrine.zufferey@lettres.unige.ch

Abstract

The identification of occurrences of *like* and *well* that serve as discourse markers (DMs) is a classification problem which is studied here on a corpus of dialogue transcripts with more than 4,000 occurrences of each item. Decision trees using item-specific lexical, prosodic, positional and sociolinguistic features are trained using the C4.5 method. The results demonstrate improvement over past experiments, reaching the same range as inter-annotator agreement scores. DM identification appears to benefit from item-specific classifiers, which perform better than general purpose ones, thanks to the differentiated use of lexical features.

1 Introduction

The identification of discourse markers (DMs) is an essential step in dialogue understanding, especially when the lexical items used as DMs are ambiguous. *Like* and *well* are two frequent lexical items and potential DMs, which are among the most difficult ones to disambiguate, and they will serve here as a case study for automatic DM identification. The task will be discussed first from a linguistic and computational point of view. Previous attempts will be summarized, followed by the data, features and classifiers used here. The results will be discussed first by comparing our highest scores with baseline ones, then by analyzing the relevance to DM identification of various features. The best performances are shown to be comparable to inter-annotator agreement scores and higher than state-of-the-art scores, and lexical collocations are shown to be the most relevant features.

2 The Discourse Markers *Like* and *Well*

Despite the wide research interest raised by DMs for many years, there is no generally accepted definition or list of DMs (Andersen, 2001; Schiffrin, 1987). Items typically featured in this class are also called discourse connectives, pragmatic markers, or cue phrases, and include words and expressions such as *actually*, *and*, *but*, *I mean*, *like*, *so*, *you know* and *well*, which "generally have little lexical import but serve significant pragmatic functions in conversation" (Andersen, 2001, page 39).

For comparison purposes, we focus here on two lexical items, *like* and *well*, in order to determine the surface features that are most relevant to DM classifiers based on machine learning. These two items are among the most frequent and the most ambiguous DMs. *Like*, for instance, can be a preposition or an adverb, a verb or even a noun. When used as a DM, *like* is in reality much more than a filler, and can be more precisely described as a loose talk marker, signalling reported speech or an imprecise formulation of the speaker's belief, as in "He was like, yeah, I can make dogs raise their ears" or "It took, like, twenty minutes"—for more examples, see (Popescu-Belis and Zufferey, 2006, pages 7–9).

Well can also fulfil a variety of pragmatic and non-pragmatic functions (Schourup, 2001). When it is not a DM, *well* can be an adverb or an adjective ("He sings well", "I am well"), or a noun or verb ('water source'). As a DM, *well* can introduce a rejection of a previous request, or a disagreement with a previ-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 10–17,
Antwerp, September 2007.

ous utterance, or can more generally mark hesitation or turn-taking, as in "Well, actually, you don't even need to do that..." or "Oh, yes, but well, uh, yes, but what I mean is that...".

3 Evaluation of DM Identification

The automatic identification of DMs is a binary classification task over the entire set of occurrences of the lexical item. Its evaluation requires a ground truth classification, and metrics to compare a candidate classification to it. The simplest evaluation metric is accuracy, i.e. the percentage of correctly classified instances (CCIs or C below). In addition, if DM identification is seen as the retrieval of the DMs among all occurrences of a lexical item, then recall (r), precision (p) and their f-measure (f) can be used to assess performance in a more detailed manner.

However, given that the distribution of DM vs. non-DM occurrences of a lexical item is seldom uniform, the above metrics should be corrected for chance agreement. To our knowledge, there are no widely used chance-corrected versions of recall and precision—the Kullback-Leibler divergence is seldom used for classification tasks—but a well-known agreement metric that *is* chance-corrected is the *kappa* (κ) score (Carletta, 1996). Although designed to measure inter-annotator agreement, κ quantifies the resemblance of two classifications by factoring out agreement by chance.

The κ score measures classification performance between -1 and 1, with random classification scoring 0. The κ measure is quite strict as it was designed to be sensitive to even small differences between coders. Therefore, a κ value above 0.67 is often considered a sign of acceptable agreement, while a value above 0.8 is considered very significant. According to Landis and Koch (1977), strength of agreement is fair for $0.2 < \kappa \leq 0.4$, moderate for $0.4 < \kappa \leq 0.6$, substantial for $0.6 < \kappa \leq 0.8$ and almost perfect above. In any case, the inter-coder agreement for the gold standard data represents the upper bound that can be legitimately expected from a classifier: even a perfect one cannot get closer to the gold standard than the humans who defined this standard.

4 Previous Studies of DM Identification

DMs play a considerable role in discourse processing tasks. For instance, some studies use discourse connectives to infer discourse structure (Reichman, 1985; Grosz and Sidner, 1986; Marcu, 1998), while others use DMs as cue words for discourse segmentation (Passonneau and Litman, 1997).

Many DMs, especially connectives or cue words, are not as highly ambiguous as *like* or *well*. Hutchinson (2004, page 686), for instance, targeted mainly the problem of automatic categorization of the pragmatic functions of discourse connectives, but only acknowledged the potential ambiguity of *and*. Similarly, Marcu's (1998) algorithm for DM identification, in relation to rhetorical parsing of written texts, aims at a list of 450 potential DMs, but *and* and *or* are ignored in many cases due to their ambiguity. It is also likely that *like* and *well* did not appear often in Marcu's 7200-word test data, over which recall was 80.8% and precision 89.5%.

Several studies have explicitly tackled DM identification in speech. Hirschberg and Litman (1993) applied a model based on intonational information to 34 DM types, and correctly classified 75.4% of their 878 classifiable tokens. Another model correctly classified 80.1% of the tokens based on human transcript and punctuation.

Siegel and McKeown (1994) proposed another transcript-based method, using decision tree classifiers constructed by a genetic algorithm, on a superset of the above data with 1,027 tokens. An interesting baseline score was obtained by a binary decision tree based only on the utterance-initial feature, which reaches 79.16% accuracy. The score of the best decision tree found by the genetic algorithm was only 79.20%. Although they did not improve performance over baseline, decision trees "discovered" some meaningful linguistic rules.

The relevance of machine learning techniques to DM identification was further emphasized by Litman (1996) in a set of experiments that extended and completed earlier studies by improving manually-derived classification models, using the same data set (34 DM types, 878 tokens). Litman used the C4.5 decision tree learner as well as an algorithm constructing sets of conditional rules. The features included prosodic features assigned by human an-

notators, textual features extracted from human transcripts, including correct punctuation, part of speech information assigned automatically, and the nature of the token itself. Most of the prosodic and textual models that were learned automatically outperformed corresponding models defined *a priori* by humans. The best performance using all available features was 16.9% error rate (83.1% accuracy) on the whole set.

DM identification was coupled to speech recognition, utterance segmentation, POS tagging, and repair correction by Heeman and Allen (1999). The best results are 97.26% recall, 96.32% precision, and 6.43% error rate, which was not, however, computed in the same sense as in the previous studies.

Comparison across studies is made difficult by the fact that the exact list of DMs differs from one study to another. In our study, only two DMs are contrasted, but they appear to be particularly multifunctional, hence difficult to disambiguate.

5 Description of the Data

The ratio between the number of targeted DM types (30–40) and the amount of data (often around 1,000 tokens) used in the previous studies did not allow for in-depth analysis of each DM, especially when a unique model was learned for all DMs. All studies except Heeman's were based on a monologue transcript (75 minutes, ca. 12,500 words), which was annotated by one or two linguists. Heeman's studies used transcripts from the TRAINS dialogue corpus, which contained 8,278 DMs among ca. 60,000 words. However, the exact list of DM types is not available (23 appeared as examples), nor the number of annotators or their agreement.

The data used here enables a more detailed study of *like* and *well* as a much larger number of occurrences is available. We use the ICSI Meeting Recorder Corpus of multi-party conversations, comprising transcripts of 75 meeting recordings with five to eight speakers (Janin et al., 2003). The meetings feature scientific discussions involving both native and non-native English speakers (52 in all). A distributional study and the *a posteriori* feature analysis show that there is no qualitative difference in the use of the two DMs by native vs. non-native but fluent speakers (Popescu-Belis and Zufferey, 2006, 6.3).

The recordings have a total duration of about 80 hours, corresponding to nearly 800,000 words in transcription. The segmentation into about 100,000 individual utterances is also available together with automatically generated word-level timing, based on forced alignment of transcript with audio, as well as indications of interruptions and unfinished utterances (Shriberg et al., 2004).

For this study, the DM and non-DM occurrences of the lexical items *like* and *well* were annotated by the two authors, with access to the dialogue transcripts and audio. In an experiment involving four non-expert annotators (Zufferey and Popescu-Belis, 2004), the observed inter-annotator agreement was $\kappa = 0.74$, but agreement between experts was not tested systematically. There are 4,519 occurrences of *like*, of which 2,052 (45%) serve as DMs, and 4,136 occurrences of *well*, of which 3,639 (88%) serve as DMs.

6 Features Used for DM Identification

The present method focuses on surface features only, since deeper analyses of an utterance seem to require in most cases the prior identification of DMs. For instance, it would not be realistic to assume the availability of a parse tree or of a deep semantic analysis of an utterance, as their construction would precisely require knowledge of DMs. However, joint models for POS tagging or parsing with DM identification could incorporate knowledge about DMs as presented here.

Lexical features model the words immediately preceding or following a DM candidate, and depend on the width of the lexical window ($2N$) and the minimal frequency (F) of words used as possible values. One feature is defined for each position with respect to the DM candidate: WORD($-N$), ..., WORD(-1), WORD($+1$), ..., WORD($+N$). The possible values of these variables are the words observed around the DM candidates, above a certain frequency F, or 'other', or 'none' if there is no such position in the utterance (this implicitly includes information about the candidate's position). For a window of width $N = 1$, i.e. using only WORD(-1) and WORD($+1$), the frequency thresholds of $F = 3$, $F = 10$ and $F = 20$ correspond respectively to 360, 150 and 90 word types as possible values.

DMs also have specific **positional and prosodic** properties, but not all the prosodic features are easy to extract automatically. The following features, derived from the forced-alignment segmentation at the word level and the ground truth segmentation into utterances, will be used: INITIAL: set to 'yes' if the candidate is the first word of an utterance, to 'no' otherwise; FINAL: set to 'yes-completed' if the candidate is the last word of a completed utterance, to 'yes-interrupted' if it is the last word of an interrupted utterance and to 'no' otherwise; PAUSE-BEFORE: the duration of the pause before the candidate, or 10 seconds if the utterance starts with it; PAUSE-AFTER: the duration of the pause after the candidate, or 10 seconds if it ends the utterance; DURATION: the duration of the candidate. The first two are positional features, while the latter three are very elementary prosodic or temporal features.

The following **speaker-related, sociolinguistic** features will also be used, with the following possible values: GENDER: 'female' or 'male'; AGE: an integer; EDUCATION: 'undergraduate', 'graduate', 'PhD', 'professor'; NATIVE: 'native' vs. 'non-native' English speaker; ORIGIN: 'UK', 'US East', 'US West', 'US other', and 'other'. Such features could be useful to a dialogue processing system that is used frequently by the same persons.

For each category, the features were selected based on potential linguistic and computational relevance. In addition, the TYPE feature represents the nature of the candidate DM, either *like* or *well*, allowing the two lexical items to be processed differently, as in (Litman, 1996).

7 DM Classifiers

The choice of a classifier for DM identification is constrained by the nature of the features: some are discrete while others are continuous; the lexical features are quite sparse and have an unclear impact on classification. Here, four types of classifiers were tested using the WEKA toolkit (Witten and Frank, 2000): Bayesian Networks (BN), Support Vector Machines (SVM), decision trees, and k-nearest neighbours ($k = 3$), which performed below the first three, so its results are not reported here.

Decision tree classifiers are made of nodes that test features of a DM-candidate, and of branches that correspond to the possible values of the features. Each terminal node is labelled with one of the two classes, DM or non-DM (Siegel and McKeown, 1994; Litman, 1996). Decision trees can be learned from training data using the C4.5 method (Quinlan, 1993), which accepts both discrete and continuous features. C4.5 constructs a nearly optimal decision tree classifier for the training data, that is, a tree that maximizes the number of correctly classified instances (CCIs) over the training data, but not necessarily recall, precision or *kappa*.

8 DM Identification Results

The best scores reached by the classifiers do not differ substantially in our experiments, as the 95%-confidence intervals computed using 10-fold cross-validation (training on 90% of the data and testing on 10%) are not disjoint. The best scores are obtained by a Bayesian Network that uses only the discrete features of the DMs—see first line of Table 1. Decision trees will be used preferentially below, as BN classifiers take longer to build and are more difficult to interpret than them, and their performance is only slightly higher.

8.1 Highest Scores vs. Baseline Scores

Baseline scores for DM identification are at least 50% CCIs because of the binary nature of the classification problem. As shown in the last three lines of Table 2, the majority classifier, which assigns to all candidates the type of the most frequent class observed in the training data reaches scores that are well above zero for at least three metrics out of five. Only κ appears to be insensitive to this bias.

Method	Test	CCIs (%)	κ	r	p	f
MAJ	*l*+*w*	65.75	0	.66	1	.79
	l	45.40	0	1	.45	.62
	w	87.99	0	1	.88	.94
ISM	*l*+*w*	70.55	.42	.64	.88	.74
	l	54.60	0	0	0	0
	w	87.98	0	1	.88	.94

Table 2: Baseline scores for the majority classifier (MAJ) and for an item-specific majority classifier (ISM), tested on *like* and *well* together (noted *l*+*w*), then separately for each item.

Method	Train	Test	CCIs (%)	κ	r	p	f
BN	*l*+*w*	*l*+*w*	**90.480**±.646	**.783**±.016	.957±.004	.904±.008	**.930**±.005
	l+*w*	*l*	84.009±1.431	.681±.028	.896±.012	.784±.021	.836±.014
	l+*w*	*w*	97.537±.456	.880±.021	.991±.004	.981±.005	.986±.003
SVM	*l*+*w*	*l*+*w*	**89.290**±.571	**.752**±.014	.964±.006	.884±.008	**.922**±.004
	l+*w*	*l*	82.908±1.216	.661±.023	.914±.016	.759±.020	.829±.013
	l+*w*	*w*	96.250±.841	.808±.037	.992±.005	.966±.009	.979±.005
C4.5	*l*+*w*	*l*+*w*	**88.862**±.511	**.751**±.011	.923±.007	.909±.006	**.916**±.004
	l+*w*	*l*	81.046±.885	.618±.018	.802±.020	.785±.013	.793±.013
	l+*w*	*w*	97.396±.443	.870±.026	.991±.002	.980±.005	.985±.002

Table 1: Best results obtained by three machine learning algorithms, trained and tested on *like* and *well* together (noted *l*+*w*), and then also tested separately on each item (noted *l* and respectively *w*). The three most significant metrics (scores in **bold**) yield clearly decreasing scores from the first to the third condition.

The use of the TYPE feature, allowing an item-specific majority classifier to distinguish between the lexical items *like* and *well*, increases the baseline scores (see ISM in Table 2). This classifier, based only on the following rules: "*like* is not a DM" and "*well* is a DM", reaches already $\kappa = 0.42$.

The scores of the four classifiers from Table 1 are significantly above the baseline. The fact that the best score is $\kappa = 0.78$ shows that automatic DM identification performances are in the same range as human inter-annotator agreement. The best scores are also higher than those obtained by the classifiers that use only a subset of features, as shown in the next sub-section.

The scores of the best BN classifier applied separately to *like* and *well* are also shown in Table 1, 2nd and 3rd lines. These are significantly higher for the identification of DM *well* ($\kappa = 0.880, f = 0.986$) than for DM *like* ($\kappa = 0.681, f = 0.836$). It is true that *well* as a DM is much more frequent than *like* as a DM (ca. 88% vs. 45%), so the baseline accuracy is higher for *well* (CCI = 88% vs. CCI = 45%, see 2nd and 3rd lines of Table 2) but this effect should be filtered out at least by the κ metric—nevertheless, which is still much higher for *well* than for *like*. *Well* appears thus to be easier to identify than *like*, with the features used here.

8.2 Relevance of the Features

The best-scoring decision tree uses four **lexical features** (WORD(-2), WORD(-1), WORD($+1$) and WORD($+2$)), their possible values being all the word types occurring at least 10 times in this 4-word lexical window ($F = 10, N = 2$). The best C4.5 learner was set to construct binary trees with at least two instances per leaf.

Four experiments were particularly informative. First, using only the WORD(-1) lexical feature, i.e. the lexical item preceding the candidate DM, C4.5 constructs trees that contain at the uppermost node the lexical collocations that are the most reliable indicators of a DM, with scores reaching CCI $= 86.5\%, \kappa = 0.68, r = 0.97, p = 0.85, f = 0.90$, which are not much below the best possible ones. When distinguishing *like* from *well* in the decision trees, thanks to the TYPE feature in addition to WORD(-1), the scores increase to CCI $= 87.4\%, \kappa = 0.72, r = 0.91, p = 0.90, f = 0.90$ (note the high value of κ).

Words situated after the candidate DM appear to be much less informative: if only TYPE and WORD($+1$) are used, CCI $= 77.8\%$ and $\kappa = 0.47$. When all lexical features encoded as WORD(n) are used ($n \leq 2$), the results are getting even closer to the best ones, but recall increases and precision decreases. The lexical features, and in particular the word before the candidate, appear thus to be nearly sufficient for DM identification of *like* and *well*. The actual values of WORD(n) that serve as lexical indicators are not, of course, the same for the two items.

Turning now to **positional and prosodic features**, experiments using combinations of one, two or three features are summarized in Table 3. A first series of experiments with positional features (left

Positional						Prosodic / temporal					
Features	CCIs(%)	κ	r	p	f	Features	CCIs(%)	κ	r	p	f
T	70.5	0.42	0.64	0.88	0.74	T	70.5	0.42	0.64	0.88	0.74
I	68.8	0.42	0.54	0.97	0.70	B	74.2	0.50	0.65	0.94	0.77
T+I	73.4	0.46	0.70	0.87	0.78	T+B	75.3	0.48	0.75	0.86	0.80
F	67.5	0.09	0.98	0.67	0.80	A	67.5	0.09	0.98	0.67	0.80
T+F	72.5	0.46	0.64	0.91	0.75	T+A	75.8	0.50	0.74	0.87	0.80
T+I+F	75.8	0.51	0.71	0.90	0.79	T+A+B	79.4	0.55	0.82	0.86	0.84

Table 3: Results with C4.5 decision trees using combinations of positional and prosodic / temporal features (T: TYPE, I: INITIAL, F: FINAL, B: PAUSE-BEFORE, A: PAUSE-AFTER).

part of the table) shows that on average, classification is improved as more features become available among the following: TYPE (T), INITIAL (I), and FINAL (F). These results are paralleled by a second series (right column), obtained with prosodic/temporal features, PAUSE-BEFORE (B) and PAUSE-AFTER (A), in which scores also increase when more features are available. As the second series uses features that implicitly encode more information than in the first one, superior results are obtained. The best decision trees using PAUSE-BEFORE contain the following rule: "*like* is a DM only when the pause before it is longer than 0.06 s", indicating that a pause approximately longer than 60 milliseconds is a good indicator of a DM. A similar value (though with a lower score) is found for the pause after DM *like*, while no effect was observed for *well*. In addition, other experiments have shown that DURATION is not a relevant feature. Prosodic features appear thus to be superior to positional ones, but remain inferior to lexical features.

The **sociolinguistic features** alone do not permit the construction of a classifier with a non-zero score if the two lexical items *like* and *well* are not distinguished. When they are, the best decision tree generated by C4.5 remains the majority classifier for *well* ("all occurrences are DMs") and a more refined classifier for *like*: a number of heavy DM-*like* users are identified, for which all occurrences of *like* that they produce are considered as DMs. The scores of this classifier are: $\mathrm{CCI} = 77.3\%, \kappa = 0.47, r = 0.88, p = 0.80, f = 0.84$. These values are clearly above the scores obtained using TYPE only.

A number of sociolinguistic features appear to be relevant in the case of *like* only (the baseline score being here $\kappa = 0$). Using EDUCATION, the best tree found by C4.5 reaches $\kappa = 0.39$ with the following rule: "if the speaker is an undergraduate or a graduate, consider all tokens of *like* as DMs; if the speaker is a post-doc or a professor, consider all tokens of *like* as non-DMs". A similar correlation ($\kappa = 0.40$) holds for the region of ORIGIN ('US West' implies heavy DM *like* user) and a stronger correlation ($\kappa = 0.44$) holds for AGE ('under 30' implies heavy DM user). These experiments thus bring statistical evidence that younger speakers from the US West tend to overuse *like* as a DM, which corroborates a view commonly held by sociolinguists, who often consider the DM *like* as a feature of adolescent speech (Andersen, 2001). Since in our data there were a majority of speakers under 30 from the US West, below PhD level, it is not possible to identify the precise feature that correlates with DM-*like* overuse among AGE, ORIGIN or EDUCATION)—more subjects are needed to "decorrelate" these features, though the present number (52) is sufficient to explore each feature in part.

8.3 Automatic Attribute Selection

Two methods were used to compare the merits of features, and appear to lead to similar results. WEKA's correlation-based feature subset selection algorithm (CFS) aims at finding the best subset of features by examining the individual predictive power of each feature and at the same time minimizing redundancy within the subset. Alternatively, independent relevance scores for each feature can be computed using two criteria: the information gain and χ^2 (Witten and Frank, 2000). Their rankings being very similar, only information gain is used here.

Like		*Well*	
Feature	IG	Feature	IG
WORD(−1)	.44	WORD(−1)	.39
WORD(+1)	.21	PAUSE-BEFORE	.23
SPEAKER	.15	INITIAL	.19
PAUSE-BEFORE	.06	WORD(+1)	.15
AGE	.06	PAUSE-AFTER	.10
PAUSE-AFTER	.05	FINAL	.10
EDUCATION	.04	SPEAKER	.04
INITIAL	.03	DURATION	.03
COUNTRY	.02	AGE	.01
FINAL	.01	COUNTRY	.005
GENDER	.01	EDUCATION	.004
DURATION	.01	NATIVE	.001
NATIVE	.001	GENDER	.001

Table 4: Separate ranking of features for *like* and *well* according to their information gain (IG). Significant IG decreases are indicated by a line.

The CFS algorithm finds the following optimal subset of attributes: {TYPE, PAUSE-BEFORE, INITIAL, WORD(−1)}, thus confirming previous observations. The word before the candidate is a key feature, along with the specific processing of each DM (TYPE), and the pause before the candidate (or its utterance-initial character).

The ranking of each feature shows that the most distinctive feature is the word before the candidate, WORD(−1), followed at some distance by PAUSE-BEFORE, INITIAL, WORD(+1) (the word after the candidate) and TYPE. The ranking can also be done separately with respect to *like* and *well*, as shown in Table 4. The lists are similar to the one just described for the joint identification task.

Attribute selection can also be used to determine the most discriminative collocations, i.e. the words that best indicate whether their neighbouring candidate is likely to be a DM or not DM (words must be used individually as features in this case). The best feature set found by CFS for *like* contains *something, things, seems* (if they precede *like*, then the occurrence is not a DM), or *that* (if it follows *like*, then the occurrence is not a DM). Similar trials focused only on *well* help to determine collocations such as *very well, as well, how well*, which are relevant to identify non-DM occurrences of *well*.

9 Discussion

To summarize, the best scores for *like* and *well* are: $\mathrm{CCI} = 90\%, \kappa = 0.78, r = 0.96, p = 0.90, f = 0.93$, obtained for a Bayesian Network; the best scores of a C4.5 decision tree or an SVM are not much lower. These scores are well above the baseline ones, although this depends on how the baseline is defined, as some very simple classifiers have scores that are well above zero. These scores also compare favourably to the ones obtained in previous studies, although the DMs and evaluation measures sometimes differ considerably.

The best scores obtained are comparable to inter-annotator agreement values observed for non-expert subjects ($\kappa = 0.74$). This indicates that automatic classifiers may have reached the highest possible performance in the present experiments, and that the set of features was sufficient to reach an accuracy comparable to human annotators. Improving the scores seems thus to require also a more reliable annotation, obtained for instance by allowing experienced annotators to discuss and to adjudicate their individual annotations.

The most important features appear to be the lexical collocations that can be learned from the training data. Among these, the word before a candidate DM is the most useful one, especially as it implicitly encodes also the utterance-initial character. Scores obtained using only lexical features are within 5% distance from the best overall scores. Decision trees based only on lexical features, or even on TYPE and WORD(−1) only, are not far from optimal ones. It is therefore surprising that these features were not used in Litman's (1996) study, maybe from lack of enough training data for each item.

Positional and prosodic features are significantly less efficient than lexical ones, when used alone, although they appear in the best decision trees just below lexical features. The sociolinguistic features are only slightly correlated to DM use, almost exclusively for *like*: the most reliable indicators are the identity and the age/education of the speakers.

The TYPE feature is crucial: *like* and *well* are much better processed separately than as a unique class. This conclusion confirms, on a large data set, the theoretical insights arguing that DMs are not a homogeneous class. Although some of the pre-

vious features do generalize to both lexical items (such as PAUSE-BEFORE), many of the features are item-specific, as found also by Litman (1996), and in particular the lexical features, which appeared to be the most relevant ones. Overall, this study has shown that DM identification can reach accuracies that are comparable to inter-annotator agreement scores, if item-specific classifiers using lexical features are trained on large corpora.

Future work should focus first on the application of the method to other ambiguous DM candidates, such as *you know*. This requires, for each item, the manual annotation of a sizeable amount of instances for training and test, and possibly some adaptation of the features. More elaborate prosodic features should also be studied. Finally, DM classifiers could be applied prior to POS tagging and parsing, or could be integrated into POS taggers or parsers.

Acknowledgments

This work has been supported by the Swiss National Science Foundation through the IM2 NCCR on Interactive Multimodal Information Management.

References

Gisle Andersen. 2001. *Pragmatic Markers of Sociolinguistic Variation*. John Benjamins, Amsterdam.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

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

Peter A. Heeman and James F. Allen. 1999. Speech repairs, intonational phrases and discourse markers: Modeling speakers utterances in spoken dialogue. *Computational Linguistics*, 25(4):1–45.

Julia Hirschberg and Diane Litman. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530.

Ben Hutchinson. 2004. Acquiring the meaning of discourse markers. In *Proceedings of ACL 2004 (42nd Annual Meeting of the Association for Computational Linguistics)*, pages 685–692, Barcelona, Spain.

Adam Janin, Don Baron, Jane A. Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, and Chuck Wooters. 2003. The ICSI Meeting Corpus. In *Proceedings of ICASSP 2003 (IEEE International Conference on Acoustics, Speech, and Signal Processing)*, Hong Kong, China.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Diane J. Litman. 1996. Cue phrase classification using machine learning. *Journal of Artificial Intelligence Research*, 5:53–94.

Daniel Marcu. 1998. A surface-based approach to identifying discourse markers and elementary textual units in unrestricted texts. In *ACL/COLING-98 Workshop on Discourse Relations and Discourse Markers*, pages 1–7, Montreal, Canada.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–140.

Andrei Popescu-Belis and Sandrine Zufferey. 2006. Automatic identification of discourse markers in multiparty dialogues. Working paper 65, ISSCO, University of Geneva, December 2006.

John R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Francisco, CA.

Rachel Reichman. 1985. *Getting Computers to Talk Like You and Me: Discourse Context, Focus, and Semantics (An ATN Model)*. MIT Press, Cambridge, MA.

Deborah Schiffrin. 1987. *Discourse Markers*. Cambridge University Press, Cambridge, UK.

Lawrence C. Schourup. 2001. Rethinking 'well'. *Journal of Pragmatics*, 33:1025–1060.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of SIGdial 2004 (5th SIGdial Workshop on Discourse and Dialogue)*, pages 97–100, Cambridge, MA.

Eric V. Siegel and Kathleen R. McKeown. 1994. Emergent linguistic rules from inducing decision trees: Disambiguating discourse clue words. In *Proceedings of AAAI 1994 (12th National Conference on Artificial Intelligence)*, pages 820–826, Seattle, WA.

Iain Witten and Eibe Frank. 2000. *Data Mining: Practical Machine Learning Tools with Java Implementations*. Morgan Kaufmann, San Francisco, CA.

Sandrine Zufferey and Andrei Popescu-Belis. 2004. Towards automatic identification of discourse markers in dialogs: The case of like. In *Proceedings of SIGdial 2004 (5th SIGdial Workshop on Discourse and Dialogue)*, pages 63–71, Cambridge, MA.

Detecting and Summarizing Action Items in Multi-Party Dialogue*

Matthew Purver[1], **John Dowding**[2], **John Niekrasz**[1], **Patrick Ehlen**[1], **Sharareh Noorbaloochi**[1] and **Stanley Peters**[1]

[1]Center for the Study of Language and Information
Stanford University, Stanford, CA, 94305 USA
{mpurver, niekrasz, ehlen, sharare, peters}@stanford.edu

[2]University of California at Santa Cruz
Santa Cruz, California, USA
jdowding@ucsc.edu

Abstract

This paper addresses the problem of identifying *action items* discussed in open-domain conversational speech, and does so in two stages: firstly, detecting the subdialogues in which action items are proposed, discussed and committed to; and secondly, extracting the phrases that accurately capture or summarize the tasks they involve. While the detection problem is hard, we show that we can improve accuracy by taking account of dialogue structure. We then describe a semantic parser that identifies potential summarizing phrases, and show that for some task properties these can be more informative than plain utterance transcriptions.

1 Introduction

Multi-party conversation, usually in the form of meetings, is the primary way to share information and make decisions in organized work environments. There is growing interest in the development of automatic methods to extract and analyze the information content of meetings in various ways, including automatic transcription, targeted browsing, and topic detection and segmentation – see (Stolcke et al., 2005; Tucker and Whittaker, 2005; Galley et al., 2003), amongst others.

In this paper we are interested in identifying *action items* – public commitments to perform a given task – both in terms of *detecting the subdialogues* in which those action items are discussed (along with the roles certain utterances perform in that discussion), and of *producing useful descriptive summaries* of the tasks they involve. While these summaries are the obvious end product in the first instance (perhaps presented as an automatically-prepared to-do list), subdialogue detection is also a useful output *per se*, as it allows users to browse the meeting recording or transcript in a targeted way.

Section 3 discusses the detection of subdialogues – short passages of conversation in which the action items are typically discussed, summarized, agreed and committed to – using a hierarchical classifier which exploits local dialogue structure. Multiple independent sub-classifiers are used to detect utterances which play particular roles in the dialogue (e.g. agreement or commitment), and an overall super-classifier then detects the critical passages based on patterns of these roles. We show that this method performs better than a flat, utterance-based approach; as far as we are aware, these are the first results for this task on realistic data.

Section 4 then investigates the production of summaries. For this, we use an open-domain semantic parser to extract phrases from within the utterances which describe one of two important properties: the *task* itself and the *timeframe* over which it is to be performed. We describe how such a parser can be built from generally available lexical resources and tailored to the particular problem of parsing speech recognition output, and show how a regression model can be used to rank the candidate parser outputs. For the timeframes, this produces

*This work was supported by the CALO project (DARPA grant NBCH-D-03-0010). We also thank Gokhan Tür, Andreas Stolcke and Liz Shriberg for provision of ASR output and dialogue act tags for the ICSI corpus.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 18–25,
Antwerp, September 2007.

more informative results than the alternative of presenting the entire 1-best utterance transcriptions.

2 Background

Subdialogue Detection User studies show that participants regard action items as one of a meeting's most important outputs (Lisowska et al., 2004; Banerjee et al., 2005). However, spoken action item detection seems to be a relatively new task. There is related work with email text: (Corston-Oliver et al., 2004; Bennett and Carbonell, 2005) both showed success classifying sentences or entire messages as action item- or task-related. Performance was reasonable, with f-scores around 0.6 for sentences and 0.8 for whole messages; the features used included lexical, syntactic and semantic features (n-grams, PoS-tags, named entities) as well as more email-specific features (e.g. header information).

However, applying the same methods to dialogue data is problematic. Morgan et al. (2006) applied a similar method to a portion of the ICSI Meeting Corpus (Janin et al., 2003) annotated for action items by Gruenstein et al. (2005). While they found that similar lexical, syntactic and contextual features were useful (together with other dialogue-specific features, including dialogue act type and prosodic information), performance was poor, with f-scores limited to approximately 0.3, even given manual transcripts and dialogue act tags. One major reason for this is the fragmented nature of conversational decision-making: in contrast to email text, the descriptions of tasks and their properties tend not to come in single sentences, but may be distributed over many utterances. These utterances may take many different forms and play very distinct roles in the dialogue (suggestions, commitments, (dis)agreements, etc.) and thus form a rather heterogeneous set on which it is hard to achieve good overall classification performance. For the same reasons, human annotators also have trouble deciding which utterances are relevant: Gruenstein et al. (2005)'s inter-annotator agreement was as low as $\kappa = 0.36$.

In (Purver et al., 2006), we proposed an approach to this problem using individual classifiers to detect a set of distinct action item-related utterance classes: *task description*, *timeframe*, *ownership* and *agreement*. The more homogeneous nature of these classes seemed to produce better classification accuracy, and action item discussions could be hypothesized using a simple heuristic to detect clusters of multiple classes. However, this was only evaluated on a small corpus of simulated meetings (5 c.10-minute meetings, simulated by actors given a detailed scenario), and only on gold-standard manual transcriptions. The first half of this paper applies that proposal to a larger, less domain-specific, naturally-occurring dataset, and also extends it to include the learning of a super-classifier from data.

Note that while previous work in the detection and modelling of *decisions* (Verbree et al., 2006; Hsueh and Moore, 2007) is related, the tasks are not the same. Firstly, our job is to identify public commitments to tasks, rather than general decisions about strategy, or decisions not to do anything (see e.g. Hsueh and Moore (2007)'s example Fig. 1). Secondly, our data is essentially open-domain, making e.g. simple lexical cues less useful than they are in a domain with repeated fixed topics. Note also that our results are not directly comparable with those of Hsueh and Moore (2007), who detect decision-making acts from a human-extracted summary rather than a raw meeting transcript, making positive examples much less sparse.

Summarization & Phrase Extraction Detecting subdialogues and utterances, though, is only part of the task – we need a succinct summary if we are to present a list of action items to a user. Ideally, this summary should contain at least the identity of the owner, a description of the task, and a specification of the timeframe. Ownership may occasionally be expressed by explicit use of a name, but is more often specified through the interaction itself – proposals of ownership usually either volunteer the speaker *"I guess I'll ..."* or request commitment from the addressee *"Could you maybe ..."*. Establishing identity therefore becomes a problem of speaker and addressee identification, which we leave aside for now, but see e.g. (Katzenmaier et al., 2004; Jovanovic et al., 2006; Gupta et al., 2007).

Timeframe and task, however, are expressed explicitly; but detecting the relevant utterances only gets us part of the way. Example (1) shows an utterance containing a task description:

(1) *What I have down for action items is we're sup-*

posed to find out about our human subject

Arguably the best phrase within this utterance to describe the task is *find out about our human subject*, as opposed to other larger or smaller phrases. Notably, although the utterance contains the phrase *action item* — likely a strong clue to the detection of this utterance as action item-related — this phrase itself is not particularly useful in a summary.

3 Subdialogue Detection

3.1 Approach

Following the proposal of (Purver et al., 2006), the insight we intend to exploit is that while the relevant utterances may be hard to identify on their own, the subdialogues which contain them do have characteristic structural patterns. Example (2) illustrates the idea: no single utterance contains a complete description of the task, and while some features (the phrases *by uh Tuesday* and *send it*, perhaps) might suggest action items, they may be equally likely to appear in unrelated utterances. However, the structure gives us more to go on: A proposes something involving B's agency, B considers it, and finally B agrees and commits to something.

(2) A: Well maybe by uh Tuesday you could
B: Uh-huh
A: revise the uh
C: proposal
B: Mmm Tuesday let's see
A: and send it around
B: OK sure sounds good

There are two ways in which this might help us with the detection task. Firstly, if these *action-item-specific dialogue acts* (AIDAs) form more homogeneous sets than the general class of "action-item-related utterance", we should be able to detect them more reliably. Secondly, if they are more-or-less independent, we can use the co-occurrence of multiple act types to increase our overall subdialogue detection accuracy.[1]

3.2 Data

Following (Purver et al., 2006), we take the relevant AIDA classes to be:

[1] In fact, there is a third: the different information associated with each act type helps in summarization – but see below.

D	*description*	discussion of the task to be performed
T	*timeframe*	discussion of the required timeframe
O	*owner*	assignment of responsibility (to self or other)
A	*agreement*	explicit agreement or commitment

Table 1: Action item dialogue act (AIDA) classes.

We annotated 18 meetings from the ICSI Meeting Corpus (Janin et al., 2003), recordings of naturally-occurring research group meetings. The meetings are divided up by subject area; our set contains 12 from one area and 6 from 4 further areas. Three authors annotated between 9 and 13 meetings each, with all three overlapping on 3 meetings and two overlapping on a further 4. Inter-annotator agreement improved significantly on (Gruenstein et al., 2005), with pairwise κ values for each individual AIDA class from 0.64 to 0.78. Positive examples are sparser, though, with only 1.4% of utterances being marked with any AIDA class. Note that while utterances can perform multiple AIDAs (see (2) above), there is a large degree of independence between the class distributions. Cosine distances between the distributions show high independence between A and all other classes, and reasonable independence for all other pairings except perhaps D-O (here, 0 represents total independence, 1 exact correlation):

A-T	A-D	A-O	T-D	T-O	D-O
0.06	0.03	0.07	0.23	0.29	0.55

Table 2: Between-class cosine distances.

3.3 Experiments

We trained 4 independent classifiers for the detection of each individual AIDA class; features were derived from various properties of the utterances in context (see below). We then trained a super-classifier, whose features were the hypothesized class labels and confidence scores from the sub-classifiers, over a 10-utterance window. In all cases, we performed 18-fold cross-validation, with each fold training on 17 meetings and testing on the remaining 1. All classifiers were linear-kernel support

vector machines, using *SVMlight* (Joachims, 1999).

We can evaluate performance at two levels: firstly, the accuracy of the individual AIDA sub-classifiers, and secondly, the resulting accuracy of the super-classifier in detecting subdialogue regions. The sub-classifiers can be evaluated on a per-utterance basis; it is less obvious how to evaluate the super-classifier as it detects windows rather than utterances, and we would like to give credit for windows which overlap with gold-standard subdialogues even if not matching them exactly. We therefore use two metrics; one divides the discourse into 30-second windows and evaluates on a per-window basis; one evaluates on a per-subdialogue basis, judging hypothesized regions which overlap by more than 50% with a gold-standard subdialogue as being correct.

As a baseline, we compare to a standard flat classification approach, as taken by (Morgan et al., 2006; Hsueh and Moore, 2007); we trained a single classifier on the same annotations, but for the simple binary decision of whether an utterance is action-item-related (a member of any AIDA class) or not.

3.4 Features

We extracted utterance features similar to those of (Morgan et al., 2006; Hsueh and Moore, 2007): n-grams, durational and locational features from the transcriptions; general dialogue act tags from the ICSI-MRDA annotations (Shriberg et al., 2004); TIMEX temporal expression tags using MITRE's rule-based TempEx tool; and prosodic features from the audio files using Praat. We also allowed "context" features, consisting of the same utterance features (suitably indexed) from the immediately preceding 5 utterances. Table 3 shows the complete set.

Lexical	ngrams length 1-3
Utterance	length in words & duration in seconds percentage through meeting
Prosodic	pitch & intensity min/max/mean/deviation pitch slope number of voiced frames
TIMEX	Number of time expression tags
MRDA	MRDA dialogue act class
Context	features as above for utts $i-1 \ldots i-5$

Table 3: Features for subdialogue detection.

However, use of lexical and dialogue act features brings up the question of robustness: ASR word error rates are high in this domain, and general dialogue act tagging accuracy low (Ang et al., 2005). We therefore investigated the use of ASR output (obtained using SRI's Decipher (Stolcke et al., 2005)) for lexical features, both via 1-best transcriptions and word confusion networks (WCNs), which encode multiple scored hypotheses for each word (Tür et al., 2002).[2] We also examined performance both with and without MRDA dialogue act tag features.

3.5 Results

Overall Performance with unigram, utterance and context features is shown in Table 4. While per-utterance results are still low (f-scores all below 0.3), commensurate with Morgan et al. (2006)'s results with flat classification, we see that the use of the super-classifier to detect subdialogue regions does give us results which might be of practical use, with overlap f-scores near 0.5. Words were the most useful feature, with no improvement gained by increasing n-gram length above 1; prosodic features give no improvement. While MRDA and TIMEX features do give small improvements at the sub-classifier level, we see no overall subdialogue accuracy gain – we are currently investigating whether super-classifier improvements can help with this.[3]

	Sub-classifiers				*Super-classifier*	
	D	T	O	A	30sec	Overlap
Recall	.19	.15	.21	.18	.51	.59
Precn.	.18	.46	.27	.16	.31	.37
F1	.19	.22	.24	.17	.39	.45

Table 4: Structured classifier; lexical + utterance features, 5-utterance context.

Baseline comparison Comparison with the flat baseline classifier (Table 5) shows that the structured approach gives a significant advantage; we hypothesize that this is because commitments in dialogue arise via the interaction itself as much as from individual utterances. Interestingly, although our approach consistently outperforms the baseline,

[2] While we do not know the exact ASR word error rate on our meeting set, Stolcke et al. (2005) report 24% WER on meetings from the same corpus.

[3] Note that although accuracies are much lower than those reported by Hsueh and Moore (2007), the tasks are not the same: in particular, they detect relevant dialogue acts from a manually extracted summary, rather than a whole meeting. See Section 2.

the delta decreases as more contextual information becomes available – Figure 1 shows how f-scores vary as a unigram feature set is expanded to include unigrams from preceding utterances. It may be that contextual features implicitly provide some of the structural information explicitly modelled in the structured approach. We plan to investigate this effect on larger datasets when available.

	30sec			*Overlap*		
	Re	Pr	F1	Re	Pr	F1
Structured	.51	.31	.39	.59	.37	.45
Flat	.65	.23	.34	.64	.24	.35

Table 5: Classifier comparison; lexical + utterance features, 5-utterance context.

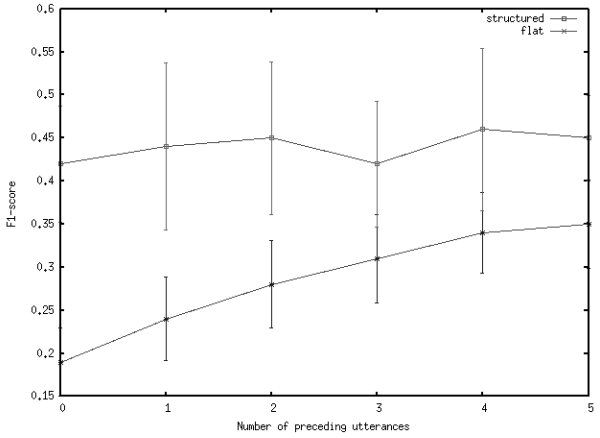


Figure 1: F-scores for structured vs. flat classifiers with 95% confidence bars; unigram features from increasing numbers of utterances in context.

Robustness Investigation of the effect of ASR output shows a drop in overlap f-score of 8-9% (absolute) or 17-20% (relative) – see Table 6. Use of WCNs improves over 1-best hypotheses by 1-2%. While this is a large drop, we are encouraged by the fact that this overall loss in accuracy is smaller than the loss at the sub-classifier level, where f-scores drop by around 35% on average, and up to 50% (relative). This suggests that the presence of multiple independent sub-classifiers is able (to some extent, at least) to make up for the drop in their individual performance. As more data becomes available and sub-classifier performance becomes more robust, we anticipate better overall results.

	Structured					Flat	
	Sub-classifiers				*Super*		
	D	T	O	A	O'lap	Utt	O'lap
Manual	.19	.22	.24	.17	.45	.24	.35
1-best	.16	.19	.15	.11	.36	.19	.32
WCNs	.15	.14	.18	.07	.37	.19	.33

Table 6: F1-scores against ASR type; lexical + utterance features, 5-utterance context.

Comparison to the baseline flat classifier shows that the structured approach is less robust (unsurprisingly, perhaps, given its more complex nature); the relative drop in the baseline overlap f-scores is lower. However, the resulting absolute performances are still higher for the structured approach, although the difference is no longer statistically significant over the number of meetings we have.

Summary We see that using our discourse-structural approach gives significantly improved performance over a comparable flat approach when using manual transcripts. While there is a drop in performance when using (highly errorful) ASR output, performance is still above the baseline.

4 Parsing and Summarization

We now turn to the second task: extracting useful phrases for summarization.

4.1 Approach

To extract timeframe and task descriptions, we exploit the fact that the critical phrases which contain them display certain characteristic syntactic and semantic features. Since the meeting topics and tasks are not known in advance, we expect that any approach which learns these features purely from a training set is unlikely to generalize well to unseen data. We therefore use a general rule-based parser with an open-domain, broad-coverage lexicon. The grammar, however, is small: as our data is highly ungrammatical, disfluent and errorful, we have developed a semantic parser that attempts to find basic predicate-argument structures of the major phrase types S, VP, NP, and PP, not necessarily trying to find larger structures (such as coordination and relative clauses) where reliability would be low.

Lexical Resources Our lexicon is built from publicly available lexical resources for English, including COMLEX, VerbNet, WordNet, and NOMLEX. Others have shared this basic approach (Shi and Mihalcea, 2005; Crouch and King, 2005; Swift, 2005).

COMLEX (Grishman et al., 1994) provides detailed morphological and syntactic information for the 40,000 most common words of English, as well as basic lexical information (e.g. adjective gradability, verb subcategorization, noun mass/count nature). VerbNet (Kipper et al., 2000) provides semantic information for 5,000 verbs, including frames and thematic roles, along with syntactic mappings and selectional restrictions for role fillers. WordNet (Miller, 1995) then provides us with another 15,539 nouns, and the semantic class information for all nouns. These semantic classes are hand-aligned to the selectional classes used in VerbNet, based on the upper ontology of EuroWordNet (Vossen, 1997). NOMLEX (Macleod et al., 1998) provides syntactic information for event nominalizations and a mapping from noun arguments to VerbNet syntactic positions; this allows us to give nominalizations a semantics compatible with verb events, and assert selectional restrictions. To add proper names, we used US Census data for people, KnowItAll (Downey et al., 2007) for companies, and WSJ data for person and organization names. Proper names account for about 1/3 of the entries in the lexicon.

These resources are combined and converted to the Prolog-based format used in the Gemini framework (Dowding et al., 1993), which includes a fast bottom-up robust parser in which syntactic and semantic information is applied interleaved. To facilitate extracting semantic features, we use Minimal Recursion Semantics (Copestake et al., 2005), a flat semantic representation; we have also modified Gemini to parse WCNs as well as flat transcriptions. Gemini computes parse probabilities on the context-free background of the grammar; in these experiments, probabilities were trained on WSJ data.

4.2 Experiments

Our parsing approach intentionally produces multiple short fragments rather than one full utterance parse. Combining this with the high number of paths through a WCN means that our primary problem is to extract a few useful phrases from amongst a very high number of alternatives. We approached this as a regression problem, and attempted to learn a model to rank phrases according to their likelihood of appearing in an action item description (again using *SVMlight*). We cross-validated over the same 18-meeting dataset, considering only those utterances manually annotated as containing timeframe and task descriptions (the T and D AIDA classes). To provide target phrases for evaluation, annotators marked those portions of the manual utterance transcriptions which should be extracted (note that these often do not match any WCN path exactly).

For each segment returned by the parser we extracted features of three general types: properties of the raw WCN paths, properties of the parsed phrases, and lexical features reflecting the identity of the words themselves – a list is given in Table 7. As lexical features are likely to be more domain-specific, and increase the size of the feature space dramatically, we prefer to avoid them if possible. Initial feature selection experiments indicate that the most useful features are acoustic probability, phrase type and verb semantic class, suggesting that syntactic and semantic information are indeed valuable.

WCN	phrase length (words & WCN arcs) start/end point (absolute & percentage) acoustic probability acoustic probability shortfall (delta below highest probability for this segment)
Parse	parse probability phrase type (S/VP/NP/PP) main verb VerbNet class head noun WordNet synset nominalization (yes, no) number of thematic roles filled noun class of *agent* thematic role (if any)
Lexical	main verb head noun all unigrams in the phrase
TIMEX	Number of time expression tags

Table 7: Features for parse fragment ranking.

4.3 Results

Choosing an evaluation metric is not straightforward: standard parse evaluation methods (e.g. checking crossing brackets against a treebank) are not applicable to our task of choosing useful fragments. Instead, we evaluate success based on how much of the human-annotated task descriptions are covered by the top-ranked fragment chosen by the

regression model. For recall we take the total proportion of the desired description covered; for precision, the total proportion of the chosen phrase which overlaps with the desired description; we then produce a corresponding f-score. We compare to a baseline of using the entire 1-best utterance transcription, and the ideal ceiling of choosing the fragment with the best f-score (still less than 1, due to ASR errors and parse segmentation). For timeframe utterances, we also compare to a second baseline of using those fragments of the 1-best transcription tagged as TIMEX expressions.

Results are shown in Table 8 for *timeframe* phrases, and Table 9 for *task description* phrases. For timeframes, the best feature set gives an f-score of .51 and precision of .62, outperforming both baselines but still some way below the ideal ceiling. Semantic classes and phrase-head lexical features help performance, although including other unigrams did not; TIMEX tags help, although a TIMEX-only baseline does badly.

	Recall	Precision	F1
Baseline 1: TIMEX	.26	.36	.31
Baseline 2: 1-best	.76	.27	.39
No sem/lex features	.33	.47	.38
+ semantic classes	.36	.53	.43
+ head verb/noun	.39	.59	.47
+ TIMEX	.43	.62	.51
Ceiling: best F1	.64	.80	.71

Table 8: Fragment ranking results: *timeframe*.

However, results for description phrases are poor, with no feature set outperforming the baseline. This is partly as the baseline recall is already quite high; note that using the parser does increase precision. Lexical features actually harm performance, perhaps unsurprisingly given the wider range of vocabulary compared to timeframes. The problem is also more difficult, hence the ideal figures are lower too; but inspection of errors suggests that inaccurate sentence segmentation (based only on pause length in these data) causes many of the problems, with many utterances annotated as providing only single words to the ideal phrase. We expect that improved sentence segmentation will improve performance, and are currently investigating this.

	Recall	Precision	F1
Baseline: 1-best	.66	.32	.43
No sem/lex features	.22	.41	.29
+ semantic classes	.35	.41	.38
+ head verb/noun	.31	.41	.35
Ceiling: best F1	.50	.78	.61

Table 9: Fragment ranking results: *description*.

5 Conclusions & Future Work

Both problems are hard, and overall performance is correspondingly lower than that achieved on less difficult tasks or less sparse data. However, they do appear tractable, even on errorful ASR output, with some encouraging initial performances obtained. Importantly, we have shown the benefits of using discourse structure in classification, and semantic features in summarization.

To improve detection performance, we are investigating more effective super-classifiers, incorporating existing task lists to provide reliable information about possible tasks to be discussed, and leveraging user interaction for learning – allowing users to confirm, delete or edit hypothesized action items, and using this as feedback to allow incremental learning (Purver et al., 2007).

For summarization, one of the major limitations of our approach is that we only consider phrases from within a single acoustically-segmented utterance, while many ideal descriptions combine information from more than one. We plan to investigate improved segmentation, and generation of summaries from multiple utterances.

References

J. Ang, Y. Liu, and E. Shriberg. 2005. Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings of ICASSP*.

S. Banerjee, C. Rosé, and A. 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 (CHI)*.

P. N. Bennett and J. Carbonell. 2005. Detecting action-items in e-mail. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

A. Copestake, D. Flickinger, C. Pollard, and I. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language and Computation*, 3(2–3):281–332.

S. Corston-Oliver, E. Ringger, M. Gamon, and R. Campbell. 2004. Task-focused summarization of email. In *Proceedings of the 2004 ACL Workshop Text Summarization Branches Out*.

R. Crouch and T. King. 2005. Unifying lexical resources. In *Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*.

J. Dowding, J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. 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)*.

D. Downey, M. Broadhead, and O. Etzioni. 2007. Locating complex named entities in web text. In *Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*.

M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. 2003. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*.

R. Grishman, C. Macleod, and A. Meyers. 1994. COMLEX syntax: Building a computational lexicon. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING)*.

A. Gruenstein, J. Niekrasz, and M. Purver. 2005. Meeting structure annotation: data and tools. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*.

S. Gupta, M. Purver, and D. Jurafsky. 2007. Disambiguating between generic and referential "you" in dialog. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*.

P.-Y. Hsueh and J. Moore. 2007. What decisions have you made?: Automatic decision detection in meeting conversations. In *Proceedings of NAACL/HLT*.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. 2003. The ICSI meeting corpus. In *Proceedings of the 2003 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

T. 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.

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 European Chapter of the ACL (EACL)*.

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*.

K. Kipper, H. T. Dang, and M. Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI)*.

A. Lisowska, A. Popescu-Belis, and S. Armstrong. 2004. User query analysis for the specification and evaluation of a dialogue processing and retrieval system. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*.

C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves. 1998. NOMLEX: A lexicon of nominalizations. In *Proceedings of EURALEX 98*.

G. A. Miller. 1995. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

W. Morgan, P.-C. Chang, S. Gupta, and J. M. Brenier. 2006. Automatically detecting action items in audio meeting recordings. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*.

M. Purver, P. Ehlen, and J. Niekrasz. 2006. Detecting action items in multi-party meetings: Annotation and initial experiments. In S. Renals, S. Bengio, and J. Fiscus, editors, *Machine Learning for Multimodal Interaction: Third International Workshop, MLMI 2006, Revised Selected Papers*, volume 4299 of *Lecture Notes in Computer Science*, pages 200–211. Springer.

M. Purver, J. Niekrasz, and P. Ehlen. 2007. Automatic annotation of dialogue structure from simple user interaction. In *Proceedings of the 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI'07)*.

L. Shi and R. Mihalcea. 2005. Putting pieces together: Combining FrameNet, VerbNet and WordNet for robust semantic parsing. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.

E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. 2004. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*.

A. Stolcke, X. Anguera, K. Boakye, Ö. Çetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng. 2005. Further progress in meeting recognition: The ICSI-SRI spring 2005 speech-to-text evaluation system. In *Proceedings of the Rich Transcription 2005 Spring Meeting Recognition Evaluation*.

M. Swift. 2005. Towards automatic verb acquisition from VerbNet for spoken dialog processing. In *Proceedings of the Interdisciplinary Workshop on the Identification and Representation of Verb Features and Verb Classes*.

S. Tucker and S. Whittaker. 2005. Reviewing multimedia meeting records: Current approaches. In *Proceedings of the 2005 (ICMI) International Workshop on Multimodal Multiparty Meeting Processing*.

G. Tür, J. Wright, A. Gorin, G. Riccardi, and D. Hakkani-Tür. 2002. Improving spoken language understanding using word confusion networks. In *Proceedings of the 7th International Conference on Spoken Language Processing (INTERSPEECH - ICSLP)*.

A. Verbree, R. Rienks, and D. 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, September 11 2006, Frontiers in Artificial Intelligence and Applications*, volume 144.

P. Vossen. 1997. EuroWordNet: a multilingual database for information retrieval. In *Proceedings of the 1997 DELOS Workshop on Cross-language Information Retrieval*.

Detecting Arguing and Sentiment in Meetings

Swapna Somasundaran
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
swapna@cs.pitt.edu

Josef Ruppenhofer
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
josefr@cs.pitt.edu

Janyce Wiebe
Dept. of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
wiebe@cs.pitt.edu

Abstract

This paper analyzes opinion categories like Sentiment and Arguing in meetings. We first annotate the categories manually. We then develop genre-specific lexicons using interesting function word combinations for detecting the opinions. We analyze relations between dialog structure information and opinion expression in context of multi-party discourse. Finally we show that classifiers using lexical and discourse knowledge have significant improvement over baseline.

1 Introduction

In this work, we bring together two areas of research which have seen great interest in recent times. Multi-party meetings have been analyzed with regard to dialog acts, hotspots, argumentation and decision points. Similarly, there is increasing activity in the automatic extraction of opinions, emotions, and sentiments in text (*subjectivity*) to provide tools and support for various NLP applications.

We believe that opinion information can enhance an interactive agent's ability to moderate a meeting; enable a summarizer to specifically report those opinions that influenced the decisions; and enhance the capabilities of Question Answering (QA) systems. As an example, consider a meeting from the AMI corpus (Carletta et al., 2005) where the participants have to design a new TV remote control. The following opinions are expressed regarding the TV remote:

U1. "It *[the remote]* is not as fast as a usual remote control"

U2. "That *[remote feature]* will be harder to learn"

U3. "We'll definitely won't go with that one *[speech recognition]*"

U4. "We can skip speech recognition directly, because it's not reachable for twenty five Euros".

Somebody who missed the meeting and had to find out details about the decisions made, may want to ask questions like:

Q1. "Why was the remote not rated highly?"

Q2. "Who argued against the speech recognition?"

Q3. "What were the points of persuasion against the speech recognition feature?"

Question *Q1* is answered by Utterances *U1* and *U2*, which express sentiments toward the remote. *Q2* is best answered by retrieving the names of all participants who had utterances similar to *U3* and *U4*. Similarly, *U4*, where the speaker is arguing for skipping the speech recognition would be a relevant answer for *Q3*. In order to be able to answer such questions, we explore two particular sub-types of subjectivity: Sentiment and Arguing. In the example utterances above, *U1* and *U2* express Sentiments, while *U3* and *U4* show speakers Arguing for their views. These subjectivity subtypes have proven useful for Question Answering on online multi-party debates (Somasundaran et al., 2007).

There has been a fair amount of work on the Sentiment category. By contrast, little work has been done on the Arguing category. We first define and annotate these opinion types in AMI meetings. We then perform inter-annotator agreement studies to verify if the two categories can be reliably detected.

We develop an Arguing lexicon as a new knowl-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 26–34,
Antwerp, September 2007.

edge source for automatically recognizing the Arguing category. We use previously developed lexicons for Sentiment detection (Wilson et al., 2005; Stone et al., 1966) to evaluate their portability to multi-party meetings. Previous efforts in recognizing opinions (or subjectivity) in monologic texts have focussed on knowledge from lexico-syntactic sources. While these have proven useful, we believe that in the conversational genre, reliably recognizing opinion expressions in utterances is a complex discourse task. Thus, we explore the novel use of dialog features for opinion recognition in combination with a lexicon. We find that this combination of knowledge sources shows promising results.

The rest of the paper is organized as follows: We introduce the data in Section 2 and our opinion definitions in Section 3. Then in Section 4 we present our annotation categories. In section 5 we explain the knowledge sources used for classification and present our experimental results in Section 6. Related work is discussed in Section 7 and finally we conclude in Section 8.

2 Data

For this work, we annotated 7 scenario-based team meetings from the AMI corpus resulting in a corpus of 4302 segments (6504 sentences) for our supervised learning experiments. In these meetings, four participants collaborate to design a new TV remote control in a series of four meetings, which represent different project phases, namely project kick-off, functional design, conceptual design, and detailed design.

In order to make the best use of the annotators' time in this work, we decided not to annotate the kick-off meetings as we believe them to be less rich in our opinion categories.

Each meeting in the AMI corpus comes with rich transcription and is annotated with dialog acts, argumentation, topics, etc. The corpus provides segment (turn) information for each speaker. Based on the rich transcriptions, we split the segments further into sentences. Sentence level classification tasks have a finer granularity and are of interest for applications like QA. On the other hand, in the absence of sentence boundary information, real time ASR systems work at the segment level. As there is interest at both levels of granularity, we present results at both the segment and sentence levels in this paper.

Some of the AMI annotations that are of interest in this work are Dialog Acts and their Adjacency Pairs. The AMI meeting is annotated with 15 Dialog Act (DA) categories: *Backchannel, Stall, Fragment, Inform, Elicit-Inform, Suggest, Offer, Elicit-Offer-Or-Suggestion, Assess, Elicit-Assessment, Comment-About-Understanding, Elicit-Comment-Understanding, Be-Positive, Be-Negative, Other*. Two DAs may be linked via an Adjacency Pair (AP) relation. One of the DAs is the source and the other is the target in the AP. There are 5 AP types, namely: *Support/Positive Assessment, Objection/Negative Assessment, Uncertain, Partial agreement/support, Elaboration*.

3 Opinion Definition

Our two opinion types are adapted from the work on attitude categories in monologic texts by Wilson et al (2005). They are defined as follows:

Sentiment: Sentiments include emotions, evaluations, judgments, feelings and stances. For example in the sentence "This idea is **good**", "good" expresses the sentiment.

Arguing: Arguing includes arguing for something, arguing that something is true, or should be done. Arguing brings out the participant's strong conviction and/or his attempt to convince others.

In multi-party discourse, speakers argue for something in a variety of ways. As arguing opinions are less well studied, we will examine some examples. Consider the following utterances, where the lexical anchors that indicate Arguing are shown in bold.

A1. "**I think** this idea will work"

A2. "This is the lightest remote **in the world**"

A3. "We **ought** to get this button"

A4. "**Clearly**, we cannot afford to use speech recognition"

A5. "It would be nice **if we could** have the curved shape"

A6. "I brought this up **because** this will affect the cost"

A7. "**We want** a fancy look and feel"

In *A1*, the speaker argues by explicitly stating his conviction. In *A2*, the speaker simply asserts his argument, while in *A3* the speaker argues for getting the button by framing it as a necessity. In *A4*, the speaker states his proposition categorically to argue

for it. Interestingly, in face to face conversations, participants also use persuasive constructs, justification or communal desire to argue for something as in *A5*, *A6* and *A7* respectively.

In examples *A1* to *A7* above, there are overt lexical anchors that indicate an arguing intent in the speaker's utterance. However, context, in addition to lexical clues is needed to infer that arguing is taking place. As part of a casual conversation, the utterance "I think John was at home" would not be Arguing, despite the presence of "I think". However, in a debate about John's whereabouts at the time of a murder, the sentence could function as Arguing. Here the context and the knowledge that there is a disparity between the speakers helps us infer that the sentence is intended to argue. Finally, sometimes arguing is done even in the absence of any overt lexical anchor. Consider:

A8. "The speech recognition is nice. Yes, speech recognition. It falls within our price range too"

In *A8*, we do not find any explicit markers. However, the speaker attempts to win approval for the speech recognition by his affirmation and his positive evaluations (sentiment) of the speech recognition and its price. These various elements together build up the argument.

4 Annotation Categories

Our annotation categories are Sentiment and Arguing. We discuss the varied ways of arguing in our annotation guide to help the annotators. As explained in Example *A8* of Section 3 sometimes Arguing is done without overt lexical anchors, which makes such cases difficult to annotate reliably. We assign these cases to a special category called Utterance Arguing.

We adapt the basic annotation frame for our opinion type from (Wilson and Wiebe, 2005). The relevant components of the frame are:

- **Text span:** The span of text that captures the opinion type. In the case of Utterance Arguing, this text span may cover the whole utterance.
- **Inferred: (true/ false)** This feature indicates that the annotator used inference for this annotation. For example, "very dark" is labeled as Sentiment in the sentence "This (TV) remote is **very dark**". This annotation is based on the knowledge that participants consider a dark color undesirable for the remote.
- **Annotator Confidence: (certain/ uncertain)** The annotators set this feature to uncertain when they are unsure of the annotation.

	Sentiment	Arguing	UtteranceArguing
segments	0.826	0.716	0.372
sentences	0.789	0.677	0.326
	Ignoring Annotator-uncertain cases		
segments	0.838	0.716	0.382
sentences	0.805	0.677	0.332
	Ignoring Annotator-uncertain and Inferred cases		
segments	0.85	0.716	0.382
sentences	0.814	0.677	0.332

Table 1: Kappa values for Inter-annotator agreement

4.1 Inter-annotator Agreement

Two annotators (two of the authors) underwent 3 rounds of training. Then we calculated inter-annotator agreement using Cohen's kappa over a previously unseen meeting (607 segments, 1002 sentences). Although the annotators tag expressions, agreement is calculated over the segment or the sentence. For this purpose, we assign a segment (or sentence) the labels of all the expressions annotated within it.

Table 1 shows the results of the agreement study. Our inter-annotator kappa values are in the Substantial Agreement Range according to Landis and Koch (1977) for Sentiment and Arguing, and in the Fair Agreement Range for Utterance Arguing. For Sentiment, when we exclude the labels from those instances that were tagged as inferential or uncertain, the agreement numbers go up to 0.85 for the segment and 0.814 for the sentence level respectively.

Compared to Sentiment, Arguing has lower kappa values at at 0.716 at the segment and 0.677 at the sentence level. We do not see any changes in the values when uncertain cases are removed. In this meeting the segments or sentence unit typically contain multiple expressions tagged for Arguing. Thus if an arguing label marked as uncertain was excluded from a given unit, but the unit had another label marked as certain elsewhere, then that unit overall still got an arguing label which counted toward the kappa calculation.

As expected, the Utterance Arguing category proved to be difficult. This is because it requires the annotators to infer whether the speaker is arguing when the utterance does not have any definite markers.

5 Knowledge Used in Classification

In this section, we discuss the development of our lexicon and the rationale for using dialog structures as knowledge sources for our automatic classifiers.

Much work in sentiment and subjectivity detection in monologic texts has focussed on lexical and syntactic features. In order to capture the lexical information we use lexicons. In the context of multiparty meetings, we hypothesize that the discourse flow and participant interaction act as useful indicators of opinion expression. We use Dialog Acts (DA) and Adjacency Pair (AP) features to capture the flow of discourse. We also believe that the lexical and discourse knowledge are complementary, and we build a system using all the features to test this hypothesis.

5.1 Sentiment Lexicon

We availed ourselves of previous work on Sentiment lexicon development, namely the General Inquirer (GI) (Stone et al., 1966), and Wilson et al's (2005) Subjectivity Clue list. The former provides a list of positive and negative words, while the latter contains a list of word and expressions that are strong/weak indicators of subjectivity, valence shifters, or intensifiers. In all, this gives us 6 lexicon categories to which a sentiment word may belong: GI Positive, GI Negative, Strong Subjective Clue, Weak Subjective Clue, Intensifier, and Valence Shifter.

5.2 Arguing Lexicon

We assembled an Arguing lexicon for meetings as follows. We inspected one AMI meeting (not used for training or testing) for words, phrases or word patterns that are indicative of Arguing. Then we explored the ICSI Meeting Recorder Dialog Act (MRDA) corpus (75 meetings, 72 hours) for similar expressions in order to develop more general patterns and increase the coverage of the lexicon. This was done in two steps. In the first, all instances of certain Dialog Act types *(dispreferred answer, negative answer, command, defending/explanation, suggestion)* were extracted and frequent n-grams $(1 \leq n \leq 4)$ identified. In the second phase, we manually inspected, for the highest ranking n-grams, a sample of 10-15 actual instances in the ICSI corpus and retained those n-grams that seemed promising. Finally, we looked over three ICSI transcripts in full to assess the coverage of the annotation concepts to be applied to the AMI data. This process produced a lexicon of 226 entries, sorted into 18 categories such as necessity, conditional, emphasis, generalization, contrast, causation, etc. to account for the various ways in which speakers argue.

Type	Example
emphasis	that's why the thing is
necessity	ought to had better
inconsistency	except that it's just that

Table 2: Examples from Arguing lexicon

As the entries given in Table 2 suggest, closed-class items such as modal verbs, adverbs, or conjunctions play a more important role in identifying instances of our Arguing class than open-class items. For instance, words like "oppose", "support", and "conclude" which directly denote aspects of arguing and reasoning are rare, whereas causal connectives such as "so", "because", and "if" are frequent.

We can understand the importance of closed-class items in terms of the distinction that Wiebe (2002) makes between direct subjective elements and expressive subjective elements. Direct subjective elements are exemplified, in the sentiment domain, by words like "love" or "criticize" which directly denote a particular kind of private state of a source, possibly in relation to a target, and which can realize their source and, if present, their target as a syntactic dependent. Expressive subjective elements, exemplified by words like "jerk" and "annoyingly", presuppose but do not denote a private state and cannot occur in syntactic construction with the source of the private state. Instead, the source is to be identified by the hearer from the candidate set made up by the interlocutors and the human referents in the discourse.

Applying this distinction to the Arguing category, we find that in the spoken conversation of meetings, where arguments are constructed in real-time, expressive subjective elements are prominent, with the

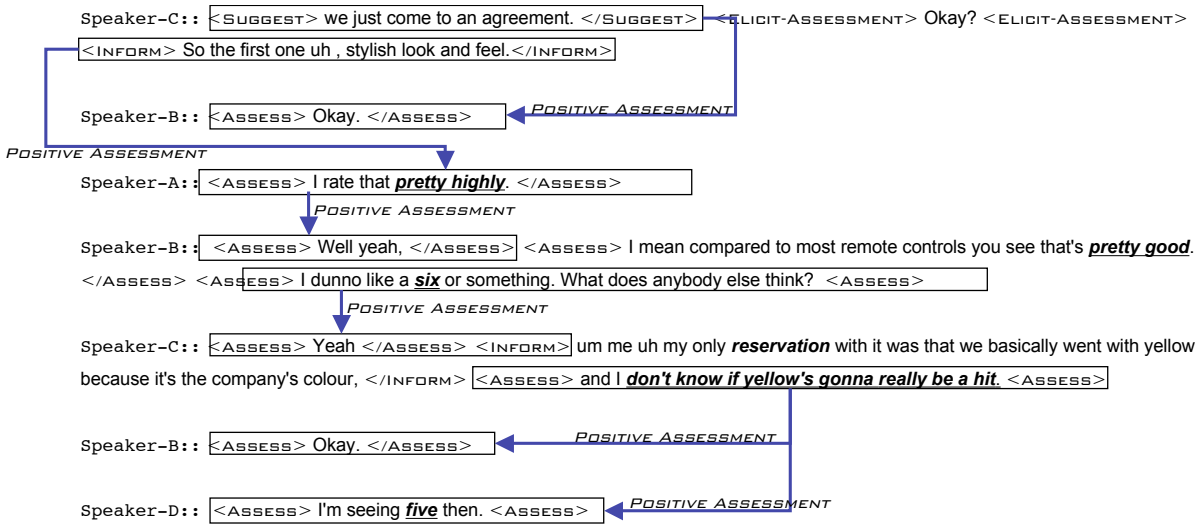


Figure 1: Sentiment expression and discourse flow.

sources typically being the speakers. This makes sense in particular for modal verbs such as "must", "need", etc. as arguing directly concerns modality: speakers discuss what is, what could be, what should be. By contrast, we find fewer direct subjective elements such as "require" or "argue". These elements, however, seem very suitable for reporting on arguments.

5.3 Dialog acts and Adjacency pairs

We observe that there is an interplay between our opinion categories and the dialog level annotations in the AMI corpus. Consider the following AMI meeting snippet where the participants rate their TV remote control design on a number of metrics such as learnability, look and feel, etc. using a scale from one (worst) to seven (best).

> *Speaker-C::* we just come to an agreement. Okay? So the first one uh , stylish look and feel .
> *Speaker-B::* Okay.
> *Speaker-A::* I rate that pretty highly.
> *Speaker-B::* Well yeah, I mean compared to most remote controls you see that's pretty good. I dunno like a six or something. What does anybody else think?
> *Speaker-C::* Yeah um me uh my only reservation with it was that we basically went with yellow because it's the company's colour , and I don't know if yellow's gonna really be a hit.
> *Speaker-B::* Okay.
> *Speaker-D::* I'm seeing five then.

Figure 1 illustrates the opinion annotations (in bold underlined text spans), DA annotations (as enclosing XML tags) and AP annotations (as directed links between segments) of the above meeting snippet. *C* introduces the first metric for evaluation, the stylish look and feel. *A* has a positive *Sentiment* about the remote in this regard and hence says he rates it "pretty highly". *B* shares *A's* positive *Sentiment*. He too evaluates the remote favorably and judges it as deserving a rating of six. Note that here "six" is considered an inferred sentiment, as it reflects the participant's evaluation of the remote. *C*, however, shows his negative *Sentiment* towards the remote by pointing out his reservation about the choice of the color yellow. *C's Sentiments* convince *D*, who then evaluates the look and feel at the lower grade of 5.

The Dialog Acts and Adjacency Pairs that capture the exchanges between the participants are indicative of the Sentiments expressed. For example, it is likely that a participant who has a positive evaluation of an object might positively assess his preceding speaker's positive assessment of the same object. We see this in Figure 1 when *A* and *B* both show positive *Sentiment* towards the remote's look and feel. *B* shows a *Positive Assessment* of *A's Assessment*. *D*, who evaluates the look and feel of the remote at a lower grade (negative *Sentiment*) has a *Positive Assessment* toward *C's Assessment* (negative *Sentiment*) of the remote. Thus the participants' sentiments towards objects are also reflected in their interpersonal dialog acts and vice versa. We also

found interesting relations bettween arguing and dialog structure. Due to space considerations, this is discussed in Appendix A

We believe Dialog structure (DA and AP) and our opinion categories are complementary rather than interchangeable. Dialog acts are focused on interpersonal exchanges and discourse functions, while opinion categories are focused on participants' private states usually towards objects (which may be other participants). In our corpus we found that it is not always necessary for a Sentiment instance to be associated with an Assess Dialog Act. Consider the utterance: "Okay, so when you have a lot of room inside. So you can make it **very easy to use**. **'Cause** you can write a lot of comments besides it." This sentence was labeled as an *Inform* DA as it functions to inform the participants of the roomy interior of the remote control. Orthogonally, it was tagged as a positive *Sentiment* ("very easy to use") and positive *Arguing* ("'Cause").

6 Experiments and Results

In this section, we perform machine learning experiments to test our hypothesis that our knowledge sources from Section 5 are useful. We perform supervised machine learning on our annotated corpus of 4302 segments (6504 sentences) using a standard SVM package (Joachims, 1999). The recognition of each opinion category is formulated as a binary classification problem. We do not attempt automatic classification for Utterance Arguing as we consider our inter-annotator agreement for this category to be too low to form a reliable gold standard.

We use two baselines: a majority-class dumb baseline that guesses false every time, and a smart SVM classifier trained on a bag of words (BOW). Then we add our opinion features individually or in combination to the baseline classifier. The lexicon features for the BOW+lex classifier are counts of words from each lexicon type in the given segment or sentence.

The AMI DA types introduced in Section 2 form the additional features for the BOW+DA classifier. The AP links described in Section 2 along with their source DA and target DA form a DA-AP-DA chain. These DA-AP-DA chains form the features for the BOW+AP classifier. Since we do not make a polarity distinction, we conflate *Positive* and *Negative Assessment* into a single category *Assessment*. As there are 15 DAs and 4 APs (after conflation) there are $15 * 4 * 15 = 900$ possible combinations; however of these, only 99 types actually occur in our annotated corpus. The BOW+DA+AP classifier has all the DA features and the AP features; the BOW+All classifier uses DA, AP and lex features.

	Acc	Prec	Rec	F-measure
	Segment Level classification			
BOW	88.42	69.52	51.95	57.99
BOW+lex	88.84	70.1	**53.07**	**59.16**
BOW+DA	89.28	**73.81**	**54.62**	**61.26**
BOW+AP	88.73	70.1	53.07	59.16
BOW+DA+AP	89.24	**73.14**	54.38	**60.9**
BOW+All	89.28	**73.17**	**54.98**	**61.37**
	Sentence Level classification			
BOW	89.43	69.22	46.69	54.62
BOW+lex	89.51	69.12	48.04	55.53
BOW+DA	89.80	71.11	49.07	**56.7**
BOW+AP	89.40	69.42	46.21	54.11
BOW+DA+AP	89.79	71.29	48.87	56.54
BOW+All	90.3	**73.22**	**51.32**	**59.20**

Table 3: Arguing Classification Results.

The accuracy of the majority-class Arguing classifier is 82.84% at the segment and 85.5% at the sentence level. All the classifiers, including the smart baseline (BOW), improve over this by about 7 percentage points at the segment and by about 4 percentage points at the sentence level. Table 3 shows the performance of our Arguing classifiers. All results are reported over 20-fold cross-validation. The results that are significantly better ($p < 0.05$) than the smart BOW baseline are shown in bold. The results in Table 3 indicate that the DA features are useful for detection of Arguing. The only classifier that performs significantly better than the smart baseline at both the segment and sentence level is the one that uses all the features (BOW+all). This corroborates our hypothesis that lexical and discourse information are complementary. The Arguing lexicon significantly improves recall and f-measure for segments, but the results are not significant at the sentence level. We think this is because our preliminary lexicon with its lesser coverage can still succeed in finding matches in the larger segmental units, but fails in the smaller sentential units. We believe increasing the breadth of coverage will remedy this. Table 4 shows the performance of the Sentiment

	Acc	Prec	Rec	F-measure
	Segment Level classification			
BOW	86.87	80.84	48.53	58.77
BOW+lex	88.29	81.43	**56.14**	**65.18**
BOW+DA	87.45	81.93	**51.48**	**62.0**
BOW+AP	87.27	81.02	**50.69**	**61.11**
BOW+DA+AP	87.36	82.73	49.55	**60.93**
BOW+All	88.66	82.01	**57.89**	**66.88**
	Sentence Level classification			
BOW	88.23	82.41	44.08	56.61
BOW+lex	89.77	81.99	**54.70**	**64.89**
BOW+DA	88.59	82.11	**47.08**	**59.14**
BOW+AP	88.67	82.68	**46.73**	**58.97**
BOW+DA+AP	88.64	82.47	**47.1**	**59.22**
BOW+All	89.95	82.49	**55.42**	**65.62**

Table 4: Sentiment Classification Results

classifiers. Here too, using all features gives the best performance at both the segment and sentence level. Additionally, we also see that each of our features, lexical or dialog-based, individually improve the recall and f-measure. The accuracy of the majority-class Sentiment classifier is 79.12% at the segment and 82.16% at the sentence level. The best classifier (BOW+All) improves over this by about 9 percentage points at the segment and 8 percentage points at the sentence level. We also see that the lexicons from the monologue text genres help in improving the recall significantly. It is encouraging that resources developed for extracting sentiments from monologic texts will be useful for processing conversational data as well.

7 Related Work

Sentiment detection is being carried out across a variety of genres and at various levels (e.g. document level by Thomas et al. (2006), phrase level by Wilson et al. (2005)).

Like much other work on subjectivity (e.g. Nasukawa and Yi (2003)), we use lexicons as knowledge sources in classification. Somasundaran et al. (2007) use a lexicon for detecting Arguing in text. In contrast, our work is on multi-speaker conversations. Biber (1988) in work on textual variation identifies a dimension of "Overt persuasion" whose categories (e.g. modal verbs and conditionals) are similar to the expressions we gathered in our lexicon. Ducrot (1973) studies arguing related items, but his work is on French and is not corpus-based. A vast body of work exists within linguistics, rhetoric and philosophy that is relevant to arguing (e.g.(Dancygier, 2006; van Eemeren and Grootendorst, 2004)).

With regard to meetings, the most closely related work includes the dialog-related annotation schemes for various available corpora of conversation (Dhillon et al. (2003) for ICSI MRDA; Carletta et al. (2005) for AMI; Burger et al. (2002) for ISL). We think our annotation scheme complements the annotations provided in these corpora in that it adds finer granularity for statement-speech acts by distinguishing expressions of sentiment and arguing from objective statements.

Our work also connects to research on hot spots (Wrede and Shriberg, 2003), and efforts to annotate the mental states of participants in meetings or interviews on the basis of multi-modal data (Devillers et al., 2005; Reidsma et al., 2006). The focus of these kinds of research is different from ours in that they target the actual mental states of the speakers in the unfolding situation, while we focus on subjective states communicated through language. While often the same, they are not necessarily identical as language allows for displacement: participants may calmly report about other people's anger, report their past or expected future mental states, etc. Our approach is similar to the one used by Galley et al. (2004) where adjacency pair information is used to detect agreement/disagreement amongst participants. Similarly, in the prediction of congressional vote, Tomas et al. (2006) use adjacency pair information to detect agreement amongst speakers. Another closely related area is argument diagramming of meetings (Rienks et al., 2005), where lines of deliberation are analyzed without making a subjective/objective distinction. Our work can also be combined with ongoing work on decision detection (Hsueh and Moore, 2007; Purver et al., 2006). While our annotations track opinions in the decision making process, the decision detection research is mostly concerned with its outcome.

8 Conclusion and Future Work

We presented the annotation of the Opinion types Sentiment and Arguing on meetings. We developed a new lexical resource for the Arguing category. We showed that previously developed Sentiment lexi-

cons have good coverage in the new genre. We hypothesized that dialog structure interacts with the expression of opinions and confirmed this through machine learning experiments. Finally, using all the features gave the best performance, confirming our hypothesis that both lexical and discourse information is needed to detect opinions in multi-party conversations.

Our future work will involve increasing the breadth and reliability of our arguing lexicon both manually and via automatic means. We also plan to use richer discourse and meeting level information as well as study interactions between opinion types.

References

D. Biber. 1988. *Variation Across Speech and Writing*. Cambridge University Press.

S. Burger, V. MacLaren, and H. Yu. 2002. The ISL Meeting Corpus: The Impact of Meeting Type on Speech Style. In *ICSLP 2002*.

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. 2005. The AMI Meetings Corpus. In *Proceedings of the Measuring Behavior Symposium on "Annotating and measuring Meeting Behavior"*.

B. Dancygier. 2006. *Conditionals and Prediction: Time, Knowledge and Causation in Conditional Constructions*. Cambridge University Press.

L. Devillers, S. Abrilian, and J.-C. Martin. 2005. Representing real-life emotions in audiovisual data with non basic emotional patterns and context features. In *Proc. of ACII*.

R. Dhillon, S. Bhagat, H. Carvey, and E. Shriberg. 2003. Meeting recorder project: Dialog act labeling guide. Technical report, ICSI Tech Report TR-04-002.

O. Ducrot. 1973. *Le preuve et le dire*. Mame.

M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. 2004. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. In *ACL*.

P.-Y. Hsueh and J. Moore. 2007. What decisions have you made: Automatic decision detection in conversational speech. In *NAACL/HLT 2007*.

T. Joachims. 1999. Making large-scale SVM learning practical. In B. Scholkopf, C. Burgess, and A. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*. MIT-Press.

R. Landis and G. Koch. 1977. The measurement of observer agreement for categorical data. In *Biometrics, Vol. 33, No. 1*.

T. Nasukawa and J. Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *K-CAP 2003*.

M. Purver, P. Ehlen, and J. Niekrasz. 2006. Detecting action items in multi-party meetings: Annotation and initial experiments. In *Machine Learning for Multimodal Interaction*. Springer-Verlag.

D. Reidsma, D. Heylen, and R. Ordelman. 2006. Annotating emotions in meetings. In *LREC 2006*.

R. Rienks, D. Heylen, and E. van der Weijden. 2005. Argument diagramming of meeting conversations. In Vinciarelli A. and Odobez J., editors, *Multimodal Multiparty Meeting Processing, Workshop at the 7th Intl. Conference on Multimodal Interfaces*.

S. Somasundaran, T. Wilson, J. Wiebe, and V. Stoyanov. 2007. QA with attitude: Exploiting opinion type analysis for improving question answering in on-line discussions and the news. In *Intl. Conference on Weblogs and Social Media*.

P. J. Stone, D. Dunphy, M. S. Smith, and D.M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

M. Thomas, B. Pang, and L. Lee. 2006. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of EMNLP*, pages 327–335.

F. H. van Eemeren and R. Grootendorst. 2004. *A systematic theory of argumentation*. Cambridge University Press.

J. Wiebe. 2002. Instructions for annotating opinions in newspaper articles. Dept. of Computer Science Technical Report TR-02-101, University of Pittsburgh.

T. Wilson and J. Wiebe. 2005. Annotating attributions and private states. In *Proc. of the ACL Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*.

T. Wilson, J. Wiebe, and P Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP 2005*.

B. Wrede and E. Shriberg. 2003. Spotting hotspots in meetings: Human judgments and prosodic cues. In *Eurospeech*.

A Appendix A. Arguing Opinions and Discourse Flow

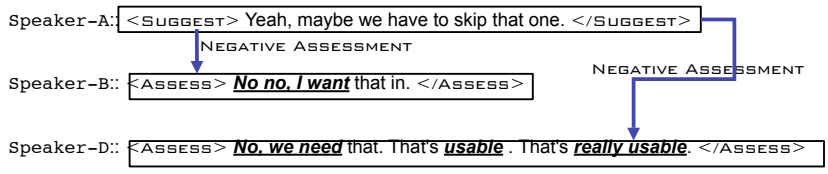


Figure 2: Arguing expression and discourse flow

As with the Sentiment opinions, for the Arguing category, too, we found an interrelation with Dialog Act exchanges. Consider the AMI meeting snippet below where the participants are discussing a beeping functionality. Speaker A has just suggested skipping it.

> *Speaker-A::* Yeah, maybe we have to skip that one.
> *Speaker-B::* No no, I want that in.
> *Speaker-D::* No, we need that. That's usable . That's really usable.

Figure 2 illustrates the annotations on this snippet. *A Suggests* that they might skip the beeping functionality. *B Argues* against this *Suggestion* with a vehement "No no". The "I want" in *B's* utterance acts as both *Sentiment* (positive towards the thing wanted) as well as *Arguing*. Thus, there is a *Negative Assessment* link between the two. *D*, too, *Argues* against *A's Suggestion* by stating that the beeping functionality is a necessity. He justifies this stance by evaluating the remote as usable and then reinforces his argument though repetition and intensification.

A Model of Compliance and Emotion for Potentially Adversarial Dialogue Agents

Antonio Roque and **David Traum**
USC Institute for Creative Technologies
13274 Fiji Way
Marina del Rey, CA 90292
{roque,traum}@ict.usc.edu

Abstract

We present a model of compliance, for domains in which a dialogue agent may become adversarial. This model includes a set of emotions and a set of levels of compliance, and strategies for changing these.

1 Overview

We present an information-state based model of compliance for an agent who is questioned. The agent tracks several emotional and interpersonal variables, which can be updated depending on the dialogue act, content, and other features of utterances. A compliance level is computed based on the values of these variables. This work is in the tradition of research in building affective dialogue systems (André et al., 2004a) embodied as virtual humans (Rickel et al., 2002), with emotional components for training or tutoring purposes (Gratch and Marsella, 2005).

A model of emotion in an affective dialogue system may, among other things, influence that system's cognitive behavior (Becker et al., 2004), model the effects of social language (Cassell and Bickmore, 2003), or control behavior such as its level of politeness (André et al., 2004b). Our study is closer in spirit to (Traum et al., 2005), in which a virtual human decides on a negotiation strategy based on its emotional appraisal of the topic, of its negotiation options, and of the human speaker. Our study also overlaps somewhat in topic with (de Rosis et al., 2003), in which a computer decides whether or not to deceive.

In this work we build a model of compliance - how helpful the agent will be - in a domain in which the agent may become reticent or adversarial, along with the emotional components that direct that agent's decision.

2 Testbed Domain

Our testbed application is in the domain of *Tactical Questioning*, in which small-unit military personnel hold conversations with individuals to produce information of military value (Army, 2006). We are specifically interested in this domain when applied to civilians, when the process becomes more conversational and additional goals involve building rapport with the population and gathering general information about the area of operations.

We have developed an application for training individuals in conducting Tactical Questioning sessions with civilians. The scenario takes place in contemporary Iraq, where the trainee must talk to Hassan, a local government functionary. If the trainee convinces Hassan to help him, the trainee will confirm suspicions about an illegal tax being levied on a new marketplace; if exceptionally successful, the trainee may even discover that the tax has been placed by Hassan's employer. But if Hassan becomes adversarial, he may lie or become insulting.

Figure 1 shows the beginning of a typical dialogue with Hassan. Rather than working to determine what the human user wants and then providing it, in turns 6 and 8 Hassan provides replies that are off-topic or of low information value. The trainee's goal is to increase the value of Hassan's responses by appealing to Hassan's emotions and making him more compli-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 35–38,
Antwerp, September 2007.

ant. Section 4 describes how this can happen.

1	Trainee	Hello Hassan
2	Hassan	Hello
3	Trainee	How are you doing?
4	Hassan	Well, under the circumstances we are fine
5	Trainee	I'd like to talk about the marketplace
6	Hassan	I hope you do not expect me to tell you anything
7	Trainee	I just want to know why people aren't using the marketplace
8	Hassan	I don't feel like answering that question

Figure 1: Scenario Dialogue

3 System Implementation

As a training application, Hassan incorporates "human-in-the-loop" interactivity, and logs utterances, language features, and emotional states at every turn, with the aim of producing a summary for *after-action review*, at which time a human trainer and trainee may discuss the session. For this reason, Hassan may react realistically to a trainee's bribes or threats of force, even though such actions are against policy for Tactical Questioning of noncombatants (Army, 2006): these behaviors would be reviewed by a human trainer during or after the training session.

The natural language components of our dialogue agent include a set of statistical classifiers working together with a rule-based dialogue manager. The Automated Speech Recognition output is sent to the classifiers, three of which detect language features, and three of which suggest possible replies. The Dialogue Manager uses its model of emotions and compliance to determine which of the suggested replies, if any, are to be made back to the user, as described in the next section. Further system implementation details are given in (Traum et al., 2007).

4 Model of Dialogue, Emotions, and Compliance

In our training scenario, trainees have a specific set of information that they want to learn from Hassan. In the general Tactical Questioning domain, a questioner seeks **compliance**: that the interviewee at least answers any questions truthfully, and ideally that the interviewee takes the initiative in offering information. Note that this is different from *cooperation* as in (Allwood, 2001), as it does not make any assumptions about cognitive consideration, joint purpose, ethical consideration, or trust; compliant behavior might or might not be cooperative. The components of our model were developed based on a study of Tactical Questioning domain documents such as (Army, 2006) and (Paul, 2006).

More details about our model of compliance are given in section 4.3. The following sections describe how the human speaker's utterances indirectly update the agent's level of compliance by means of a model of emotion.

4.1 Dialogue Features

A human trainee's utterance is analyzed by statistical classifiers to detect its principal dialogue move, topic, and degree of politeness.

We define several dialogue moves relevant to the domain of tactical questioning. *Opening* moves are general greetings and introductions. *Complimentary* moves are those in which the trainee compliments or flatters the person being questioned. *General Conversation* includes talk meant to build a sense of social bonding between the agent and the trainee, as well as expressions of goodwill and off-topic statements. *Task Conversation* is talk related to information the trainee is interested in: in the case of this scenario, questions about the marketplace and taxation, about the agent and his business, and so on. *Threatening* moves are those that include a threat against the agent, and *Offering* moves offer to provide something. Finally, *Closing* dialogue moves are those that end the conversation.

The topic of the utterance will be a topic from one of three sets, or 'other'. The Information Request topics allow the agent to identify what the trainee is referring to in Task Conversation dialogue moves: the marketplace, taxation, and so on. The set of Threat-related and the set of Offer-related topics refer to the kinds of threats and offers that a trainee may make in the course of a conversation.

Finally, the third language feature to analyzed is the utterance's level of politeness. This will be identified as either polite, impolite, or neutral.

4.2 Emotional and Social variables

We identify four emotional and social variables (emotions, for short) applicable to the domain. They have been named to be intuitive to a trainer over-

seeing a session. *Respects Trainee* represents the degree of trust and respect the agent feels for the trainee. *Feels Respected* represents the extent to which the agent feels honored and respected. *Social Bonding* represents how much of a social relationship the agent feels for the trainee, and *Fear* represents how afraid the agent feels.

These emotions are represented as integer value components in an Information State dialogue manager (Traum and Larsson, 2003). They are updated by rules based on the state of the information state components and the language features identified in the trainee's utterance. For example, a Complimentary dialogue move would increase the agent's Feels Respected and Social Bonding values and decrease its Fear. A Threatening dialogue move would increase the agent's level of Fear but decrease its Feels Respected and Social Bonding values. A General Conversation dialogue move that was Polite would increase the Social Bonding value.

4.3 Compliance

For this study, we focused on the effect of compliance on the agent's verbal responses in terms of how much information the agent provides in response to the trainee's questions, whether the information is useful, to what extent the information is true, and whether the reply includes polite, neutral, or rude words.

Our model of compliance consists of three levels, which have the following effects.

At the *Compliant* level, the agent will answer the trainee's direct questions truthfully, and will try to provide useful information. The agent will be friendly and polite.

At the *Reticent* level, the agent will not provide any useful information. The agent may express that they do not wish to comply, may reply with off-topic remarks, or may make other low-information responses. The agent will generally be neither rude nor polite, but may be dismissive.

At the *Adversarial* level, the agent again will not provide any useful information, and may reply with off-topic or low-information responses. However, the agent may also be rude or insulting. Furthermore, the agent may reply deceptively: offering, in a neutral or polite way, high-information statements that are not true.

The agent's level of compliance may not be immediately apparent to the human speaker: for example, an agent replying in a neutral way with no information may be at the Reticent or Adversarial level, or it may be at the Compliant level and simply not have any useful information to provide. Similarly, answers with expected responses, such as greetings or farewells, may be answered the same at many compliance levels. Finally, if an agent is providing high-information responses, the human participant may not know if those are useful truths or plausible lies.

4.4 Compliance and Emotions

In the course of a dialogue, the agent's level of compliance may vary. After every utterance, the agent's emotions are checked to see if they change the agent's level of compliance. The goal of the trainee is to make the agent compliant by producing utterances that will update the agent's emotions in ways that will make the agent compliant. There are three basic strategies that the trainee can pursue, which are defined by the ways in which emotions affect compliance.

In the *Empathic* strategy, the trainee attempts to make the agent sympathetic to the trainee, and therefore to the trainee's goals. This is modeled by having the agent's compliance level become Compliant when the agent's Respects Trainee, Feels Respected, and Social Bonding scores all rise above a certain threshold. However, if those three emotions are below a given threshold, the agent's compliance level becomes Adversarial.

In the *Offering* strategy, the agent becomes compliant after the trainee makes an Offering dialogue move whose Topic is from the set of Offers that the agent is defined as being receptive to.

In the *Threatening* strategy, the trainee uses a Threat dialogue move to raise the agent's Fear above a certain threshold. If the trainee then makes a Threat that the agent is vulnerable to, the agent will become Compliant.

5 Future Directions

An evaluation of the entire system is described in (Traum et al., 2007). We hope to perform an evaluation of the compliance and emotion components

separately. One possibility is to do a semi-Wizard of Oz evaluation in which the ASR and language analysis tasks are performed by a human, to factor out errors in those components. Another possibility is to compare the system's performance in updating its information state with the performance of human coders in updating the information state, as was done in (Roque et al., 2006). Alternately, we could focus on how plausible the model of emotions and compliance is in terms of human processes by comparing it to data from human surveys, as was done in (Mao and Gratch, 2006).

The model of emotion and compliance that we have presented is motivated by the domain of Tactical Questioning, and the features and policies that we have implemented have been guided by that domain. As we continue to develop Hassan and other Tactical Questioning agents, we plan to add capabilities that will allow us to build more general and sophisticated models of emotion and compliance.

6 Acknowledgments

This work has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. We would like to thank the other members of the TACQ team at ICT for work on the system in which these ideas are implemented and discussions regarding the compliance model, especially Anton Leuski, Susan Robinson, and Bilyana Martinovski. We would also like to thank the anonymous reviewers for their comments.

References

Jens Allwood. 2001. Cooperation and flexibility in multimodal communication. In Harry Bunt and Robbert-Jan Beun, editors, *Cooperative Multimodal Communication*, volume 2155 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin/Heidelbert.

Elisabeth André, Laila Dybkjær, Wolfgang Minker, and Paul Heisterkamp, editors. 2004a. *Affective Dialogue Systems, Tutorial and Research Workshop, ADS 2004, Kloster Irsee, Germany, June 14-16, 2004, Proceedings*, volume 3068 of *Lecture Notes in Computer Science*. Springer.

Elisabeth André, Matthias Rehm, Wolfgang Minker, and Dirk Bühler. 2004b. Endowing spoken language dialogue systems with emotional intelligence. In *Affective Dialogue Systems, Tutorial and Research Workshop, ADS 2004, Kloster Irsee, Germany, June 14-16, 2004, Proceedings*, pages 178–187.

Department of the Army. 2006. Police intelligence operations. Technical Report FM 3-19.50. Appendix D: Tactical Questioning.

Christian Becker, Stefan Kopp, and Ipke Wachsmuth. 2004. Simulating the emotion dynamics of a multimodal conversational agent. In *Affective Dialogue Systems, Tutorial and Research Workshop, ADS 2004, Kloster Irsee, Germany, June 14-16, 2004, Proceedings*, pages 154–165.

Justine Cassell and Timothy Bickmore. 2003. Negotiated collusion: Modeling social language and its relationship effects in intelligent agents. *User Modeling and User-Adapted Interaction*, 13:89–132.

Fiorella de Rosis, Cristiano Castelfranchi, Valeria Carofiglio, and Giuseppe Grassano. 2003. Can computers deliberately deceive? a simulation tool and its application to turings imitation game. *Computational Intelligence*, 19(3).

Jonathan Gratch and Stacy Marsella. 2005. Some lessons for emotion psychology for the design of lifelike characters. *Journal of Applied Artificial Intelligence*, 19(3-4):215–233. Special issue on Educational Agents - Beyond Virtual Tutors.

Wenji Mao and Jonathan Gratch. 2006. Evaluating a computational mode of social causality and responsibility. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Japan.

Matthew C. Paul. 2006. Tactical questioning: human intelligence key to counterinsurgency campaigns. *Infantry Magazine*, Jan-Feb.

Jeff Rickel, Stacy Marsella, Jonathan Gratch, Randall Hill, David Traum, and Bill Swartout. 2002. Towards a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, pages 32–38, July/August.

Antonio Roque, Hua Ai, and David Traum. 2006. Evaluation of an information state-based dialogue manager. In *Brandial 2006: The 10th Workshop on the Semantics and Pragmatics of Dialogue*, University of Potsdam, Germany, September 11-13.

David Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In R. Smith and J. van Kuppevelt, editors, *Current and New Directions in Discourse and Dialogue*, pages 325–353. Kluwer, Dordrecht.

David Traum, William Swartout, Stacy Marsella, and Jonathan Gratch. 2005. Fight, flight, or negotiate: Believable strategies for conversing under crisis. In *5th International Conference on Interactive Virtual Agents*. Kos, Greece.

David Traum, Antonio Roque, Anton Leuski, Panayiotis Georgiou, Jillian Gerten, Bilyana Martinovski, Shrikanth Narayanan, Susan Robinson, and Ashish Vaswani. 2007. Hassan: A virtual human for tactical questioning. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*.

Acquiring and Evaluating a Dialog Corpus through a Dialog Simulation Technique

David Griol, Lluís F. Hurtado, Emilio Sanchis, Encarna Segarra
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, E-46022 València, Spain
{dgriol,lhurtado,esanchis,esegarra}@dsic.upv.es

Abstract

In this paper, we present an approach for automatically acquiring a dialog corpus by means of the interaction of a dialog manager and a user simulator. A random selection of the answers has been used for the operation of both modules, defining stop conditions for automatically deciding if the dialog is successful or not. Therefore, an initial corpus is not necessary to develop these two modules. In this work, we use a statistical dialog manager to evaluate the behavior of the corpus acquired using this approach. This dialog manager has been learned from the simulated corpus and has been evaluated using a previous corpus acquired for the task with real users.

1 Introduction

Learning statistical approaches to model the different modules that compose a dialog system has reached a growing interest during the last decade (Young, 2002). Although, in the literature, there are models for dialog managers that are manually designed, over the last few years, approaches using statistical models to represent the behavior of the dialog manager have also been developed (Williams and Young, 2007), (Lemon et al., 2006), (Torres et al., 2003).

In this field, we have recently developed an approach to manage the dialog using a statistical model that is learned from a data corpus. This work has been applied within the domain of a Spanish project call DIHANA (Benedí et al., 2006). The task that we considered is the telephone access to information about train timetables and prices in Spanish. A set of 900 dialogs was acquired in the DIHANA project using the Wizard of Oz technique. A set of 300 different scenarios was used to carry out the acquisition. Two main types of scenarios were defined. Type S1 defined only one objective for the dialog. Type S2 defined two objectives for the dialog. This corpus was labeled in terms of dialog acts to train the dialog model. The results of this work can be found in (Hurtado et al., 2006).

The success of statistical approaches depends on the quality of the data used to develop the dialog model. A great effort is necessary to acquire and label a corpus with the data necessary to train a good model. One solution for this problem consists of the development of a module that simulates the user answers. 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 present an approach to acquire a labeled dialog corpus from the interaction of a user simulator and a dialog manager. In this approach, a random selection of the system and user answers is used. The only parameters that are needed for the acquisition are the definition of the semantics of the task (that is, the set of possible user and system answers), and a set of conditions to automatically discard unsuccessful dialogs. We have acquired a corpus for the DIHANA task using this approach. This corpus has been used for training our statistical dialog manager. Then, the Wizard of Oz corpus of the DIHANA project has been used to evaluate the be-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 39–42,
Antwerp, September 2007.

havior of this dialog manager with real users.

2 Our approach for automatically acquiring a dialog corpus

As stated in the introduction, our approach for acquiring a dialog corpus is based on the interaction of a user simulator and a dialog manager. Both modules use a random selection of one of the possible answers defined for the semantic of the task (user and system dialog acts).

The user simulation simulates the user intention level, that is, the simulator provides concepts and attributes that represent the intention of the user utterance. Therefore, the user simulator carries out the functions of the ASR and NLU modules. The semantics selected for the dialog manager is represented through the 51 possible system answers defined for the task. The selection of the possible user answers is carried out using the semantics defined for the user in the NLU module.

An error simulator module has been designed to perform error generation and the addition of confidence measures in accordance with an analysis of the DIHANA corpus. This information modifies the frames generated by the user simulator and also incorporates confidence measures for the different concepts and attributes. Experimentally, we have detected 2.7 errors per dialog. This value can be modified to adapt the error simulator module to the operation of any ASR and NLU modules.

2.1 The corpus acquisition

A maximum number of system turns per dialog was defined for acquiring a corpus. 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 system 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 contained in the DR, that is, not provided by the 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.

A total of two million simulated dialogs were necessary in order to acquire a dialog corpus following the same characteristics of the DIHANA corpus and using the same set of scenarios developed for its acquisition. The computer used for the acquisition was able to simulate 1,000 dialogs every 40 minutes.

Table 1 summarizes the statistics of the acquisition for the two types of scenarios that were defined.

	Type S1	**Type S2**
Simulated dialogs	10^6	10^6
Successful dialogs	15,383	1,010
Different dialogs	14,921	998
Number of user turns per dialog	4.9	6.2

Table 1: Statistics of the new corpus acquisition

Table 2 shows an example of a dialog corresponding to a Type S1 scenario (to know timetables from Valencia to Barcelona for May the 30th).

S1: (Opening:Nil:Nil)
Welcome to the railway information system. How can I help you?
U1: Departure-Hour [0.9]
Destination: Barcelona [0.9]
Departure-Date: 05-31-2007 [0.1]
Arrival-Hour: 08.00-23.00 [0.9]
S2: (Confirmation:Date:Date)
Do you want to leave on May the 31th?
U2: Negation[0.9]
Departure-Date: 05-30-2007 [0.9]
S3: (Answer:Departure-Hour:Departure-Hour)
(New-Query:Nil:Nil)
{*Timetables information*} Anything else?
U3: Negation
S4: (Closing:Nil:Nil)
Thank you for using the system.

Table 2: An example of a Type S1 dialog extracted from the acquired corpus

The dialog begins with a greeting turn generated by the system (S1). In the first turn, the simulated

user provides the *Destination* and the *Departure-Date*. In addition, it facilitates the *Arrival-Hour* (set as optional data for the scenario). The error simulator introduces in this first turn an error value in the *Departure-Date* slot (it changes day 30 by 31) and assigns confidence scores to the different slots. In this case, a low confidence is assigned to this erroneous value.

In the second system turn, a confirmation for the *Departure-Date* is selected. Considering the information defined in the objective of the scenario, the user simulator selects a *Negation* dialog act and provides the correct value for the *Departure-Date* according to the objective (U2). In this turn, the error simulator assigns a high confidence value to the information provided by the user. In the following system turn (S3), the dialog manager selects to make a query about timetables to the database. As the necessary information is available, the database query module carries out the query and the dialog manager provides the information defined as objective for the dialog. Having this information, the user simulator selects a request for closing the dialog in the following turn (U3).

3 Dialog management in the DIHANA project

We have developed a Dialog Manager (DM) based on the statistical modelization of the sequences of dialog acts (user and system dialog acts). A detailed explanation of the dialog model can be found in (Hurtado et al., 2006). We represent a dialog as a sequence of pairs (*system-turn*, *user-turn*):

$$(A_1, U_1), \cdots, (A_i, U_i), \cdots, (A_n, U_n)$$

where A_1 is the greeting turn of the system, 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.

The objective of the dialog manager at time i is to generate the best system answer. This selection, that is a local process, takes into account the previous history of the dialog, that is to say, the sequence of states of the dialog preceding time i:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | S_1, \cdots, S_{i-1})$$

where set $\mathcal{A}$ contains all the possible system answers.

As the number of all possible sequences of states is very large, we defined a data structure in order to establish a partition in the space of sequences of states. This data structure, that we call Dialog Register (DR), contains the concepts and attributes provided by the user throughout the previous history of the dialog. Using the DR, the selection of the best system answer is made using this maximization:

$$\hat{A}_i = \underset{A_i \in \mathcal{A}}{\operatorname{argmax}} P(A_i | DR_{i-1}, S_{i-1})$$

The last state (S_{i-1}) is considered for the selection of the system answer due to the fact that a user turn can provide kinds of information that are not contained in the DR, but are important to decide the next system answer. This is the case of the task-independent information.

The selection of the system answer is carried out by means of a classification process, in which a multilayer perceptron (MLP) is used. The input layer holds the codification of the pair (DR_{i-1}, S_{i-1}) and the output of the MLP can be seen as the probability of selecting each one of the 51 different system answers defined for the DIHANA task. For the DIHANA task, the DR is a sequence of 15 fields, where each concept or attribute has a field associated to it.

4 Evaluation

A statistical dialog manager was learned using the corpus acquired with the dialog simulator technique (M1 manager). The DIHANA corpus was used as test set to evaluate the behavior of this dialog manager with a real users corpus.

We also learned another dialog manager using the DIHANA corpus as training set (M2 manager). A 5-fold cross-validation process was used to carry out the evaluation of this manager. Therefore, all the DIHANA corpus is used for testing both M1 and M2 dialog managers.

We defined three measures to evaluate the performance of both dialog managers:

1. The percentage of answers that follows the strategy defined for the acquisition of the DIHANA corpus ($\%strategy$).

2. The percentage of answers that are coherent with the current state of the dialog, but that not necessary follow this strategy ($\%correct$).

3. The percentage of answers that are considered erroneous according to the current state of the dialog ($\%error$).

Table 3 shows the results obtained for the different measures after the evaluation.

	M1 manager	M2 manager
$\%strategy$	54.57%	97.34%
$\%correct$	88.83%	99.33%
$\%error$	11.17%	0.67%

Table 3: DM evaluation results

It can be observed that the M1 manager provides a 88.83% of answers that are coherent with the current state of the dialog. Using the DIHANA corpus in order to learn the dialog model (M2 manager), the 97.34% of the answers provided by this dialog manager follows the strategy defined for the WOz. With regard to the M1 manager, only the 54.57% follows this strategy. Therefore, we can see that the M1 dialog manager separates from the strategy defined for the WOz as expected. Regarding to the $\%error$ measure, the M1 dialog manager provides a 11.17% percentage of answers that are not compatible with the state of the dialog.

5 Conclusions

In this paper, we have presented an approach to automatically acquire a dialog corpus by means of the interaction of a user simulator and a dialog manager. For the development of both modules, we defined the semantics of the possible answers for the system and the user in a specific task. A random selection of these answers and a set of stop conditions were used in order to acquire a dialog corpus, deciding automatically if the dialog has to be considered successful.

The corpus that has been obtained by means of this approach has been used to learn a dialog manager, using a statistical dialog model. We have used a previous corpus acquired with real users to evaluate this dialog manager. The results of the evaluation show that the learned dialog model could be used as an initial dialog manager, generated without many effort and with very high performance. This initial dialog manager could be improved with a posteriori interaction with real users.

As future work, we want to use this approach to acquire a dialog corpus within the framework of a new project called EDECAN. The main objective of the ongoing EDECAN project is to develop a dialog system for booking sports facilities in our university. Using this approach, we want to acquire a corpus that makes possible the learning of a dialog manager for the domain of the EDECAN project. This dialog manager will be used in a supervised acquisition of a dialog corpus with real users.

6 Acknowledgements

Work partially supported by the Spanish MEC and FEDER under contract TIN2005-08660-C04-02.

References

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 *Proc. of LREC'06*, Genove, Italy.

L.F. Hurtado, D. Griol, E. Segarra, and E. Sanchis. 2006. A Stochastic Approach for Dialog Management based on Neural Networks. In *Procs. of InterSpeech'06*, Pittsburgh, USA.

O. Lemon, K. Georgila, and J. Henderson. 2006. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the talk towninfo evaluation. In *Proc. of IEEE-ACL Workshop on Spoken Language Technology (SLT 2006)*, Aruba.

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.

F. Torres, E. Sanchis, and E. Segarra. 2003. Development of a stochastic dialog manager driven by semantics. In *Proc. of EuroSpeech'03*, pages 605–608.

J. Williams and S. Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. In *Computer Speech and Language 21(2)*, pages 393–422.

S. Young. 2002. The Statistical Approach to the Design of Spoken Dialogue Systems. Technical report, Cambridge University Engineering Department.

An Empirical View on IQA Follow-up Questions

Manuel Kirschner and Raffaella Bernardi
KRDB Center, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
{kirschner,bernardi}@inf.unibz.it

Abstract

In a realistic Interactive Question Answering (IQA) situation, one third of the users pose follow-up questions, i.e., go beyond a single question per dialogue. We identify two different perspectives according to which these follow-ups can be described: informational transitions and context dependency. By understanding exactly how informational transitions occur in IQA dialogues, we propose a method to guarantee that focus tree based IQA systems provide wide coverage of follow-up questions that trigger the respective set of informational transitions.

1 Introduction

This is an empirical study of follow-up questions in Interactive Question Answering (IQA) dialogues that we collected through a previous Wizard-of-Oz study. In this paper, we show that user follow-up questions are an interesting phenomenon because they occur relatively frequently in IQA dialogues, and are potentially difficult for an IQA system to understand. We will look at them from two different perspectives: (i) which informational transition can be identified between the follow-up and the dialogue context, and (ii), how some of the follow-up questions are context-dependent in that they can only be properly understood in combination with information from the dialogue context. In understanding (i), we try to find patterns and regularities in our data that enable us to predict the topics that users of an IQA system will ask about next. This knowledge will help in improving an IQA system, since we can ensure that the system will be prepared to answer the specific follow-up questions that we predicted for a specific situation in an IQA dialogue. As for (ii), on the other hand, we need to understand also *how* users typically pose follow-up questions: as we show in this paper, many follow-up questions are context-dependent, and need to be combined with information from the previous dialogue in order to be understandable for the IQA system. After analyzing follow-up questions from these two perspectives, we propose a new way of processing a certain class of follow-ups in an actual IQA system on the library domain.

2 Statistics of Follow-ups in IQA Dialogues

We conducted a Wizard-of-Oz experiment where the participants were free to choose their question topic, and the way in which to interact with the system. In this experiment, we collected 63 user-librarian dialogues by letting spontaneous visitors of the library web-site interact with what was announced as a new IQA system, but was in reality a web-based instant-messaging-like interface (Kirschner, 2006).

From the total of 192 user utterances in our corpus (spread across 63 dialogues and 166 user turns), we identified 35 that are both follow-up initiatives (i.e., from the set of 90 questions or assertions that are not from the very first user turn in each dialogue) and that are also about a topic from the library information domain, or some task related to this domain.

While from the set of 90 follow-up initiatives the proportion of user utterances we marked as off-topic is high (56, versus the 35 domain/task-related

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 43–46,
Antwerp, September 2007.

ones), we can assume that they will not pose a major problem to the IQA system. We conjecture that in most cases these utterances can be easily ignored by the natural-language understanding module, which should robustly spot only questions and assertions about task-related topics. Moreover, the analysis shows that many users do take the opportunity that IQA dialogue offers and do ask follow-up questions. Even more, the latter actually contains some of the most important parts of the dialogues (besides the first user question in each single dialogue), and the most interesting and difficult user utterances for an IQA system to process.

3 Informational Transitions

In some of the literature, the term *thematic relatedness* is used to describe transitions between utterances. We assume this is just a matter of different terminology; for the sake of clarity, we define that those follow-up questions that trigger some informational transition at the same time define the set of thematically related follow-ups. Also, note that throughout this paper, we use the term follow-up (question) to denote *any* user question that is not the very first question in a given IQA dialogue; thus, it does not imply that the follow-up be in some specific way related to the previous dialogue.

The general goal of all the approaches to be presented in this section is to explore specific relations holding between two discourse segments or dialogue turns. This is of primary interest in the context of building an IQA application, since by understanding how the conversation topic evolves via user follow-up questions, we can improve the way the system will understand and answer these follow-ups. In our empirical approach, we want to analyze how informational transitions are used in real IQA dialogues. Thus, a preliminary goal is to find a method of reliably identifying these phenomena in our dialogue data. In what follows, we describe three previous approaches to this problem, focusing on their generalizability and practical applicability for identifying informational transitions in data. At the end of this section, we will then propose a somewhat restricted (but on the other hand more practical and concise) method of identifying (a subset of) informational transitions.

In the context of planning coherent discourse in a natural language generation system, (McCoy and Cheng, 1991) gives a comprehensive account of informational transitions (there called focus shifts). For each node type, they list certain focus shift candidates, i.e., the items that are likely to come into focus in a coherent discourse (cf. Table 1). While their list of focus shift targets for the different node types is comprehensive, this is at the same time a major problem when it comes to a practical implementation: it is not at all clear how to (algorithmically) determine the correct node types, and thus the viable candidate targets for informational transitions.

In a related approach that targets IQA dialogues rather than single-speaker discourse, (Chai and Jin, 2004) define informational transitions between subsequent user questions in IQA dialogues in terms of the question "topic". The topic is either of type `entity` or `activity` and closely resembles the object and activity node types given in Table 1. While the informational state is now described in terms of only two types of elements (entity/object and activity/action) instead of the five postulated by (McCoy and Cheng, 1991), the rich set of discourse roles that these elements can introduce would still render an automatic construction of a representation of the informational state extremely difficult.

A further description of informational transitions in IQA dialogues is given in (Bertomeu et al., 2006). Unlike the two previously mentioned approaches, this work considers also system responses as possible sources for informational transitions. In fact, the authors identify specific thematic relations that may hold between a user follow-up question and the immediately previous user question, some previous user question, the immediately previous system answer or some previous system answer. Interestingly, this approach is based entirely on questions and answers corresponding to (sets of) entities that can be retrieved from a database. Thus, informational transitions are defined here in terms of the extensions of entities that are being referred to in thematically related turns of the dialogue, and in terms of which properties of these entities are being referred to. However, the transitions also lack the generality of the previously introduced approaches, since they are only useful for analyzing similar kinds of (natural language database query) dialogues that contain

Node type	Focus shift targets
`object`	Attributes of the object, actions the object plays a prominent role in (e.g., is actor of)
`action`	Actor, object, etc., of the action – any participant (Fillmore) role; purpose (goal) of action, next action in some sequence, subactions, specializations of the action
`attribute`	objects which have the attribute, more specific attribute
`setting`	objects involved in the setting; actions which typically occur in this setting
`event`	actions which can be grouped together into the event

Table 1: Informational transition targets for different focus nodes (from (McCoy and Cheng, 1991, p. 112))

only rather constrained types of questions and answers.

We will base our work on the observations on these three works.

3.1 Coverage vs. Conciseness: Searching for a Definition of Thematic Relatedness

From (McCoy and Cheng, 1991), we adopt the general idea of introducing candidate focus shift targets that represent coherent continuations of the discourse (or in our case, dialogue). To avoid the difficulty of choosing between up to five different node types that could represent the current focus of attention, we restrict ourselves to just `action`-type nodes. This is advantageous in two ways. On the one hand, actions correspond to verbs, which are inherently connected to some argument structure defining the verb's semantic roles. By querying available lexical resources like PropBank (Palmer et al., 2005), we can retrieve the verb's arguments. The corresponding semantic roles of the verb yield possible topics of follow-up questions. Thus, we can take advantage of existing lexical resources to automatically find focus nodes that represent follow-up questions involving any of the semantic roles of the verb. On the other hand, we conjecture that actions/verbs form a suitable and robust basis for describing the (informational) meaning of utterances in IQA, since most user utterances include a predicate (or an implicit reference to some predicate in the dialogue history), and syntactic parsers can be used to extract the main verbs of sentences. Taking the main verb plus any arguments to represent the core meaning of user questions seems to be an interesting possibility for automatically detecting certain informational transitions.

Once we adopt the action-based paradigm for focus nodes, we can instantiate two of the informational transition relations proposed by (Chai and Jin, 2004). In the following, we define our own set of informational transitions, starting from the definitions in (Chai and Jin, 2004), but addressing their shortcomings mentioned previously.

First of all, we use verbs and their semantic roles, plus a focus marker, as the only elements needed for representing the informational perspective, and for defining our transition types. This allows us to replace the somewhat unclear terms from the original definitions in (Chai and Jin, 2004) with clearly defined ones: verbs and arguments, as defined in PropBank. Secondly, we parametrize the transitions with respect to their origin: last user question (U_{-1}), or last system response (S_{-1}).

We restrict ourselves to transitions where the main verb either stays the same, or the follow-up question contains a synonymous verb, or no verb at all (to account for fragmentary questions). We now define the resulting three types of informational transitions.

1. TOPIC EXTENSION (U_{-1}):
Example: U1: "Can every student use inter-library loan?" – U2: "Even high-school students?"

   1. Either no verb exists in the follow-up question, or the main verb of the follow-up question is synonymous to the main verb in the last user question.
   2. Either the roles of the verb are filled differently by the follow-up (CONSTRAINT REFINEMENT), or different roles of the verb are filled by the follow-up (PARTICIPANT SHIFT).
   3. The question focus (the expected answer type) stays the same.

2. TOPIC EXPLORATION (U_{-1}):
Example: U1: "Can every student use inter-library

loan?" – U2: "How?"

1. Either no verb exists in the follow-up question, or the main verb of the follow-up question is synonymous to the main verb in the last user question.
2. The question's focus (the expected answer type) changes.

3. TOPIC EXPLORATION (S_{-1}):
Example: U1: "Can high-school students use the library?" – S1: "Yes, if they got a library card." – U2: "So how do I get it?"

1. The main verb of the follow-up question is synonymous to SOME main verb in the system response.
2. Either the roles of the verb are filled differently by the follow-up (CONSTRAINT REFINEMENT), or different roles of the verb are filled by the follow-up (PARTICIPANT SHIFT).

4 Context-dependent User Follow-up Initiatives

Besides studying the thematic relatedness of follow-up questions with respect to previous dialogues, context-dependency yields a new perspective under which to analyze follow-ups. We call a follow-up question context-dependent if it requires any information from the dialogue context in order to be fully understandable. Although this might not generally hold for more complex types of dialogue, we found that in our corpus of IQA dialogues, every user follow-up initiative that we consider context-dependent according to the above definition actually exhibits some discourse phenomena .

In a nutshell, our study shows that (1) discourse phenomena can be resolved without global context (or dialogue history), and (2) the last system response S_{-1} was often the location of the antecedents of discourse phenomena.

5 Conclusions

We showed that in a realistic IQA situation, one third of the users pose follow-up questions, i.e., go beyond a single question per dialogue. We have then introduced two different perspectives according to which the follow-ups can be described and further categorized: informational transitions and context dependency. For the latter, we have looked at discourse phenomena, and studied how these appear in IQA dialogue data. As for informational transitions, we showed that a rather concise definition is possible if we considerably reduce the scope of the problem, thus limiting the types of informational transitions we deal with. A concise definition is required for letting an IQA system predict informational transitions automatically, given some local dialogue history. The empirical evaluation of this definition shows that it fails in predicting any larger set of specific follow-up initiatives. The problem of concisely identifying informational transitions in IQA seems to be a more complex one, as the variety of different thematic relations found in our corpus alone suggests. While in future work we will try to fine-tune our definitions to further extend the modelling of follow-up initiatives in IQA, on the practical side we have started to extend our baseline IQA system for the library information domain by implementing the three proposed definitions of informational transitions, since they provide a principled way of extending the system.

References

N. Bertomeu, H. Uszkoreit, A. Frank, H.-U. Krieger, and B. Jörg. 2006. Contextual phenomena and thematic relations in database qa dialogues: results from a wizard-of-oz experiment. In *Proc. of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pp. 1–8, New York, NY.

J. Y. Chai and R. Jin. 2004. Discourse structure for context question answering. In *Proc. of the HLT-NAACL 2004 Workshop on Pragmatics in Question Answering*, Boston, MA.

M. Kirschner. 2006. Building a multi-lingual interactive question-answering system for the library domain. In *Proc. of SemDial'06*, Potsdam, Germany.

K. F. McCoy and J. Cheng. 1991. Focus of attention: Constraining what can be said next. In C. L. Paris, W. R. Swartout, and W. C. Mann, eds, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pp. 103–124. Kluwer Academic Publishers, Norwell, MA.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

An Implemented Method for Distributed Collection and Assessment of Speech Data

Alexander Siebert, David Schlangen *and* Raquel Fernández
Department of Linguistics
University of Potsdam, Germany
{das|siebert|raquel}@ling.uni-potsdam.de

Abstract

We present an approach to decreasing the cost of collecting speech data by a) distributing experimental setups as a downloadable computer program that records data and sends it back to an experiment server and b) by 're-using' subjects for instant quality evaluation of the collected data. As an example of the kind of settings in which this approach can be used, we also shortly describe an experiment we have conducted; evaluation of the collected data showed no negative effect of the 'unsupervised' collection method.

1 Introduction

While running experiments in a distributed fashion over the Internet has become accepted practice in Psychology, this methodology has so far rarely been adopted where collection of speech data is involved.[1] In the work reported here, we wanted to make available the advantages of online experimentation that are often cited (the following list is adapted from (Birnbaum, 2001)) to speech data collection:

- Freedom from the constraints of testing people at a particular time and place;
- Automatic coding and construction of data files (no data entry by assistants);
- Opportunity to obtain large and heterogeneous samples;
- Possibility to conduct cross-cultural research without the expense of travelling;
- Reduced costs of experimental assistants.

Collecting speech data poses additional technical challenges; the usual problems with data collected in this way (reliability; self-selection of subjects; data quality) also have to be addressed. The methodology we have devised (and implemented) to tackle these questions will be described in the next section. As a concrete example of an experimental setting which profits from this approach we briefly describe in Section 3 a data collection we conducted. We close with a discussion of related work (Section 4) and planned future work (Section 5).

2 Distributed Data Collection

In this section we describe the data collection methodology and the implementation we have built. We describe both in rather abstract terms here to underline the generality of the approach; a more concrete example is to follow in the next section.

2.1 Methodology

The approach is probably best explained by running through one data collection cycle. Figure 1 illustrates the data flow through the different steps. First (Step 0), the subject signs up for the experiment, using a form presented by the (web-)server. At this point, eligibility tests can be executed to filter out subjects that do not fit criteria that experimenters might want to set (e.g., first language, handed-ness, etc.).[2] Successful applicants then get access to the

[1] See e.g. (Birnbaum, 2001) for an introduction to conducting psychology experiments over the Internet, and the discussion below in Section 5 for speech-related work.

[2] A technical factor that limits the pool of potential subjects is that broadband Internet access (for down- and uploading materials) and a headset (for recording) is required on the side of

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 47–50,
Antwerp, September 2007.

experiment software. The software at this point does not contain the actual experiment script, which is only downloaded when the subject starts the actual experimental run (Step 1). The script, which controls the stimulus items, the order in which they are presented, and also the data that is to be evaluated in Part II (see below), is created on-the-fly by the server (Step 2), according to what is needed in the current state of running the experiment.

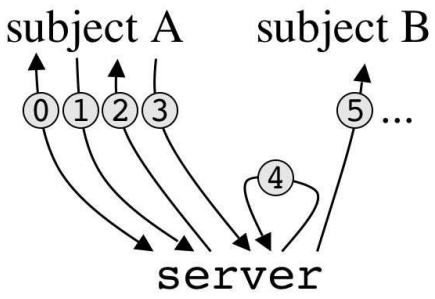


Figure 1: The Data Collection Cycle

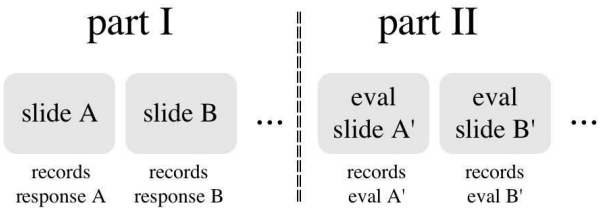


Figure 2: Schematic View of One Run

Figure 2 shows schematically one run of the experiment software for one subject. The software presents a number of "slides" to the subject and records her reactions. These "slides" can contain static information (e.g., text to read out, instructions to follow, etc.) but can also offer interactive content (e.g., puzzles to solve by manipulating items, or questionnaires); the reactions to record can range from GUI events (e.g. mouse clicks) to audio, and the responses can be timed at sub-second accuracy level. (In psychology terminology, a slide would be a single stimulus, and the recorded reaction would be the response.)

In Part II of the experiment, and this to our knowledge is an entirely novel strategy, material recorded from other subjects can be presented to the current subject, together with an evaluation questionnaire. E.g., in a simple recording experiment where the slides just contain sentences to read out, this phase II would consist of presenting to the current subject the pairs of slide and recording from a previous subject. The task then would be to evaluate the quality of the recording (or even whether the audio indeed contains a reading of the sentence!).[3]

Finishing the run brings us back to Figure 1, and Step 3, where the collected data is sent back to the experiment server. In this step audio data can optionally be compressed (lossy into MP3 format or lossless using `bz2`) to reduce the amount of data to be transferred. Step 4 then implements a consistency check. If there are criteria to do so, the data from Phase I might be pre-checked (e.g., recordings whose length deviates significantly from some preset threshold or from the mean of the data collected so far), and also the evaluation data from Phase II can be checked. The goal here is to flag all (and only) "suspicious" data, which can then be checked by the experimenter, while trying to keep as much of the data collection as possible running without further intervention.

In Step 5 finally the cycle starts again for a different subject, this time with subject A's data being available for evaluation in B's Phase II.

2.2 Implementation

On a more technical level, the data collection tool proper can be seen as a GUI shell that organises the advancement of the "slides", makes available facilities for recording data (audio, timings, GUI events, etc.), and presents data for quality assessment / evaluation. The presentation of the actual content of the slides is left to code that interfaces with this shell. (We are currently working out the best way of making this interface as general as possible; the release version will at least include an option for simple display of static content and as an example the code used in our data collection described below.)

In Phase II, the tool offers comprehensive audio controls to the user (a position slider and the usual tape-deck controls), it also allows to record all use

the user. However, in 2007 these are not unrealistic requirements.

[3] In a way we're taking our cue here from community websites that allow users to evaluate other users' contributions and hence collectively rank them.

the subject makes of these controls (see discussion of our example task below in Section 3.3).

The tool is implemented in C++ using the QT toolkit (for platform independence). It runs on Windows and Linux computers (there currently are problems with the audio library on Apple Macintosh) which must be equipped with a soundcard and headset. It weighs in at less than 5MB—a tolerable download.

3 An Example: Collecting Puzzle Moves

In this section we describe the setting for which we initially built the tool; it is at the more complex end of the spectrum of possible uses and hence nicely illustrates the potential of this strategy.

3.1 Collecting Data

The project in which this approach was developed is interested in modelling a puzzle task at both the content level, where one of the questions is how reference is made to pieces of the puzzle, and at the coordination level, where one of the questions is how different levels of interactivity shape the conversation.

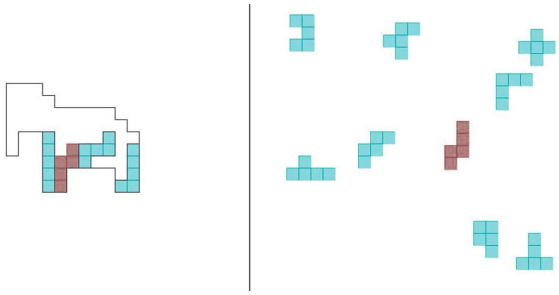


Figure 3: Example Pentomino Scene

More concretely, the task given in the data collection described here consists in describing verbally moves in a Pentomino puzzle game. Figure 3 presents one example scene; the move that is to be described here involves naming the highlighted piece on the right, describing the necessary rotation operation, and finally describing the target location in the outline on the left. This is Phase I in the terminology described above. In Phase II then scenes are presented without highlights and the recorded commands of other subjects are played, the task being to execute these commands (i.e., identify piece, rotate it, and identify target location) and then to indicate the confidence in the action performed. The audio is presented through the player tool described above and all actions (pause, repeat, skip) are recorded, as well as the judgement and the actual correctness of the execution.

Using our tool, we presented 30 scenes for execution and as many scenes for evaluation to 10 subjects (native German speakers; mostly university students). This resulted in 210 minutes of audio material, 9 sets of evaluation judgements, and a large amount of additional behavioural data (actions during evaluation).[4] The mean length of one scene description was 41 sec, with successfully followed descriptions being significantly shorter than those that couldn't be followed. Of the latter there were only 36 (12%), however, which indicates that the subjects took the recordings task seriously and produced valuable data.

As this is only a very indirect evaluation of the methodology, we also compared the audio quality of the collected recordings with that of recordings from the corpus described in (Schlangen and Fernández, 2007), which were collected with similar equipment (consumer-level headsets) but in controlled studio conditions. We used as our metric for comparison the "speech to noise ratio" as computed by the `stnr` tool from the NIST Speech Quality Assurance Package,[5] and, quite interestingly, found no significant differences between the corpora.

In the following we describe briefly two questions we addressed with these data.

3.2 Learning visual semantics

One of the goals of our project is to bridge natural language semantics, in particular for referring expressions, to perceptual features (along the lines of e.g. (Roy, 2002)). To this end, we need a large number of descriptions in our domain. The interactive material we have recorded in a different experiment (Fernández et al., 2007) provided some, but proved time-consuming to collect, annotate and segment, which is why we set out to collect more

[4] There's an obvious catch in the methodology we haven't mentioned yet: when the first subject does her run, there isn't any data available to evaluate yet. In our case, we separated for the first subject phase I (collection) and phase II (assessment).

[5] Available from `http://www.nist.gov/speech/tools/index.htm`.

in a non-interactive setting. The quality assessment data reported above convinced us that the descriptions collected in this way were not worse than those collected in the interactive setting.

Using a simple set of visual features and a simple vector-based learning and recognition model implemented as a baseline (aligning nouns with vectors of visual features; class / reference of test items determined by minimal distance) already achieved an accuracy of 62%.[6]

3.3 'Interactivity' in a non-interactive setting

In (Fernández et al., 2007) we ran the puzzle experiment in a fully interactive setting and in one with restricted interactivity (push-to-talk). The completely non-interactive material collected here gives us a good further comparison. We were especially interested in the use subjects made of the player tool to recreate some semblance of 'interactivity' through stopping, skipping and repeating audio material. The analysis of this is still going on.

4 Related Work

As mentioned in the introduction, conducting experiments over the Internet is common practice in Psychology these days (Birnbaum, 2001; Reips, 2002),[7] However, these experiments rarely involve audio. (Font Llitjos and Black, 2002; Black and Tokuda, 2005) present experiments on collecting *evaluations* of speech over the Internet; SpeechRecorder (Draxler, 2006) offers recording over the Internet much like our system, but with no provisions for recording other behavioural measures like reaction times. The combination of experiment / collection with instant user-based quality assessment that our approach offers is, to our knowledge, novel.

5 Conclusions and Future Work

We have presented an implemented methodology for distributed collection of speech data. The implemented tool is flexible in the kind of stimuli that can be presented (static and dynamic) and can record audio and other behavioural data (with sub-second accuracy). As a novel strategy for overcoming reliability problems connected to "unsupervised" data collections it allows for immediate, equally "unsupervised" quality assessment. We believe that there is a wide range of use cases in which the tool can support collection of spoken data, e.g. recording "think aloud" protocols for cognitive tasks, collecting domain utterances with simulated dialogue systems, and many more.

We are currently exploring ways of letting the software run in the user's web-browser (using Flash, or AJAX-style programming) rather than as an independent executable, but first experiments indicate that this cannot yet provide the timing accuracy and reliability that our current tool has reached.[8]

References

Michael H. Birnbaum. 2001. *Introduction to Behavioral Research on the Internet*. Prentice-Hall, NJ, USA.

Alan W. Black and Keiichi Tokuda. 2005. The blizzard challenge – 2005: Evaluating corpus-based speech synthesis on common datasets. In *Proceedings of Interspeech2005*, Lisbon, Portugal, September.

Christoph Draxler. 2006. Web-based speech data collection and annotation. In *Proceedings of 'Speech and Computer (SPECOM2006)'*, St. Petersburg, Russia, June.

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.

Ariadna Font Llitjos and Alan Black. 2002. Evaluation and collection of proper name pronunciations online. In *Proceedings of LREC2002*, Las Palmas, Canary Islands.

Ulf-Dietrich Reips. 2002. Standards for internet-based experimenting. *Experimental Psychology*, 49(4):243–256.

Deb K. Roy. 2002. Learning visually-grounded words and syntax for a scene description task. *Computer Speech and Language*, 16(3).

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, August.

[6] More detailed results will hopefully soon be reported.

[7] See also `http://psych.hanover.edu/ ... research/exponnet.html` for an up-to-date list of open experiments.

[8] **Acknowledgements:** The work reported here has partially been funded by EU (Marie Curie Programme) and DFG (Emmy Noether Programm). Thanks to the anonymous reviewers for their helpful comments.

Beyond Repair – Testing the Limits of the Conversational Repair System

David Schlangen *and* Raquel Fernández
Institute for Linguistics
University of Potsdam, Germany
{das|raquel}@ling.uni-potsdam.de

Abstract

We report on an experiment on the effects of inducing acoustic understanding problems in task-oriented dialogue. We found that despite causing real problems w.r.t. task performance, many instances of induced problems were not explicitly repaired by the dialogue participants. Almost all repairs referred to the immediately preceding utterance, with problems in prior utterances left unacknowledged. Clarification requests of certain forms were in this corpus more likely to trigger reformulations than repetitions, unlike in different settings.

1 Introduction

Clarification requests (CRs), i.e., utterances that request repair of understanding problems, are typically studied on corpora of transcribed conversations (see, *inter alia*, (Purver, 2004; Rodríguez and Schlangen, 2004)). While much knowledge about the use of this utterance type has been gathered this way, there are principled limitations to this approach:

• If there is a CR, the problem that caused it must be inferred from its form and the original speaker's reply, as it cannot be directly observed.

• As it is not obvious for the annotator whether there has been a problem or not, strategies for *avoiding* to ask for clarification cannot be studied straightforwardly.

• The effectiveness of the repair system can only indirectly be studied.

In this paper, we present the results of an experiment where we addressed these limitations through the controlled induction of understanding problems.

The remainder of the paper is structured as follows. In Section 2 we describe the method used in our experiment, the results of which are then presented in Section 3. A general discussion and conclusions close the paper.[1]

2 The Noisy Channel Experiment: Method

2.1 Overview

The experiment consisted in a voice-only cooperative task with two participants: an instruction giver (IG) had to describe in order of the numbering the placement of pieces on a puzzle (see Figure 1) to an instruction follower (IF), who only had access to the unsolved puzzle with unnumbered pieces.

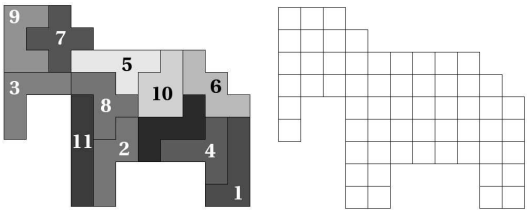


Figure 1: Solution and Outline

In half of the runs we manipulated one audio-channel by replacing (in real-time, at random points) all signal with noise, effectively blocking out the speech for the hearer. Around 10% of one speaker's

[1] The work described here is the second part of an experiment whose first part has been described in (Schlangen and Fernández, 2007). The part described here shares the general set up with that other work (i.e., introduction of noise in one channel), but uses different materials (a different task) and coding, and has a different focus for the analysis.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 51–54,
Antwerp, September 2007.

signal was removed in this way. The random, automatic placement of noise meant that we had no control over which part of the utterance exactly was masked, but we judged this preferable over more controlled manual placement of noise, which necessarily would have disabled real-time interactivity. The design is related to (Skantze, 2005), where distortion was introduced through a simulated ASR, although not in real-time.

We expected the manipulation to have an effect on the effort needed to complete the task and each of its steps (placing individual puzzle pieces). Further, and more specifically, given previously observed correlations between CR forms and problem types, we expected an increase in use of CR forms previously connected to clarifying acoustic problems. As our design tells us exactly which part of the stimulus was problematic, we also wanted to explore relations between this and whether, and if so, how clarification was requested.

2.2 Procedure

26 subjects (13 pairs) participated in the experiment. All were native English speakers (from a variety of native countries) that responded to a public call for participation. Half of them where college students while the other half had a range of different occupations. The age range was from 20 to over 40. None of the subjects reported any hearing difficulties.

The pairs of subjects were split into IG and IF and placed in different sound-proof rooms, connected by an audio-line via headsets. They were then separately briefed on the task. IG's solution was displayed on a computer screen, IF's puzzle board was implemented in a computer program. All audio was recorded; in the runs with the manipulation, both the audio before adding noise and after adding noise was recorded. IF's computer screen was video-taped.

2.2.1 Data Analysis

For analysis, the recordings were transcribed using Praat (Boersma, 2001) and annotated using MMAX (Müller and Strube, 2001); the annotators had access to both the textual transcripts and the audio material.

We segmented the recordings into *utterances* (following the guidelines in (Meteer and Taylor, 1995)) and *moves*, which we defined as all utterances belonging to the placement of one piece. We then annotated the *transition status* at move boundaries, split into `grounding state`, where a) the participants can be explicitly *confident* about their placement ("OK, I've got it. Next one!"); b) rather *unconfident* ("Well, I'll put it there. Let's see what happens."); c) they can put the current sub-task *on hold* and go back to a previous piece; d) which in turn then can be moved and placed with any of these previous grounding outcomes, or can be *reconfirmed*; and `success`, which we checked on the video recordings. Values for this feature are: *success*, *failure*, *not moved* (for moves that revisited previously placed pieces, but did not move them), and *on hold* for moves that are on hold while a previous piece is repaired.

Within the moves, we marked regions belonging together thematically, and annotated them with the following categories: a) identification of the *piece* that is to be placed; b) specifying its *orientation* and c) *location* on the grid; other common dialogue actions were d) talking about the *task setup* ("I am supposed to do these in order"); e) the *grounding status* ("well, let's see what happens"); f) noting *problems* ("This doesn't work. Something must be wrong."); g) giving a *description of the state* of the board ("To the left I have the Swiss cross, and next to it..."). Everything else was coded as h) *other*.

Finally, we identified utterances that were CRs and coded them with (Rodríguez and Schlangen, 2004)'s scheme; for reasons of space, we refer to that paper or to (Schlangen and Fernández, 2007) for a description of the values.

3 Results

3.1 Recordings

The 13 experimental runs resulted in 9 usable recordings, as two runs had to be excluded because of equipment failure and two because subjects aborted the task or didn't follow instructions.

3.2 Dialogue-based Analysis

The pairs in the noise condition finished the task in an average 1130 seconds, producing in average 653 utterances; the pairs in the control group needed 618 seconds and 422 utterances. These differences are statistically significant (Welch's t-test; t=2.7, df=4.7,

	success	failure	not_moved	on_hold
noise	57.14%	17.86%	10.71%	14.29%
no-noise	89.19%	5.40%	2.70%	2.70%

	confid	unconf	on_hold	reconf
noise	61.90%	9.52%	21.43%	7.14%
no-noise	94.60%	0%	5.40%	0%

Table 1: Success of Moves, in Percent of all Moves (top) and Grounding Status at Move-Transitions

$p<0.05$ for length in seconds; t=2.8, df=7.0, $p<0.05$ for utterances). There are however no significant differences between the groups (χ^2) w.r.t. how much time was spent on different sub-tasks like identifying pieces or placements: the pairs in the noise condition don't do different things, they just do the same things for longer / more often.

3.3 Move-based Analysis

Table 1 shows the distributions of move outcomes. The majority of moves in the no-noise condition end with confident and successful placement. In contrast, in the noise condition only just over half of the moves are actually successful, and consequently there are more moves that are repairs of previous mistakes. The differences between the groups are significant (χ^2, for both $p<0.01$).

The mean length of moves in terms of utterances is very similar for both groups (28.5 for noise group, 30.81 for control group), and indeed the difference is not significant: there seems to be a constant upper limit on how much time is spent on each move before the players move on, confidently or not.

Table 2 shows the ratio of contributions by IG and IF within each move, averaged over all moves and separated according to *grounding status* and *description of state*; e.g., the "54/46" in the second line means that 54% of contributions in moves in the noise group that ended in a wrong placement came from IG and 46% from IF. Problems in a move that lead to an unsuccessful conclusion and/or not-confident grounding only in the control group had an effect on the contribution ratio, leading to more contributions by IG. (The differences are significant, χ^2 tested, * $p< 0.05$, *** $p<0.001$.)

	noise	no noise	signf.
all	55 / 45	57 / 43	
wrong	54 / 46	68 / 32	*
corr.	56 / 44	56 / 44	
!conf	54 / 46	74 / 26	***
conf	57 / 43	57 / 43	

Table 2: Ratio IG/IF contributions, by move success

3.4 Utterance-based Analysis

The recordings of the noise group have been segmented into 3249 utterances, those of the control group into 1607. In the noise group, there were 561 utterances that contained noise, i.e., 30.1% of all IG utterances (only those can contain noise). Only 28 of those (= 5.0%) triggered a clarification request (that is, were coded as being the antecedent of one). In the noise group, there was only one CR that was not triggered by a noise utterance; in the control group there were 8 CRs altogether.

The majority of turns (both of IG and IF; turn defined as sequence of utterances before speaker change), was one utterance long, this tendency being stronger in the control group (61.8% compared to 55.6% in the noise group; difference in length distribution is significant, χ^2, $p<0.001$). However, there were turns of length up to 13 utterances.

In all utterances within IG turns in the noise group (i.e., at all distances from the speaker transition), noise events were equally likely to occur. However, a noise event in an utterance *at* the transition point—that is, in either the last utterance of a longer turn or in a single utterance turn—had a chance of 8.33% of triggering a CR. A noise event one utterance away from the transition point only has a 0.87% likelihood of triggering a CR. There are no CRs in the corpus whose antecedent is further away.

Lastly, we turn to a more fine-grained analysis of the clarification requests that occurred. We compared the distributions of CR-features in this corpus with that resulting from the the other task done in the same setting, where items like strings of numbers and sentences were read from a screen by IG for the IF to write them down (see (Schlangen and Fernández, 2007)).

What is interesting here is that despite the manipulation being the same, there were significant differences in the CRs that occurred: in the puzzle task

of the present paper, there were significantly more CRs that did not point at the exact problem location (`extent`), more CRs that did not present a hypothesis (`severity`), fewer CRs constructed through repetition of material (`rel-antec`), and fewer replies to CRs that were repetitions, and more reformulations or elaborations (`answer`). (All differences were tested with a χ^2 test, $p<0.01$.)

4 Discussion and Conclusions

We now briefly summarise these observations: Pairs in the noise condition needed significantly longer to finish the task, and this was not due to higher effort for repairing understanding problems, but rather to higher effort needed for repairing task-level problems, i.e. wrong placements. In fact, while there were more repairs in the noise condition than in the control condition, most induced problems went unacknowledged – and as the performance differences show, it seems to be valuable information that they miss.

That CRs typically clarify the immediately preceding utterance has been observed before (Purver, 2004; Rodríguez and Schlangen, 2004). Our setting allows us to see the strength of this constraint: even if there are problems with earlier utterances within a turn—and we know that they are there, as we produced them—, they are a lot less likely to be repaired than those in the last utterance of a turn. We speculate that IF judged the information gain they would achieve by clarifying too low to take the step to interrupt IG's turns. They rather settled on a more independent strategy with more reliance on tentative placements (as shown by the grounding status), which for this task turned out to be less successful than understanding IG's commands. It seems that there needs to be a baseline of understanding before utterance-level clarification is even attempted.

Another interesting observation is that while the *forms* of the CRs that are present are not significantly different from those in comparable conditions but with different task (see previous section), the CRs are interpreted differently: significantly often, forms that trigger verbatim responses in that other corpus trigger reformulations or elaborations here. There are two possible explanations (not mutually exclusive): the CR addressees are more primed to expect clarification requests that target the meaning level (Clark, 1996) and hence treat the CRs as being such. Or, given the spontaneous, rather unplanned nature of these also often rather long description utterances, there are memory limitations that make verbatim responses harder.

To summarise, our results show that a) clarification is not *automatic*, but underlies complex considerations about the value of the missing information; b) CR forms are interpreted in a (task-)context-dependent way.

In future work, we will look in more detail at the dialogue acts of the utterances at turn-boundaries. We also plan to test task-performance in the same setting, but with the IF instructed to follow a clarification policy of 'always interrupt and clarify if there is noise'.[2]

References

Paul Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9–10):341–345.

Herbert H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge.

Marie Meteer and Ann Taylor. 1995. Dysfluency annotation stylebook for the switchboard corpus. http://www.cis.upenn.edu/∼bies/manuals/DFL-book.pdf.

Christoph Müller and Michael Strube. 2001. MMAX: A Tool for the Annotation of Multi-modal Corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 45–50, Seattle, USA, August.

Matthew Purver. 2004. *The Theory and Use of Clarification Requests in Dialogue*. Ph.D. thesis, King's College, Unversity of London, London, UK.

Kepa Joseba Rodríguez and David Schlangen. 2004. Form, intonation and function of clarification requests in german task-oriented spoken dialogues. In *Proceedings of Catalog (SemDial04)*, pages 101–108, Barcelona, Spain, July.

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, August.

Gabriel Skantze. 2005. Exploring human error recovery strategies: Implications for spoken dialogue systems. *Speech Communication*, 45(3):325–341.

[2]**Acknowledgements:** Thanks to: S. Bachmann, A. Steinhilber, H. Bohle (transcription and annotation); M. Waeltermann (noise program); J. Dreyer (ZAS Berlin), B. Pompino-Marshall (HU Berlin), P. Healey and G. Mills (QMU London) (lab use); M. Stede and A. Corradini (discussions of set-up). This work was supported by EU (Marie Curie Programme) and DFG (Emmy Noether Programme).

Dialogue Policy Learning for combinations of Noise and User Simulation: transfer results

Oliver Lemon
Edinburgh University
olemon@inf.ed.ac.uk

Xingkun Liu
Edinburgh University
xliu4@inf.ed.ac.uk

Abstract

Once a dialogue strategy has been learned for a particular set of conditions, we need to know how well it will perform when deployed in different conditions to those it was specifically trained for, i.e. how robust it is in *transfer* to different conditions. We first present novel learning results for different ASR noise models combined with different user simulations. We then show that policies trained in high-noise conditions perform significantly better than those trained for low-noise conditions, even when deployed in low-noise environments.

1 Introduction

For any dialogue system, a major development effort is in designing the *dialogue policy* of the system, that is, which dialogue actions (e.g. `ask(destination_city)` or `explict_confirm`) the system should perform. Machine-learning approaches to dialogue policies have been proposed by several authors, for example (Levin et al., 2000; Young, 2000; Henderson et al., 2005). These approaches are very attractive because of their potential in efficient development and automatic optimization of dialogue systems.

We will address the issue of whether policies trained for one dialogue situation can be used successfully in other dialogue situations (Paek, 2006).

For example, perhaps you have trained an optimal policy for an operating environment where the word-error rate (WER) is 5%, but you want to deploy this policy for a new application where you are not sure what the average WER is. So, you want to know how well the policy *transfers* between operating situations. Likewise, perhaps you have trained a policy on a data set of cooperative users, but you want to know how that policy will behave in contact with less co-operative users. So, you want to know how useful the policy is with different users.

These transfer issues are important because when deploying a real dialogue application we will not know these parameters exactly in advance, so we cannot train for the exact operating situation, but we want to be able to learn robust dialogue policies which are transferable to different noise/user/time-penalty situations, which we do not know about precisely before deployment.

1.1 Related work

The issue of policy transfer has been partially explored before as part of recent work on types of user simulations (Schatzmann et al., 2005). Here, the authors explore how well policies trained on different types of user simulation perform when tested with others. They train and test on three approaches to user simulation: a bigram model (Eckert et al., 1997), the Pietquin model (Pietquin, 2004), and the Levin model (Levin et al., 2000). They show that strategies learned with a "poor" user model can appear to perform well when tested with the same user model, but perform badly when tested on a "better" user model. However, the focus of (Schatzmann et al., 2005) is on the quality of the user simulation techniques themselves, rather than robustness of the learned dialogue policies. We will focus on one type of stochastic user simulation but different types of

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 55–58,
Antwerp, September 2007.

users and on different environmental conditions.

(Frampton and Lemon, 2006) train a policy for 4-gram stochastic user simulation and test it on a 5-gram simulation, and vice-versa, showing that the learned policy works well for the 2 different simulations. However, these simulations are trained on the same dataset (Walker et al., 2001) and thus do not simulate different *types* of user or noise conditions. Similarly (Henderson et al., 2005) test and train on different segments of the COMMUNICATOR data, so the results presented there do not deal with the issue of policy transfer. (Lemon et al., 2006) show that a single policy trained on a human-machine dialogue corpus also performs well with real users of a dialogue system.

2 The experimental set-up

We experiment with a 3-slot information-seeking system, resulting in 8 binary state variables (1 for whether each slot is filled, 1 for whether each slot is confirmed, 2 for whether the last user move was "yes" or "no"), resulting in 256 distinct dialogue states. There are 5 possible system actions (e.g. implicit-confirm, greet, present-info).

We use the SHARSHA Hierarchical Reinforcement Learning algorithm of REALL (Shapiro and Langley, 2002) to learn over the policy space for obtaining 3 information slots. For all combinations of Turn Penalty, noise, and user models we train each policy on 32,000 iterations (approx. 8000 dialogues). We then test each policy (including the hand-coded policies) over 1000 dialogues in the conditions for which they were trained. Statistical significance is measured by independent samples t-tests, over 1000 test dialogues.

We use the hierarchical structure of REALL (Shapiro and Langley, 2002) programs to encode commonsense constraints on the dialogue problem, while still leaving many options for learning. The hierarchical plans encode obvious decisions such as: " never confirm already confirmed slots".

2.1 Reward function

We use a reward function which incorporates noise modelling, as in (Rieser and Lemon, 2007). For each dialogue we have, as is now commonly used:

```
reward = completionValue
         - dialogueLength*TurnPenalty
```

However, for our noise modelling, the `completionValue` of a dialogue is defined as the percentage probability that the user goal is in the actual result set that they are presented with. See (Rieser and Lemon, 2007) for full details. In our experiments Low Noise (LN) means that there is a 100% chance of confirmed slots being correct and an 80% chance of filled (but not confirmed) slots being correct. In a real application domain we will not know these probabilities exactly, but we want to be able to learn dialogue policies which are transferrable to different noise situations, which we do not know about precisely before deployment.

2.2 Simulated users

We use 2 probabilistic user simulations: "Cooperative" (C) and "Uncooperative" (U). Each simulated user produces a response to the previous system dialogue move, with a particular probablility distribution conditioned on the previous system move. For example, if the system asks for slot1 (e.g. "what type of food do you want?") the cooperative user responds to this according to the a probability distribution over dialogue acts estimated from the COMMUNICATOR corpus (Walker et al., 2001).

In contrast, the "Uncooperative" user simply has a flat probability distribution over the all the possible dialogue acts: it is just as likely to be silent as it is to supply information. This is not intended to be a particularly realistic user simulation, but it provides us with behaviour that is useful as one end of a spectrum of possible behaviours.

2.3 Baseline hand-coded policies

The hand-coded dialogue policies obey the same commonsense constraints as mentioned above but they also try to confirm all slots implicitly or explicitly (based on standard rules) and then close the dialogue, except for cases where particular dialogue length thresholds are surpassed. For example, if the current dialogue length is greater than 10 the hand-coded policy will immediately provide information.

3 Results versus hand-coded policies

In general, learning takes about 500 dialogues before a policy of confirming as many slots as possible in the shortest time is discovered. Early in the training runs the learner experiments with very short

dialogues (smaller length penalties), but usually receives less completion reward for them and so learns how to conduct the dialogue so as to trade-off between turn penalties (TP) and completion value. For example, in the High Noise, Cooperative user, turn penalty 5 case, after a policy is discovered, testing the learned policy in the same situation (but with learning and exploration turned off), the average dialogue reward is 49.94 (see figure 1, plotting average reward every 50 test dialogues, and table 1).

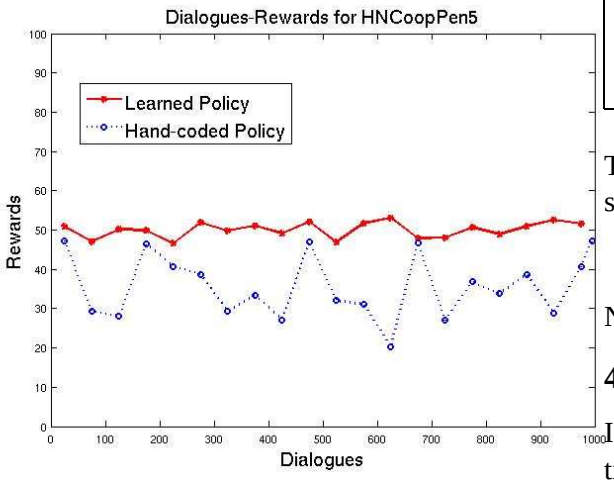


Figure 1: Testing: High noise, cooperative user, TP 5: Learned versus Hand-coded policy

Contrast this now with the performance of the hand-coded policy in the same situation (high noise, cooperative user, TP=5), over 1000 test dialogues, also shown in figure 1. The average reward for the hand-coded policy is 36.43 in these conditions, which means that the learned policy provides a relative increase in average reward of 37% in this case. This result is significant at $p < .01$.

Table 1 shows all results for the High Noise, Cooperative user case, for turn penalties (TP) ranging from 0 to 20. Here we can see that the learner is able to develop policies which are significantly better than the hand-coded policy. The exception is the TP=10 case, where the learned policy is not *significantly* better than the handcoded one ($p = .25$). For the significant results, the average relative increase in reward for learned policies is 28.4%

Considering the average dialogue lengths in each case, note that the hand-coded policy is able to complete the dialogues in, on average, fewer than 7 moves, which is less than the hand-coded length threshold (10). The learned policies, on the other hand, are able to discover their own local length/completion value trade-offs, and we see that, as expected, average dialogue length decreases as Turn Penalty increases.

	Learned Policy		Hand-coded Policy	
TP	Av. Reward	Length	Av. Reward	Length
0	85.70**	8.71	72.43	6.86
1	76.31 **	9.36	64.62	6.80
5	**49.94** **	7.18	**36.43**	6.95
10	4.16	4.05	1.77	6.89
20	-37.68 **	2.99	-63.76	6.80

Table 1: Results: Cooperative user, High Noise (**= significant at $p < .01$)

Similar results hold for the other combinations of Noise, User type, and Turn Penalty.

4 Transfer results

In the following experiments we chose to investigate the representative TP=5 case. We thus have 2 degrees of variation: user type (Cooperative/Uncooperative, C/U), and noise conditions (High/Low, H/L). Testing all combinations of these learned policies, for 1000 dialogues each, we obtained the results shown in table 5.

	Training			
Testing	C,L	C,H	U,L	U,H
C,L	**73.66**	**74.72**	54.86	54.48
C,H	**49.64**	**50.08**	21.07	25.36
U,L	23.67	27.84	**37.62**	**39.37**
U,H	09.99	14.40	**08.93**	**10.22**
Average:	39.24	*41.76*	30.62	32.36

Table 2: Transfer results for learned policies

Looking at table 2, we can see, for example, that training with a Cooperative user in Low noise (1st column) and testing with the same conditions (1st row) results in an average dialogue reward of 73.66. However, taking the same trained policy (C,L 1st column) and testing it with a Uncooperative user in High Noise conditions (row 4) results only in an average reward of 9.99. We would expect that the lead-

ing diagonal of this table should contain the highest values (i.e. that the best policy for certain conditions is the one trained on those conditions), but surprisingly, this is not the case. For example, training a C,H policy and testing it for C,L gives better results than training for C,L (and testing for C,L). This is significant at $p < .05$. This shows that a C,H policy in fact *transfers* well to C, L conditions.

Looking at the 4 policies C,L, C,H, U,L, and U,H we can see that C,H has the best transfer properties. Interestingly, C,H is the best policy for all of the testing conditions C,L, C,H, and U,H. But should we then train only in High noise conditions? Consider the following set of results (highlighted in bold font in table 5):

train C,H and test C,L $>$ train C,L and test C,L
train C,H and test C,H $>$ train C,L and test C,H
train U,H and test U,L $>$ train U,L and test U,L
train U,H and test U,H $>$ train U,L and test U,H

This indeed shows that it is better to train in High noise conditions than low noise, no matter what conditions you deploy in. These results are all significant at $p < .05$ except for the case "train C,H and test C,H $>$ train C,L and test C,H" ($p = .37$). This means that for cooperative users, training in High noise is *as good as* training in Low noise. These results show that, when training a policy for an operating environment for which you don't have much data (i.e. the developer does not yet know the noise and user characteristics) it is better to train and deploy a High noise policy, than to deploy a policy trained for Low noise conditions. Similar results show that policies trained on uncooperative users perform well when tested on cooperative users but not vice versa.

5 Conclusion

We addressed the robust of learned strategies in *transfer* to different conditions. We provided transfer results for dialogue policy learning and are the first to present results for different ASR noise models combined with different user models. We first showed that our learned policies for a range of environmental conditions (Noise, Users, Turn Penalties) significantly outperform hand-coded dialogue policies (e.g average 28% relative reward increase for cooperative users in high noise). We then compared different learned policies in terms of their transfer properties. We showed that policies trained in high-noise conditions perform significantly better than those trained for low-noise conditions, even when deployed in low-noise environments.

Acknowledgements This work is funded by the EPSRC (grant number EP/E019501/1) and by Scottish Enterprise under the Edinburgh-Stanford Link.

References

W. Eckert, E. Levin, and R. Pieraccini. 1997. User modelling for spoken dialogue system evaluation. In *Proceedings of ASRU*, pages 80–87.

Matthew Frampton and Oliver Lemon. 2006. Learning more effective dialogue strategies using limited dialogue move features. In *Proceedings of ACL*.

J. Henderson, O. Lemon, and K. Georgila. 2005. Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR data. In *IJCAI workshop on Dialogue Systems*.

O. Lemon, K. Georgila, and J. Henderson. 2006. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Evaluation. In *Proc. ACL/IEEE 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):11–23.

Tim Paek. 2006. Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical deployment. In *Dialogue on Dialogues*. Interspeech2006 - ICSLP Satellite Workshop.

Olivier Pietquin. 2004. *A Framework for Unsupervised Learning of Dialogue Strategies*. Presses Universitaires de Louvain, SIMILAR Collection.

V. Rieser and O. Lemon. 2007. Learning dialogue strategies for interactive database search. In *Interspeech*.

J. Schatzmann, M. N. Stuttle, K. Weilhammer, and S. Young. 2005. Effects of the user model on simulation-based learning of dialogue strategies. In *IEEE ASRU Workshop*.

D. Shapiro and P. Langley. 2002. Separating skills from preference: using learning to program by reward. In *Intl. Conf. on Machine Learning.*

M. Walker, R. Passonneau, and J.Boland. 2001. Quantitative and qualitative evaluation of DARPA Communicator spoken dialogue systems. In *Proc. ACL*.

Steve Young. 2000. Probabilistic methods in spoken dialogue systems. *Philosophical Transactions of the Royal Society (Series A)*, 358(1769):1389–1402.

Dynamic n-best Selection and Its Application in Dialog Act Detection

Junling Hu, Fabrizio Morbini, Fuliang Weng
Bosch Research and Technology center
4009 Miranda Ave.
Palo Alto, CA 94304
{junling.hu, fabrizio.morbini, fuliang.weng}@us.bosch.com

Xue Liu
School of Computer Science
McGill University
Montreal, QC H3A 2A7
Canada
xueliu@cs.mcgill.ca

Abstract

We propose dynamically selecting n for n-best outputs returned from a dialog system module. We define a selection criterion based on maximum drop among probabilities, and demonstrate its theoretical properties. Applying this method to a dialog-act detection module, we show consistent higher performance of this method relative to all other *n*-best methods with fixed n. The performance metric we use is based on ROC area.

1 Introduction

Recent years have seen increasing application of machine learning in dialog systems. From speech recognizer, to natural language understanding and dialog manager, statistical classifiers are applied based on more data available from users. Typically, the results from each of these modules were sent to the next module as n-best list, where n is a fixed number.

In this paper, we investigate how we can dynamically select the number n for *n-best* outputs returned from a classifier. We proposed a selection method based on the maximum drop between two adjacent probabilities of the outputs, where all probabilities are sorted from the highest to lowest. We call this method n*-best selection, where n* refers to a variable n.

We investigated the theoretical property of n*-best, particularly its optimality relative to the fixed n-best where n is any fixed number. The optimality metric we use is ROC (Receiver Operating Characteristic) area, which measures the tradeoff of false positive and false negative in a selection criterion. We test the empirical performance of n*-best vs. n-best of fixed n for the task of identifying the confidence of dialog act classification. In two very different datasets we use, we found consistent higher performance of n*-best than n-best for any fixed n.

This paper is the first attempt in providing theoretical foundation for dynamically selecting n-best outputs from statistical classifiers. The ROC area measure has recently been adopted by machine learning community, and starts to see its adoption by researchers on dialog systems.

Even though n*-best method is demonstrated here only for dialog act detection domain, it can be potentially applied to speech recognition, POS (part-of-speech) tagging, statistical parser and any other modules that return n-best results in a dialog system.

2 Dynamically selecting n for *n*-best outputs

The *n*-best method has been used extensively in speech recognition and NLU. It is also widely used in machine translation (Toutanova and Suzuki, 2007). Given that the system has little information on what is a good translation, all potential candidates are sent to a later stage, where a ranker makes a decision on the candidates. In most of these applications, the number of candidates *n* is a fixed number. The *n*-best method works well when the system uses multi-pass strategy to defer decision to later stage.

2.1 n*-best Selection

We call *n*-best* a variant of *n*-best where *n* is a

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 59–62,
Antwerp, September 2007.

variable, specifically the *n**-best method selects the number of classes returned from a model, such that the number *n** satisfies the following property:

$$n^* = \arg\max_n (p_n - p_{n+1}) \qquad (1)$$

where p_n and p_{n+1} are the probabilities of class *n* and class *n*+1 respectively. In other words, *n** is the cut-off point that maximizes the drop $p_n - p_{n+1}$.

2.2 Theoretical Property of n*-best

We have the following observation: When the output probabilities are ranked from the highest to the lowest, the accumulated probability distribution curve is a concave function.

We further show that our derivation of *n** is equivalent to maximizing the second derivative of the accumulative probability curve, when the number of classes approaches infinity. In other words,

$$n^* = \arg\max_n (-P''(n+1)),$$

Due to the page limit, we omit the proof here.

3 Evaluation Metric

To compare the performance of the *n**-best method to *n*-best selection of fixed n, we need to define an evaluation metric. The evaluation is based on how the n-best results are used.

3.1 The Task: Dialog Act Detection

The task we study here is described in Figure 1. The dialog-act classifier uses features computed from the parse tree of the user utterance to make predictions on the user's dialog acts.

The n-best results from the dialog-act classifier are sent to the decision component that determines whether the system is confident about the result of the classifier. If it is confident, it will pass the result to later stages of the dialog system. If it is not confident, the system will respond "I don't understand" and save the utterance for later training.

The decision on how confident we are about inter preting a sentence translates into a decision on whether to select that sentence for re-training. In this sense, this decision problem is the same as active leaning.

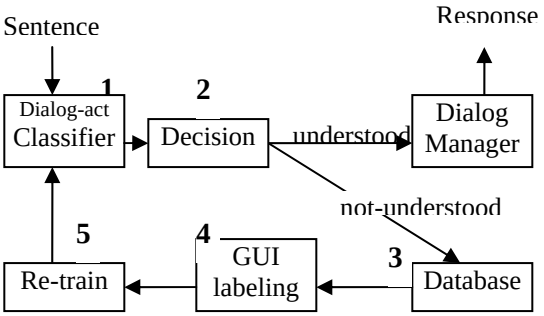


Figure 1. Detection Dialog Act with Confidence

3.2 Error Detection as Active Learning

Let *S* be the collection of data points that are marked as low confidence and will be labeled by a human. Let N_2 be the set of all new data. Let *h* be the confidence threshold and n the number we return from n-best results. We can see that (Figure 2) *S* is a function of both *n* and *h*. For a fixed *h*, the larger *n* is, the smaller *S* will be.

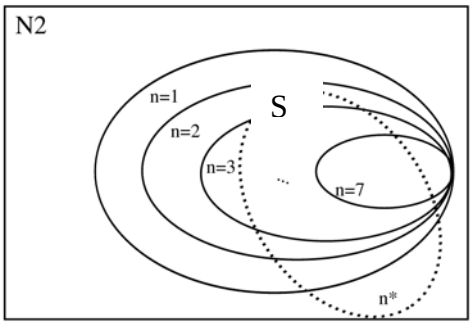


Figure 2 The Decreasing set of S as n increases

Our goal is to choose the selection criterion that produces a good *S*. The optimal *S* is one that is small and contains only true negative instances.

In active learning research, the most commonly used evaluation metric is the error rate (Tur et al, 2005; Osugi et al, 2005). The error rate can also be written as $1 - \frac{TP}{TP + FP}$, where *TP* is the number of true positives and *FP* is the number of false positives. This measure does not capture the trade off between giving the user wrong answers (false positive) and rejecting too many properly classified

user utterances (false negatives). We find a better measure that is based on ROC curve.

3.3 ROC curve and ROC Area

ROC (Receiver Operating Characteristic) curve is a graphical plot of the fraction of true positives vs. the fraction of false positive. ROC curve is an alternative to classical machine learning metrics such as misclassification rate.

An ROC space is defined by *FPR* (False Positive Rate) and *TPR* (True Positive Rate) as *x* and *y* axes respectively, where

$$FPR = 1 - \frac{TN}{FP + TN}, \quad TPR = \frac{TP}{TP + FN}$$

The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing the case in which all only true positives are returned by a particular model. The 45 degree diagonal line is called the no-discrimination line and represents the classifier that returns the same percentage of true positive and false positive.

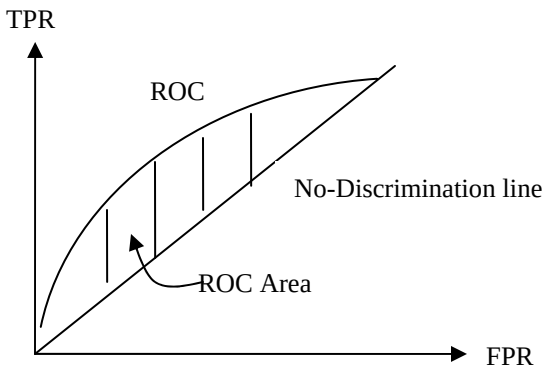


Figure 3. ROC curve and ROC area

4 Experimental Results

We tested the performance of our *n**-best method on two datasets. The first dataset contains 1178 user utterances and the second one contains 471 utterances. We use these two sets to simulate two situations: **Case 1,** a large training data and a small testing set; **Case 2,** a small training data and a large testing set.

4.1 Experimental data

All utterances in both datasets were hand labeled with dialog acts. There can be more than one dialog act associated with each utterance. An example of training instance is: "*(a cheap restaurant), (Query:restaurant, Answer, Revision)*" the first part is the user utterance, the second part (referred as L_d) is the set of human-labeled dialog acts. In total, in the domain used for these tests, there are 30 possible user dialog acts.

We compared *n**-best with fixed *n*-best methods with *n* from 1 to 6. For each of these methods, we calculate *TP*, *FP*, *TN* and *FN* for values of the threshold *h* ranging from 0.1 to 1 in steps of 0.05. Then we derived *TPR* and *FPR* and plotted the ROC curve.

Figure 4 shows the ROC curves obtained by the different methods in **Case 1**. We can see that the ROC curve for *n**-best method is better in most cases than the other methods with fixed *n*.

Figure 5 shows the ROC curves in **Case 2**, where the model is trained on a small dataset and tested on a large dataset. We can see that the ROC curves for all methods are nearer to the non-discrimination line than in the previous case. This suggests that the classifier has a lower discrimination quality given the small set used for training. However, the *n**-best method still out-performs the other *n*-best methods in the majority of scenarios.

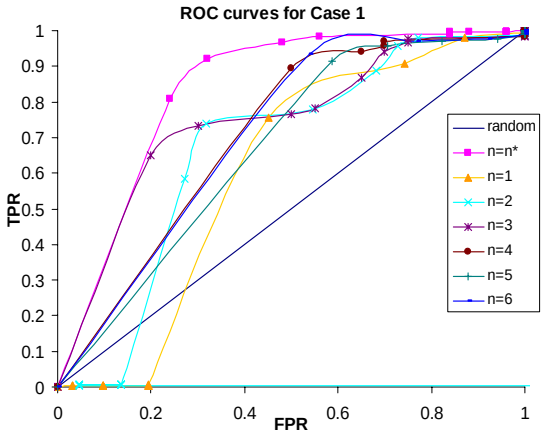


Figure 4. ROC curves from n*-best and n-best

To get a summary statistics, we calculated the size of the ROC area. Figures 6 and 7 plot the size of the ROC area of the various methods in the two test cases. We can see that *n**-best out-performs all other *n*-best methods.

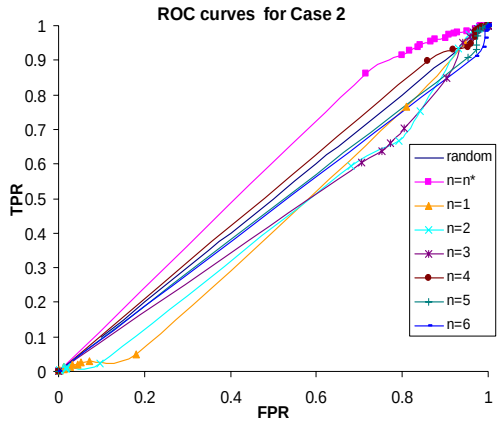


Figure 5. ROC curves obtained by n* and n-best .

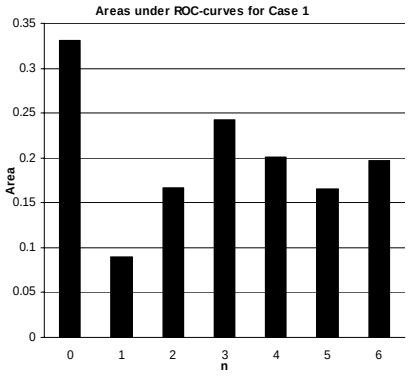


Figure 6. ROC Area for n*-best and n-best (*n** is represented as n=0)

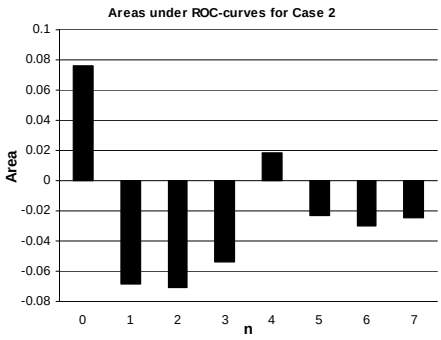


Figure 7. ROC Area for n*-best and other n-best methods (*n** is represented as n=0)

5 Conclusions

We propose dynamic selecting n for n-best outputs returned from a classifier. We define a selection criterion based on maximum drop among probabilities, and call this method *n**-best selection. We demonstrate its theoretical properties in this paper.

We measured the performance of our *n**-best method using the ROC area that has been designed to provide a more complete performance measure for classification models. We showed that our *n**-best achieved better ROC curves in most cases. It also achieves better ROC area than all other *n*-best methods in two experiments (with opposite properties).

Our method is not limited to detection of dialog acts but can be used also in other components of dialog systems.

References

C. Cortes, M. Mohri. 2004. AUC optimization vs. error rate minimization. *Advances in Neural Information Processing Systems* 16, eds., Sebastian Thrun, Lawrence Saul, and Bernhard Sch olkopf, MIT Press, Cambridge, MA.

Matt Culver, Deng Kun, and Stephen Scott. 2006. Active Learning to Maximize Area Under the ROC Curve. *Proceedings of the Sixth International Conference on Data Mining, IEEE Computer Society.* 149-158.

Sangkeun Jung, Cheongjae Lee, Gary Geunbae Lee. 2006. Dialog Studio: An Example Based Spoken Dialog System Development Workbench. 2006. *Proceedings of the Dialogs on dialog: Multidisciplinary Evaluation of Advanced Speech-based Interactive Systems*. Interspeech2006-ICSLP satellite workshop, Pittsburgh.

Thomas Osugi, Deng Kun, and Stephen Scott. 2005. Balancing Exploration and Exploitation: A New Algorithm for Active Machine Learning boundaries. *Proceedings of the Fifth IEEE International Conference on Data Mining* (ICDM'05). 330-337.

Kristina Toutanova and Hisami Suzuki. 2007. Generating Case Markers in Machine Translation. *Proceedings of NAACL-HLT 2007*, Rochester, New York. 49-56.Matt Culver, Deng Kun, and Stephen Scott. 2006. Active Learning to Maximize Area Under the ROC Curve. *Proceedings of the Sixth IEEE International Conference on Data Mining*. 149-158.

Gokhan Tur, Dilek Hakkani-Tür and Robert E.Schapire. 2005. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171-186.

Emergent Conversational Recommendations: A Dialogue Behavior Approach*

Pontus Wärnestål, Lars Degerstedt, Arne Jönsson
Department of Computer Science
Linköping University, Sweden
{ponjo,larde,arnjo}@ida.liu.se

Abstract

This paper presents and evaluates a behavior-based approach to dialogue management, where a system's complete dialogue strategy is viewed as the result of running several dialogue behaviors in parallel leading to an emergent coherent and flexible dialogue behavior. The conducted overhearer evaluation of the behavior-based conversational recommender system CORESONG indicates that the approach can give rise to informative and coherent dialogue; and that a complete dialogue strategy can be modeled as an emergent phenomenon in terms of lower-level autonomous behaviors for the studied class of recommendation dialogue interaction.

1 Introduction

The purpose of a *recommender system* is to produce personalized recommendations of potentially useful items from a large space of possible options that is hard to manually browse or search. *Conversational* Recommender Systems (CRSs) approach user preference acquisition from a dialogue point of view, where preferences are captured and put to use in the course of on-going natural language dialogue. The approach is motivated by its aim to make interaction efficient and natural (Burke et al., 1997; Thompson et al., 2004), to acquire preferences from the user in a context when she is motivated to give them (Carenini et al., 2003), as well as to facilitate exploration of the domain and the development of the user's preferences (Wärnestål, 2005). A CRS's *dialogue strategy* to achieve these aspects of the interaction is thus crucial for its performance and usability. In particular, we are interested in exploring robust and emergent factual and preferential dialogue with recommendation capabilities.

This paper presents our behavior-based approach to dialogue management and reports on an evaluation of the CRS CORESONG's dialogue behaviors.

2 Dialogue Behaviors in Recommendation Dialogues

By a *dialogue behavior* of a dialogue agent, we understand a conceptual and computational functionality in the agent's dialogue strategy. Computationally, a dialogue behavior is coded into a *Dialogue Behavior Diagram* (DBD), that describes a state automaton where each state contains (one or more) commands and transitions with optional conditions. The DBD automaton is similar to the UML activity diagram.

DBDs invoke, and use, results from other software modules, denoted jointly as *external resources* (e.g. databases and recommender engines).

Four DBDs constitute the complete recommendation dialogue model: **Conventional**, **Direct Delivery**, **Indirect Delivery**, and **Interview**. A more detailed account of each of these behaviors are found in (Wärnestål et al., 2007).

Delivery Behaviors On a fundamental level, the goal for CORESONG (or any recommender system)

*This work is supported by the Swedish National Graduate School for Language Technology (GSLT), and Santa Anna IT Research.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 63–66,
Antwerp, September 2007.

is to provide the user with a delivery, such as an explicitly requested piece of information from a database resource, or a recommendation from a recommender engine. The **direct** delivery typically uses a database that the user can query. In cases where a successful database call has been made (that is, a non-empty result set not larger than a predefined size limit is returned as response to a user's request), a delivery is written to the out stream. In order to support domain exploration and to drive the dialogue forward (Wärnestål, 2005), positive preference statements trigger the system to fetch information *related* to the preference from an external database resource (e.g. utterances S3a, S4a and S5a in Figure 1).

While the direct delivery behavior is functional for information-providing dialogue, and can cater for basic recommendations, it has been found that human-like recommendations occur in a slightly different fashion (Wärnestål et al., 2007). Therefore, the **indirect** delivery behavior has been designed (Wärnestål et al., 2007). It provides a *motivation* before presenting the actual recommendation (as exemplified by utterances S5b/S5c in Figure 1). Since the system cannot know whether the suggestion is previously familiar to the user, it delivers the recommendations in the form of questions (S5c).

Interview Behavior The purpose of the interview behavior is to collect relevant information about domain entity types (e.g. genres, artists or albums in the music domain) or items. This is a useful dialogue behavior in cases where deliveries cannot be completed due to e.g. ambiguous or incomplete user requests. It is also useful in preferential interviews used for recommendations, where the system acquires user preferences to be used by a recommender engine resource. A particular system may thus employ several interview DBD instances, connected to different external resources.

2.1 Emergent Dialogue

We view the dialogue system's complete behavior as emerging from the different DBDs. By *emergent* functionality in a system, we understand components that operate simultaneously in order to achieve a desired behavior. This is contrasted to hierarchical systems, where sub-functions are invoked from a central component or representation.

U1	What genres are there?
S2a	I found these genres in the database: [...].
S2b	I'd like to know more about your preferences.
S2c	What else do you like?
U2	I like the genre Rock&Roll
S3a	These artists belong to the genre Rock&Roll: [...].
S3b	I'd like to know more about your preferences.
S3c	What else do you like?
U3	What songs were made by Elvis Presley?
S4a	I found these songs by the artist Elvis Presley in the database: [...].
S4b	I'd like to know more about your preferences.
S4c	What else do you like?
U4	I like the album Live at Madison Square Garden
S5a	These songs belong to the album Live at Madison Square Garden: [...].
S5b	You might like the song Suspicious Minds because it is a Rock&Roll song by Elvis Presley.
S5c	Have you heard it?
U5	Yes
S6a	Ok. What do you think about it?
U6	It's good

Figure 1: *Dialogue log from* CORESONG *interaction (*RECOMMENDER *experiment configuration). [...] denotes lists of genres, artists, albums or songs. S = system, U = user.*

Our approach to dialogue system design is inspired by the layered subsumption architecture (Brooks, 1991) where layers correspond to behaviors that are organized hierarchically, and where higher-level behaviors can *subsume* lower-level layers by inhibition or modification.

A dialogue agent's complete strategy is described by a set of DBD instances that run as a DBD *strata machine*. The DBD strata machine streams input and merges each behavior's output (see Figure 2). There is no central representation of the complete dialogue, and the individual behaviors do not model each other since each DBD processes the incoming token stream autonomously. Therefore, the outputs from the DBDs need to be integrated (and typically reduced) into a coherent system turn, and is managed by two constructs in the Output Weaver: *behavior priority* and an *order heuristic*.

Behavior Priority DBDs are indexed with a priority and order the out statements accordingly (ascending order). The *request* with highest priority will be chosen. This hinders the occurrence of two requests back to the user which obviously could be confusing. The order of CORESONG's DBDs are (lowest to highest priority): Conventional, Direct Delivery,

Indirect Delivery, and Interview (Figure 2). DBD instances connected to the recommender engine have higher priority than those of the music database[1].

Order Heuristic Due to the behavior priority, there is only one request action available each turn. The order heuristic places this request at the end of the output, so that *informing* system action statements are guaranteed to precede the request. This guarantees that the constrain request (S2c) in the first system utterance in Figure 1 always occur after the direct delivery (S2a) even though their statements origin from different DBD instances.

3 Experiment

To validate the behavior based approach to dialogue management we conducted an "overhearer" experiment (Whittaker and Walker, 2004) by using four different behavior configurations of the CORESONG system (see Table 1). The reason for using the overhearer model is to avoid natural language interpretation problems (since the coverage of grammar and lexicon is not our focus), and letting personal music preferences that may not be covered by our recommender engine and database affect the subjects' experience of dialogue interaction. The experiment was run with 30 subjects.

3.1 CoreSong

Configuration of dialogue behaviors and attached external resources is easily done in CORESONG by switching DBD instances on or off. The two external resources used by the DBD instances are (a) a music information database and (b) a content-based recommender engine (Burke, 2002).

A DBD instance implementation consists of defining LookUp calls, and the surface realization of the action statements in the DBDs.

The Input Streamer (IS) feeds the interpretations of user input to each of the DBD instances in the DBD strata machine. Each DBD instance processes the input and writes to an out stream using the command out. The Output Weaver module (OW) then weaves together each DBD instance's output as outlined in Section 2.1.

[1]Note that interview and delivery behaviors of the same external resource are naturally designed to be mutually exclusive.

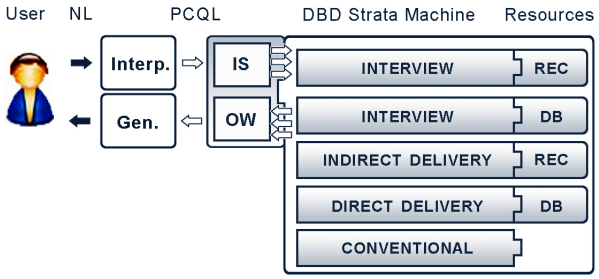


Figure 2: *The standard CoreSong behavior configuration, with database (DB) and recommender engine (REC), interview and delivery behaviors. Interp = Interpretation Module, Gen = Generation Module, IS = Input Streamer, OW = Output Weaver.*

Table 1: Experiment configurations. DD = Direct Delivery, IW = Interview, ID = Indirect Delivery, Db = Database, R = Recommender Engine.

Config.	DD(Db)	IW(Db)	DD(R)	ID(R)	IW(R)
Q-A	x	x			
BLUNT	x	x	x		x
PRYING				x	x
REC	x	x		x	x

Four different DBD instance configurations were used to generate the test dialogues, as shown in Table 1. The different configurations effectively modify CORESONG's complete dialogue strategy. Q-A, for example, with only the database resource, results in a question-answer system without recommendation capabilities, whereas the PRYING configuration supports a preference interview but with no power to deliver answers to factual requests. The BLUNT configuration has the power to deliver both database results and recommendations; but the recommendations are not delivered with motivations and follow-up questions as the indirect delivery (RECOMMENDER configuration) is designed to do. Figures 1 (RECOMMENDER) and 3 (BLUNT) exemplify the differences.

3.2 Procedure

Each subject was presented with the four test dialogues, one at a time, displayed in a web browser. For each of the dialogues they were asked to fill

U1	What genres are there?
S2a	I found these genres in the database: [...].
S2b	What else do you want to know?
U2	I like the genre Rock&Roll
S3a	These artists belong to the genre Rock&Roll: [...].
S3b	What else do you want to know?
U3	What songs were made by Elvis Presley?
S4a	These songs belong to the artist Elvis Presley: [...].
S4b	What else do you want to know?
U4	I like the album Live at Madison Square Garden
S5a	These songs belong to the album Live at Madison Square Garden: [...].
S5b	You might like the song Suspicious Minds.
S5c	What else do you like?

Figure 3: *Dialogue sample for the* BLUNT *configuration.*

out a questionnaire on a 5-point Likert-scale regarding their agreement with four statements, intended to determine *informativeness*, *preference modeling*, *coherence*, and *naturalness* of the dialogue excerpts. For example, the statement: "The system's utterances are easy to understand and provide relevant information" reflects informativeness (Whittaker and Walker, 2004).

4 Results and Discussion

In general, the participants considered the Q-A and RECOMMENDER configurations to have the highest informativeness (86.2% and 85.5% respectively). This is expected, since they both are equipped with the database direct delivery behavior. The PRYING configuration, lacking in database delivery functionality, received a lesser rating on informativeness. For our current work, the notion of coherence is of high importance, since this quality of the dialogue was thought to be at risk when abandoning a monolithic dialogue strategy model. It is interesting that the coherence measure is high for all configurations: PRYING (70.3%), BLUNT (79.3%), RECOMMENDER (84.1%) and Q-A (86.2%). Furthermore, the RECOMMENDER configuration was high-ranking in all four aspects: Informativeness (85.5%), preference management (80.0%), naturalness (79.3%), and coherence (84.1%).

The data for the configurations over the parameters were compared using a one-way analysis of variance (ANOVA)[2]. Preference management was perceived as significantly lower in the Q-A configuration compared to the other three configurations, where preferences indeed were modeled and de facto influenced the dialogue. PRYING received significantly lower ratings on coherence compared to the other three configurations. This is most likely due to that factual user queries were only used as indicators of preferences, and were not responded to in the way that configurations with delivery behaviors did. The RECOMMENDER configuration received a significantly higher rating on naturalness compared to the other three configurations.

The results show that BCORN's non-centralized approach that views dialogue strategy modeling as an emergent phenomenon is feasible, and encourages future development of the approach. They also imply that natural and coherent recommendation dialogue can be explained in terms of the suggested dialogue behaviors.

[2] $p < 0.001$ n.s. for all differences reported below.

References

Rodney A. Brooks. 1991. Intelligence without representation. *Artificial Intelligence*, 47:139–159.

Robin D. Burke, Kristian J. Hammond, and Benjamin C. Young. 1997. The findme approach to assisted browsing. *IEEE Expert*, 12(4):32–40.

Robin D. Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, 12:331–370.

Giuseppe Carenini, Jocelyin Smith, and David Poole. 2003. Towards More Conversational and Collaborative Recommender Systems. In *Proceedings of the International Conference of Intelligent User Interfaces*, pages 12–18, Miami, Florida, USA.

Cynthia Thompson, Mehmet Göker, and Pat Langley. 2004. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, 21:393–428.

Steve Whittaker and Marilyn Walker. 2004. Evaluating dialogue strategies in multimodal dialogue systems. In W. Minker, D. Bühler, and L. Dybkjaer, editors, *Spoken Multimodal Human-Computer Dialogue in Mobile Environments*, pages 247–268. Kluwer Academic Publishers.

Pontus Wärnestål, Lars Degerstedt, and Arne Jönsson. 2007. Interview and delivery: Dialogue strategies for conversational recommender systems. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (Nodalida)*, Tartu, Estonia, May.

Pontus Wärnestål. 2005. User evaluation of a conversational recommender system. In Ingrid Zukerman, Jan Alexandersson, and Arne Jönsson, editors, *Proceedings of the 4th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 32–39, Edinburgh, Scotland U.K.

Exploiting Semantic and Pragmatic Information for the Automatic Resolution of Spatial Linguistic Expressions

Andrea Corradini
Computational Linguistics Department
University of Potsdam, D-14476 Golm, Germany
andrea@ling.uni-potsdam.de

Abstract

We present a computational model for the interpretation of linguistic spatial propositions in the restricted realm of a puzzle game. Based on an experiment aimed at analyzing human judgment of spatial expressions, we establish a set of criteria that explain human preference for certain interpretations over others. Each criterion is associated to a metric that combines the semantic and pragmatic contextual information regarding the game as well as the utterance being resolved. By resorting to machine learning techniques we determine a model of spatial relationships from the data collected during the experiment. Sentence interpretation occurs by matching the potential field of each of its possible interpretations to the model at hand. The system's explanation capabilities lead to the correct assessment of ambiguous situated utterances for a large percentage of expressions.

1 Introduction

The interpretation of spatial expressions is an important aspect of human cognition. Several experimental and theoretical studies have analyzed how language is linked to the non-linguistic spatial world with the goal to shed some light on the human mental processes that underlie the understanding of linguistic utterances involving space. Findings from these research endeavors have paved the way for the development of computational systems able to analyze, interpret and generate natural language descriptions of space and the physical world.

In this work, we focus on the interpretation of three types of linguistic relationships that form the basis for spatial expressions: topological relations like "*near*", projective relations such as "*left of*", and the relation "*between*". Projective relations need the specification of a frame of reference.

Within the scenario of a speech-operated 2D puzzle game, we have been developing a computing system able to understand the meaning of and consequently act upon linguistic instructions like e.g. "*land the green piece over the T-shaped one*" that can be ambiguous to a human who is not embedded in the same situation and sharing the same conversational context of the speaker/writer.

The paper is structured as follows. First, we discuss relevant related works. We then present the motivation for this research and the computational model that we developed based on the experiments we carry out. Eventually, we propose a system evaluation and a discussion on future extensions.

2 Related Work

Researchers in the field of language-oriented artificial intelligence have proposed several methods to deal with the inherent ambiguity of language and to handle traditional linguistic phenomena like presupposition, quantification, anaphora, under specification, and elliptic expressions. In parallel to research on these well-known sources of ambiguity, the understanding of propositions that depend on situational context has emerged as an active area of study and the treatment of spatial information in utterances has evolved into an ever growing field.

A relevant number of conceptual models that relate language to visual spatial information have been proposed (Eschenbach 1999; Tapus et al., 2005). Backed by theoretical works and/or empirical experiments (Costello & Kelleher, 2006; Logan & Sadler, 1996), more and more computational models that exploit the potential of verbal communication to interact with visual or spatial data have been implemented particularly for natural language interfaces to graphical systems and human-robot interaction.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 67–70,
Antwerp, September 2007.

The SHRDLU system (Winograd, 1971) is probably the first relevant work that shows how syntax, semantics, and reasoning about the world can be successfully combined to produce a system that understands natural language to control the actions of a simulated robot arm. Following this pioneering work, other prototypes and models have been put forward for topological and projective relations. Several works based on language modeling and visual context (Gorniak & Roy, 2004; Roy et al., 2002; Roy & Mukherjee, 2005) involve aspects of grounded situation model. These approaches lead to the development of visual context sensitive grounded systems that understand, learn and generate natural language. A research methodology that addresses common problems in spatial communication arising during human-robot conversation is outlined in (Moratz et al., 2001). In (Kelleher et al., 2005) visual information, context and salience are integrated to leverage the understanding and generation of spatial expressions in the context of virtual reality applications. A variety of metrics and potential field measures are introduced in (Kelleher et al., 2006; Regier & Carlson, 2001) as a powerful tool to model and characterize spatial relations among 2D objects as perceived by human subjects. An integration of potential field models with visual information to control a robot that follows natural language commands to perform manipulative actions is presented in (Brenner et al., 2007) for the task of action planning in situated communication. In (Gorniak & Roy, 2005; Gorniak et al., 2006) the use of situated communication in computer games is investigated.

Excluding (Roy et al., 2002), the works outlined above have not resorted to machine learning techniques. Our work shares with (Kelleher et al., 2005; Kelleher et al., 2006; Regier & Carlson, 2001) the idea of encoding spatial information using a set of local metrics. It differentiates from them in the way we perform the assessment of the values of the metrics.

3 Resolving Spatial Expressions

3.1 Situated Communication in Pentomino

Pentomino is a popular recreational math puzzle game. The game consists of twelve different pieces that are built as arrangement of five square units joint along their edges. The objective is to fill up a given game board using all pieces. To accomplish this task, players can select, rotate, translate, flip, remove, mirror, and land pieces onto the board. In early studies on human-human communication to play Pentomino, we noticed that subjects resort extensively to localization expressions when they intend to collaboratively resolve a puzzle thus making this game an excellent prototyping arena for situated natural language understanding.

Our model is integrated into a digital version of Pentomino where speech can be used as a complementary input mode (Corradini et al., 2007). We exploit the game semantics and pragmatic along with context information available from both the visual display on the user interface and the game history to interpret spatial expressions used to play the game. At anytime, the player is allowed to customize a few application settings that affect the visual feedback and thus in turn visual-grounding (Roy et al., 2002; Roy & Mukherjee, 2005) of context information that bridges the symbolic realm of linguistic concepts with entities in the game world.

3.2 An Experimental Study

To investigate human interpretation of spatial situations, we run a psycholinguistic experiment that parallels the task of an automated system for playing Pentomino. We collected data from 38 participants (22 males and 16 females) both native and non-native English speakers with age ranging from 13 to 72 years ($\mu = 31.3$, $\sigma = 13.5$). Subjects were given a set of 40 image-text pairs and instructed about the game objective and rules. We showed the subjects a snapshot of a puzzle game and the next instruction to carry out in text format as a single separate instruction. Subjects were then asked to update the board according to their interpretation of the instructions with the goal to maximize the possibility to finish the game after carrying out the move. We chose such a setting both to elicit controlled spatial interpretations in different situations and to collect data that can give insights on factors, motivations, and mechanisms that play a role in turning the mental picture of a linguistic sentence into an actual spatial configuration.

A post-study analysis of the corpus of 1520 task solutions showed that while all subjects implicitly used themselves as frame of reference (see Figure 1) a few different configurations were proposed for each single task. One annotator searched for pragmatic and semantic errors in the solved tasks. We considered as a pragmatic error any spatial ma-

nipulations that, once performed, would at once appear to lead to no game solution i.e. result in the creation of one or more islands of cells with less units than the number of squares making up a single Pentomino piece. We refer to these small holes onboard to as *smHoles* (see Figure 1). We classified as semantic errors all cases of spatial actions and instructions that violate the game rules or were impossible to carry out. A second annotator scored 24 randomly selected user forms i.e., a 63.2% random sample. Compared to the first annotator there was a 98% match on what the error events were. In total, we found an average 8.3% of pragmatic errors ($\mu = 4.8$, $\sigma = 4.6$) and a negligible 0.02% ($\mu = 0.6$, $\sigma = 1.5$) of semantic errors. After removing these error cases from our corpus, we analyzed the remaining 1394 picture-instruction couples (91.7% of the data) to infer a best estimate of the space considered by the subjects given a spatial relation among reference objects.

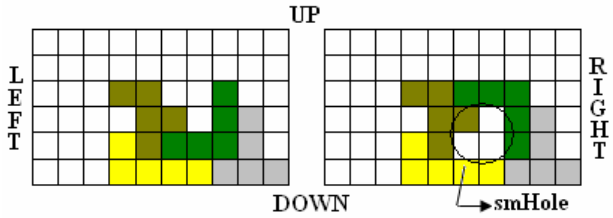


Figure 1. (left) A correct semantically and pragmatically interpretation of the instruction used in Section 1; (right) a pragmatically incorrect one. Text around the borders indicates the implicit frame of reference.

The computational model we developed bases on both the analysis of the data collected and the fact that in the context of a restricted language and limited number of visual entities, subjects tend to refer to objects by listing their properties and attributes such as color, shape and size (Roy, 2002).

3.3 Criteria & Metrics

From data of our experiment, we realized that for relations of the kind "*near*", "*under*", "*left to*" etc., over 97% of the subjects considered locations on the board grid that are within a certain small distance to the referent. In the case of "*between*" relations, 87% of the subjects considered points at locations mid-way to the referents. According to the relation at hand, we refer to the area including the points that satisfy the proximity requirement as *region of interest* or *RoI* in short. It restricts the set of possible locations referred to in the utterance.

We define a series of metrics over the *RoI* based on the notion of field potential (Kelleher et al., 2006). They describe degrees of likelihood of acting upon an object at a given location according to a set of criteria that capture and incorporate the most commonly used interpretation strategies adopted by subjects of our experiment. Given a sentence that refers to object *Obj* via a spatial relation *Rel* to another reference object *Ref*, they are motivated by the observation that people tend to:

C1) operate on *Obj* that is as closer as possible to *Ref* (*Proximity criterion*)

C2) operate on *Obj* at positions that maximize the number of physical contacts with other game entities such board edges or other pieces (*Adherence criterion*)

C3) operate on *Obj* at positions that maximize the intersection area between *Obj* and the *RoI* (*Communality criterion*)

C4) operate on *Obj* at positions that either minimize distance between *Obj*'s and *Ref*'s centers of mass or, in case of a "*between*" relation, are equidistant from those of all other referents (*Center of Mass criterion*)

C5) Play uniformly i.e. they concentrate on a region on the board which try they fill in incrementally before moving to other distant areas of the board (*Location Saliency criterion*)

C6) Avoid the creation of *smHoles* since they make the game unsolvable (*Fillability criterion*)

The criterion C6 captures aspects relative to game pragmatics and semantic knowledge. Criteria C1 to C4 reflect game's geometrical considerations at a given time. The criterion C5 accounts for the dialogue context in terms of game history. For each criterion we defined a corresponding metric to quantify its salience value at a specific location.

3.4 Spatial Expression Resolution & Results

Anytime a spatial utterance is processed, we try to carry out the underlying instruction at each point in the *RoI*. If this is possible, we then calculate the normalized metric values on those points. We thus have a kind of field potential whose intensity is modulated by the degrees of likelihood of each criterion after the particular instruction is executed at a given location. To select the correct placement,

we use multiple linear regression to model the relationship between these likelihoods and an expected response variable depending on the location by fitting a linear equations to the observed data. The model is defined by the k parameters $\beta_1..\beta_k$ of the system of linear equations:

$$Y_i(P) = \beta_0 + \beta_1 f_{i,1}(P) + \ldots + \beta_k f_{i,k}(P) \qquad (1)$$

Here k is the number of criteria, $f_{i,k}(P)$ the values (the independent variables) of the metrics applied at location P in the *RoI*, $Y_i(P)$ the expected goodness value (the dependent variable) at P, i an index running over the number of possible placements of the piece being manipulated and for each of the 5 units making up that piece. In our model, $Y_i(P)$ is set to 1 for all units P of the piece 1 if its manipulation can be found in our corpus of human interpretations, to 0 otherwise. Ultimately, the values β_j act as weighing coefficients for the metrics' values. We use equation (1) as a combined likelihood to gauge how close a spatial configuration is to the model of human interpretations. Specifically, we rank any location in the *RoI* according to the value obtained by summing up equation (1) over each point of the piece after this is operated upon.

We used half of the data for the determination of the model parameters and half for the evaluation. By taking the maximum value of the ranked list, the model interpreted spatial descriptions as humans did in our experiment in 61.4% of the expressions. Correct interpretations were ranked either second or third in 16.2% of the cases.

4 Discussion and Conclusion

We implemented a computational model that attempts to approximate human interpretation and judgment of situated language in the micro-world of a 2D puzzle game. We believe that the probabilistic nature of our method can be very useful in a dialogue system for spawning clarification requests or suggesting the location for a certain instruction.

Our system confirms that adopting an approach that considers several sources of information such as context, semantic and pragmatic evidence can be beneficial to the understanding of situated utterances (Gorniak & Roy, 2005). The metrics, now tailored for our restricted game domain, are extendible to other grid-like scenarios and spatially aware systems, even in 3D. The resolution of spatial relations is also portable to the case of one-to-many relations by applying our strategy between the one object and each one of those in the group.

We are expanding the system to include a few more metrics and dialogue capabilities between player and system, for error resolution and in contexts that need clarification to resolve ambiguities.

Acknowledgement. The EU Marie Curie grant #FP6-2002-Mobility-3014491 supported this work.

References

Brenner, M., Hawes, N., Kelleher, J., and Wyatt, J. 2007. *Mediating between Qualitative and Quantitative Representations for Task-Oriented Human-Robot Interaction.* Proceedings of the IJCAI.

Corradini, A., et al. 2007. *A Robust Spoken Language Architecture to Control a 2D Game.* Proc. of AAAI Int'l FLAIRS, pp. 199-204.

Costello, F., and Kelleher, J. 2006. *Spatial prepositions in context: The semantics of near in the presence of distractor objects.* Proceedings of the ACL-Sigsem Workshop on Prepositions.

Eschenbach, C. 1999. *Metric Details for Natural Language Spatial Relations.* ACM Transactions on Info. Systems, 16:(4):295–321.

Gorniak, P., and Roy, D. 2004. *Grounded Semantic Composition for Visual Scenes.* Journal of AI Research, Vol. 21, pp. 429-470.

Gorniak, P., and Roy, D. 2005. *Speaking with your Sidekick: Understanding Situated Speech in Computer Role Playing Games.* Proceedings of AI and Digital Entertainment.

Gorniak, P., Orkin, J., and Roy, D. 2006. *Speech, Space and Purpose: Situated Language Understanding in Computer Games.* Annual Meeting of Cogn. Science Society Workshop on Computer Games.

Kelleher, J., Costello, F., and van Genabith, J. 2005. *Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context.* Artificial Intelligence, Special volume on connecting language to the world, 167(1-2):62-102.

Kelleher, J., Kruijff, G.J., and Costello, F. 2006. *Proximity in Context: an empirically grounded computational model of proximity for processing topological spatial expressions.* Proc. of COLING.

Logan, D. and Sadler, D. 1996. *A Computational Analysis of the Apprehension of Spatial Relations.* Language and Space, MIT Press.

Moratz, R., Fischer, K., and Tenbrink, T. 2001. *Cognitive Modeling of Spatial Reference for Human-Robot Interaction.* International Journal on Artificial Intelligence Tools, 10(4): 589-611.

Regier, T., and Carlson, L. 2001. *Grounding spatial language in perception: An empirical and computational investigation.* Journal of Experimental Psychology: General, 130(2):273-298.

Roy, D. et al. 2002. *A Trainable Spoken Language Understanding System for Visual Object Selection.* Proceedings of the ICSLP.

Roy, D. 2002. *Learning Visually Grounded Words and Syntax for a Scene Description Task.* Computer Speech and Language.

Roy, D., and Mukherjee, N. 2005. *Towards situated speech understanding: Visual Context Priming of Language Models.* Computer Speech and Language, 19(2):227–248.

Tapus, A., et al. 2005. *Towards a multilevel cognitive probabilistic representation of space.* Proc. of the SPIE, Vol. 5666, pp. 39-48.

Winograd, T. 1971. *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language.* MIT AI Technical Report 235.

Hassan: A Virtual Human for Tactical Questioning

David Traum and **Antonio Roque** and **Anton Leuski**
Panayiotis Georgiou and **Jillian Gerten** and **Bilyana Martinovski**
Shrikanth Narayanan and **Susan Robinson** and **Ashish Vaswani**
University of Southern California
Los Angeles, CA USA
traum@ict.usc.edu

Abstract

We present Hassan, a virtual human who engages in Tactical Questioning dialogues. We describe the tactical questioning domain, the motivation for this character, the specific architecture and present brief examples and an evaluation.

1 Introduction

Virtual Humans can be useful for tutoring or training in a variety of interactive situations in which experiential learning can be beneficial, such as in (Traum et al., 2005a) and (Rickel et al., 2002). Virtual humans contain a number of components, including a virtual body, usually embedded in a virtual world, actions that the agent can perform, including movements and sound, cognitive capabilities to decide on which actions to do and updating internal state, and perceptual abilities for recognizing the actions of users and other things in the world.

In this paper we present Hassan, a virtual human for training in Tactical Questioning dialogues. We focus on the spoken dialogue components. A companion paper (Roque and Traum, 2007) describes the dialogue manager and emotion model more fully.

Currently there is no single "best practice" model for building virtual humans or especially their spoken dialogue components. While generally there are separate modules for speech recognition, natural language understanding, dialogue management, and output (e.g., Generation and Synthesis, or text selection and audio clip playing), there is no consensus on the best ways of engineering these modules. Part of the reason for this is that we are still fairly early in the search space, considering all of the possible techniques applied to the various domains that require spoken dialogue capability. Another issue is that there are several different goals for dialogue systems, and optimizing on one may lead to suboptimality for other goals. Some of these goals include: task success & efficiency, correct understanding & output, user satisfaction, believability/realism, authorability, reusability, revisability, and short development time.

Given the different relative importance of these goals and the specific features of the domain can lead to different choices for the spoken language technology components. For example, the virtual humans in (Rickel et al., 2002; Traum et al., 2005b) put a premium on depth of understanding within complex domains (teamwork, negotiation), but were somewhat narrow in the scope of what the virtual humans could talk about, and had a heavy authoring burden, requiring experts to create new domains. On the other hand, question-answering characters (Leuski et al., 2006) have a lower burden for depth, but must handle a broader range of questions and maintain believability and user satisfaction.

For our current endeavor, tactical questioning (see Section 2), we require capabilities between these two extremes. We need the authorability and general robustness of characters like SGT Blackwell (Leuski et al., 2006) but with more of the emotional and cognitive modeling of the situation from agents like Dr Perez (Traum et al., 2005b).

In this paper, we present Hassan, a Virtual Human for Tactical Questioning implemented using this in-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 71–74,
Antwerp, September 2007.

termediate architecture. In section 2, we describe the Tactical Questioning Domain and the Hassan scenario. In section 3, we describe the components of the system. In section 4, we describe the preliminary evaluation, and we conclude with future directions in Section 5.

2 Domain: Tactical Questioning

Tactical Questioning dialogues are those in which small-unit military personnel, usually on patrol, hold conversations with individuals to produce information of military value (Army, 2006). We are specifically interested in this domain when applied to civilians, when the process becomes more conversational and additional goals involve building rapport with the population and gathering general information about the area of operations. Hassan is a virtual human designed to act as a roleplayer and allow trainees to practice tactical questioning and get feedback from experienced instructors on their performance on several learning goals.

The scenario for Hassan takes place in contemporary Iraq. In a fictional storyline, the US authorities have built a marketplace as part of the reconstruction effort, but the local population continues to use the old, broken-down marketplace instead. It is the goal of the trainee to discover why. To do this, the trainee talks to Hassan, a local politician. If the trainee convinces Hassan to help him, the trainee will confirm that a tax has been levied on the new marketplace, and that the tax has been placed by Hassan's employer; if exceptionally successful, the trainee may even learn where that employer lives. If Hassan becomes adversarial, he may lie and tell the trainee that an American soldier is collecting the tax. Figure 1 shows the beginning of a typical dialogue with Hassan.

Trainee	Hello Hassan
Hassan	Hello
Trainee	How are you doing?
Hassan	Well, under the circumstances we are fine
Trainee	I'd like to talk about the marketplace
Hassan	I hope you do not expect me to tell you anything
Trainee	I just want to know why people aren't using the marketplace
Hassan	I don't feel like answering that question
Trainee	I think you know something about a tax
Hassan	I am simply doing business. It is rude of you to imply otherwise

Figure 1: Scenario Dialogue

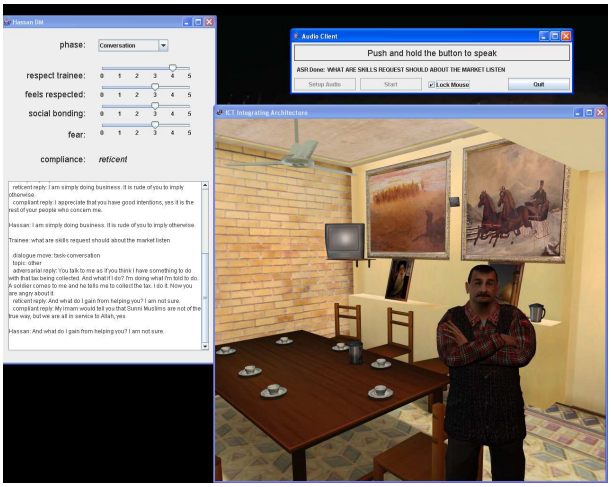


Figure 2: Hassan, a Virtual Human for Tactical Questioning, with some other components

3 Virtual Human Implementation

Figure 2 shows several components of Hassan during a session. The virtual environment includes the embodied character, which is the only component the trainee usually sees. Above that is a speech capture component showing the Automated Speech Recognition (ASR) results of an utterance. Also visible is a GUI showing the state of various of Hassan's emotional components. Although the system can run autonomously, its emotional state can also be modified at run-time by an instructor. The virtual environment is set in the Unreal Tournament game engine, similar to the agent in (Traum et al., 2005b). It also uses the *Smartbody* character controller (Thiebaux et al., 2007) to control the movements of the character, including lipsynch and nonverbal communicative behaviors, and the Nonverbal Behavior generator (Lee and Marsella, 2006) to select and synchronize non-verbal behaviors with the output text.

The language components include a speech recognizer, a set of statistical classifiers to recognize dialogue features and suggest responses, and a dialogue manager, to maintain a current cognitive and emotional model and chose the appropriate response. Our initial version of Hassan used the same architecture as SGT Blackwell, with a single clas-

sifier to pick the answer, and rudimentary dialogue manager to avoid repetition where possible and be able to answer further on the same topic. Our initial tests showed that this was inadequate for the tactical questioning domain, where one needs not just local coherence between questions and answers, but also an emotional progression of the character in which the kinds of questions and behavior early on in the conversation will effect the kinds of answers given later on. E.g., a trainee can increase or reduce fear.

In order to address this issue, we added a more sophisticated information-state based dialogue manager which can track several states that are important to deciding how compliant an agent should be. We also introduced a number of statistical classifiers (built using our NPCEditor software) to pick out important dialogue features as well as the best answer given a particular compliance level. Figure 3, shows the natural language components of our dialogue agent, including a set of NPCEditors working together with a rule-based Dialogue Manager. We discuss each of these components briefly below.

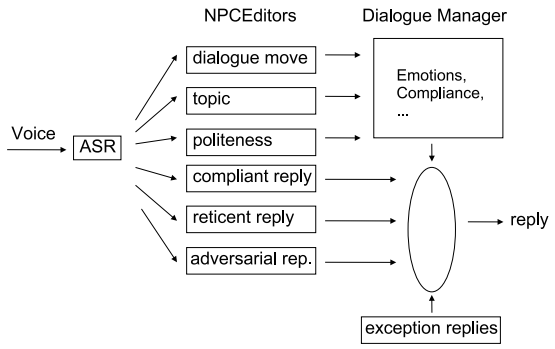


Figure 3: Architecture of Language Components

3.1 Automated Speech Recognition

The trainee talks to Hassan using a headset microphone and a push-to-talk button. The ASR component uses the Sonic statistical speech recognition engine (Pellom, 2001), with custom acoustic and language models (Sethy et al., 2005).

3.2 NPCEditor: Statistical Classification

Our NPCEditor tool allows one to build statistical classifiers for "non-player characters". It allows several output modes including email, chat, and several interprocess communication protocols. The classification can be between input and output text (e.g., the answer to a question), or between input text and output features (NLU) or input features and output text (NLG). It has been used in a variety of ways in our Virtual Human agents. The NPCEditor allows inputting and annotation of training data, training a classifier, and run-time performance all within the same software platform. The classification techniques and their use to select answers is described in (Leuski et al., 2006).

3.3 Dialogue Features

The NPCEditor statistical classifiers identify three utterance features of the user utterance: a dialogue move, a main topic and a level of politeness. The set of dialogue moves for the Tactical Questioning Domain are shown in Figure 4. The main topic is an aspect of significance for the domain and character. There are different topics for requests (e.g. marketplace, taxation), threats (e.g. loss of status) and offers (e.g. security, recognition, or secrecy). Politeness is one of *polite, neutral* or *impolite*. These three features work together to inform the decisions made by the dialogue manager.

Opening	greetings, introductions, ...
Complimentary	compliments, flattery, ...
General Conversation	non-task-related talk
Task Conversation	task-related talk
Threatening	threats
Offering	offers to provide something
Closing	moving to end the conversation

Figure 4: Dialogue Moves

3.4 Dialogue Manager

The dialogue manager of the system is based on the information-state approach (Traum and Larsson, 2003). It tracks a set of four information state variables relating to respect, bonding and fear, and calculates from these a current *compliance level* for the character. The utterance features from the classifiers are used to update these variables, which may result in a change in compliance level. A response is selected by choosing the response given by the classifier for that compliance level (or an exception reply for special circumstances). More about the dialogue manager and compliance computation can be found in (Roque and Traum, 2007).

4 Evaluation

A preliminary evaluation of the first version of this agent was held to produce data for analysis and to measure user satisfaction. Eight sessions were held with an equal combination of college-level military trainees, and information professionals in our research facility. Post-questionnaires allowed the trainees the opportunity to rate their experience.

Preliminary results indicate the users felt the system was off-topic too often to adequately judge the effects of the emotional components. In reply to ranking from 1 to 7 how satisfied they were with their questioning of the agent, the mean value given was 3.4. In reply to ranking from 1 to 7 how they rated Hassan as an interviewee, the mean value was also 3.4. A partial review of the logs indicates that these low scores may have been due to discrepancies in the reply authoring, which did not properly handle the generation of off-topic replies when confidence in an on-topic reply was low.

5 Future Work

While the current version of Hassan, with several information state variables, dialogue features, and 3 compliance levels is definitely an improvement in consistency over the previous version with one NPCEditor and no emotion-based information state, there is still much room for improvement. We are currently investigating techniques to track longer segments than the question-answer pair, as well as more sophisticated discourse processing on both the NLU and NLG side, while keeping the authoring relatively simple.

Acknowledgments

This work has been sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. We would like to thank Patrick Kenny, Stacy Marsella, Jina Lee, Andrew Marshall, and Aaron Hill for providing and assisting with the NVBGenerator, Smartbody animation controller, and graphical environment.

References

Department of the Army. 2006. Police intelligence operations. Technical Report FM 3-19.50, Department of the Army. Appendix D: Tactical Questioning.

Jina Lee and Stacy Marsella. 2006. Nonverbal behavior generator for embodied conversational agents. In Jonathan Gratch, Michael Young, Ruth Aylett, Daniel Ballin, and Patrick Olivier, editors, *IVA*, volume 4133 of *Lecture Notes in Computer Science*, pages 243–255. Springer.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, Sydney, Australia, July.

Bryan Pellom. 2001. Sonic: The university of colorado continuous speech recognizer. Technical Report TR-CSLR-2001-01, University of Colorado.

Jeff Rickel, Stacy Marsella, Jonathan Gratch, Randall Hill, David Traum, and Bill Swartout. 2002. Towards a new generation of virtual humans for interactive experiences. *IEEE Intelligent Systems*, pages 32–38, July/August.

Antonio Roque and David Traum. 2007. A model of compliance and emotion for potentially adversarial dialogue agents. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, september. this volume.

Abhinav Sethy, Panayiotis Georgiou, and Shrikanth Narayanan. 2005. Building topic specific language models from webdata using competitive models. In *Proceedings of Eurospeech*, Lisbon, Portugal.

Marcus Thiebaux, Andrew N. Marshall, Stacy Marsella, Edward Fast, Aaron Hill, Marcelo Kallmann, Patrick Kenny, and Jina Lee. 2007. Smartbody: Behavior realization for embodied conversational agents. In *7th International Conference on Intelligent Virtual Agents (IVA)*.

David Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In R. Smith and J. van Kuppevelt, editors, *Current and New Directions in Discourse and Dialogue*, pages 325–353. Kluwer, Dordrecht.

David Traum, William Swartout, Jonathan Gratch, Stacy Marsella, Patrick Kenny, Eduard Hovy, Shri Narayanan, Ed Fast, Bilyana Martinovski, Rahul Baghat, Susan Robinson, Andrew Marshall, Dagen Wang, Sudeep Gandhe, and Anton Leuski. 2005a. Dealing with doctors: A virtual human for non-team interaction. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, September 2-3.

David Traum, William Swartout, Stacy Marsella, and Jonathan Gratch. 2005b. Fight, flight, or negotiate: Believable strategies for conversing under crisis. In *5th International Conference on Interactive Virtual Agents*. Kos, Greece.

Identifying formal and functional zones in film reviews

Heike Bieler and **Stefanie Dipper** and **Manfred Stede**
Applied Computational Linguistics
University of Potsdam
Karl-Liebknecht-Str. 24–25
D-14476 Golm
Email: {bieler,dipper,stede}@ling.uni-potsdam.de

Abstract

We describe our system for breaking a film review (as an instance of a *semi-structured document*) into its formal and functional constituents. Based on a corpus study, we devised a set of 25 zone labels indicating the role that a unit can play within the review. We identify *formal* zones with a set of symbolic rules, while the distinction between *descriptive* and *evaluative* paragraphs is drawn with a statistical classifier. The approach achieves between 70 and 79% precision in recognizing the zones in our corpus.

1 Introduction[1]

Many text genres can be characterized as *semi-structured*: They do not display a completely conventionalized structure (as, e.g., many *weather reports* or *cooking recipes* do), but there nevertheless are some rules and tendencies that allow the reader to quickly recognize a document as an instance of the genre, and to isolate important portions. As a case in point, we are working with film reviews coming from various newspapers and web sites. While their overall structure is definitely not identical, there are similarities on what portions (henceforth *zones*) to expect, and in what order to expect them. Furthermore, in our corpus studies with English and German film reviews, we found a very clear correspondence between logical document structure (breakup in headers, lines, paragraphs) and content structure: Units playing a distinct functional role for the review are very likely to be separated in the logical structure as well. This lead us to the goal of automatically identifying the content structure of such documents. Our underlying application is automatic summarization: Identifying the zones of the film review is a prerequisite for ensuring that the summary contains information from all relevant zones (e.g., movie title, director, description of story, author's evaluation).

Following Stegert (1993), we distinguish between *formal* and *functional* elements of reviews, with the former being 'constituents' whose presence is characteristic for the genre, and the latter making contributions to the communicative goal of the author. The formal zones follow conventionalized patterns of shape and of linear order. They include the title, the name of the reviewer, list of cast, copyright notice, etc. As for the communicative goal of a film review, it is typically twofold: inform the reader about the contents of the film, and provide a subjective evaluation. The running-text paragraphs of a review belong to these two *functional* zones, and our initial corpus study had revealed that they are almost always confined to paragraphs: Authors very rarely mix description and opinion within a paragraph in their reviews. In the following, we discuss related work, then explain our approach to identifying formal zones, and finally turn to opinion classification.

Corpus The basis of our current implementation is a corpus consisting of 213 German film reviews from 7 different web sites. The reviews contain a total of 4,252 paragraphs, i.e., zones that we aim to identify.

[1]The research reported in this paper was funded by Bundesministerium für Bildung und Forschung, grant 03WKH22.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 75–78,
Antwerp, September 2007.

2 Related Work

The genre of film reviews has become relatively popular in computational linguistics, but the problem addressed is typically that of classifying an entire review as either positive or negative (e.g. Chaovalit and Zhou (2005)). Our work in effect takes a significant further step: We first break down the review into its various content zones, and then see opinion classification only as one subproblem, pertaining to a subset of the paragraphs.

The subtask of opinion identification has received much attention in recent years. Subjectivity in natural language encompasses a range of different phenomena, including the means to express opinions, emotions, or evaluations. Example applications are automatic classification of opinion texts (e.g. editorials) vs. factual texts (e.g. business texts or news) (Wiebe et al., 2004) or positive vs. negative ratings in reviews (Turney, 2002; Pang et al., 2002; Zhuang et al., 2006). The classification is applied to documents (e.g., Wiebe et al. (2004)) or sentences (Yu and Hatzivassiloglou, 2003).

In contrast to the above approaches, which are exclusively developed for English, we aim at learning subjectivity clues for German data. Moreover, in our classification task, paragraphs rather than documents or sentences are being classified.

Tag	Description
<audience-restriction>	Age restrictions for viewing (in the U.S.: MPAA rating)
<author>	Author of review
<author_place>	Author of review and source of publication
<author_rating>	Author of review and overall rating
<cast>	List of actors, possibly with their roles
<credits>	Credits (Producer, Camera, etc.)
<country_year>	Country and year of production
<date>	Date of review
<director>	Director of film
<format>	Technical format of film (16:9, 4:3, PAL, black/white, etc.)
<genre>	Genre of film (Comedy, Thriller, Documentary, etc.)
<language>	Language of film
<language-subtitles>	Language of subtitles
<legal-notice>	Copyright statement for review
<note>	Various meta-notes (e.g., review has been published earlier at different source)
<quote>	Quotation taken from film or other source
<rating>	Overall rating (5 stars, etc.)
<runtime>	Length of film
<show-loc_date>	Screening locations and dates
<structure>	Explicitly-structuring element, usually a single-word headline
<tagline>	Very short “grabbing” headline
<title>	Title of film
<DATA>	Mixed information, enumerated (credits, cast, etc.)
<dvd-DATA>	DVD release information

Table 1: Tag set for formal zones

3 Formal zones

The inventory of formal zones we determined in the corpus study is shown in Table 1. Recall that we are tagging zones paragraph-wise, which is warranted by the aforementioned relatively clean layout–function correspondence in the genre; at the same time, this decision leads to the occasional need for zones that combine different information. We thus found that `author` is often given together with the `place` of publication, and often with his or her overall `rating` for the film. The other frequent case of “mixing” information are enumerations of cast and contributors (credits); for these, we use the tag `DATA`, which also has a variant for DVD-related information (see bottom of the table).

Our corpus for evaluation (see below) contains a total of 1,156 zones. Zones that occur most often are `DATA` (which make up 18% of all zones), `title` (16%) and `structure` (15%). The zones that are highly relevant for text summarization certainly include the `title` zone, but also zones that are considerably less frequent, like `director` (3%), `rating` (0.4%) or `author_rating` (1%).

3.1 Identifying formal zones

After hand-annotating portions of our corpus, we inspected the various instances of the formal zones and found that they display striking formal characteristics that can quite well be captured in regular expressions. A very simple case is `legal-notice`, which invariably contains the copyright symbol or the word itself. Less simple yet tractable is a zone like `author`, since person names can be recognized by the number of words, capital letters, optional middle initials. Also, information about the position of the text span plays an important role here: the author is always given toward the beginning or

the end of the text. The same holds for `title`, which in addition regularly occurs in neighbourhood to `author` (but the order can vary). What we are *not* exploiting for the time being is layout information such as HTML tags of the original documents. Instead, we convert all input to plain text, and thus our approach operates in the same way for both internet and newspaper material.

Given the observations on regularities in the formal zones, we decided to follow a symbolic approach for them, i.e., we wrote recognition rules encoding features like the ones just mentioned. As a convenient tool for this purpose, we used LAPIS (Miller, 2002), a toolbox for "lightweight text processing". The data set for developing these rules (i.e., for first taking inspiration and then fine-tuning the rules), consisted of 101 film reviews. The evaluation was then performed on a set of 112 unseen reviews.

3.2 Evaluation

The symbolic rules perform excellently on the zones `rating`, `author_rating`, `audience-restriction` and `format` (all with 100% precision and 100% recall). Results for other zones relevant for summarization are: `title` (P: 61%, R: 65%), `director` (P: 42%, R: 78%). Average performance of the rules is 70% precision and 63% recall.

An error analysis of the automatic `title` zone classifications reveals that zones that erroneously get classified as `title` are `DATA` (33% of the misclassifications), `tagline` (25%), and `structure` (17%). On the other hand, `title` is often misclassified as `tagline` (53%) or `director` (15% — this happens with 2-words film titles like *Brokeback Mountain*). Very often, indeed, none of the rules matched a `title` zone, and the rules did not come up with a classification at all (28%). To overcome such problems, we are currently adding a post-processing step that reconsiders all the tag assignments in the light of the overall situation — in this step we can use non-local information like the corpus observations that `author` or `title` (as a single text span) appears at least once in the document but no more than twice (see Section 5).

4 Functional zones

Functional zones are paragraphs with free text. We distinguish two main types of functional zones: descriptive zones (`describe`) and comment zones (`comment`). Descriptions are paragraphs that describe the story, different aspects or peculiarities of the film, without commenting about it. They therefore can be considered as 'objective' information. In contrast, comment zones are paragraphs that contain expressions of opinions by the author, i.e., 'subjective' information. In our application (text summarization), it is very important to be able to reliably distinguish between the two types.In our data, there are slightly more `comment` paragraphs (54%) than `describe` paragraphs (46%).

4.1 Identifying functional zones

Feature set For classifying the functional zones, we used as training features a bag-of-"words" approach. In a detailed evaluation of tf*idf measures used as relevance weights, we found that 5-grams perform best for German data, so our bag of "words" consists of weighted character 5-grams. All 5-grams occurring in the paragraph that is to be classified are weighted according to the $tf * idf$ measure, where tf is the frequency of the 5-gram in the paragraph, and idf is the inverse document (i.e., paragraph) frequency according to a reference corpus: a large collection of internet film reviews.

Training procedure Pang et al. (2002) compare different machine learning methods and achieved accuracies between 72.8% and 82.9%, depending on the training features and the method. In their evaluation, Support Vector Machines (SVM) perform best for many of the feature combinations.

In our approach, we also use SVM. Our feature sets, however, do not consist of words or POS tags but 5-grams. We used the tool SVMLight (Joachims, 1999) and performed a threefold-crossvalidation on the 213 reviews, which contain 1,159 functional zones..

4.2 Evaluation

The table below presents the results from the functional zone classification. Overall accuracy is quite satisfactory, at 79.34%. `Comment` zones are classified more successfully than `describe` zones.

Zone type	Precision	Recall	Accuracy
`comment`	81.60%	79.69%	79.34%
`describe`	76.83%	78.94%	

5 Conclusion and outlook

For many applications, including summarization, but also question–answering and others, the range of portions and their relative relevance for the application heavily depends on the *genre*. For the example discussed here, film reviews, it is evident that information about the *content structure* of a document can be of immense help for creating a balanced summary, for choosing zones in which the answer to a question is sought, etc.

Based on a corpus study, we have developed an inventory of zone labels for the genre *film review* and implemented a system for automatically identifying these zones, i.e., for breaking up a document into its content structure. The precision currently ranges from 70% for formal zones to 79% for the two functional zones. Our approach is hybrid: it utilizes both symbolic rules and a statistical classifier. The overall algorithm first decides heuristically whether to invoke the symbolic rules or the classifier (the functional zones are longer-text paragraphs that occur in the middle of the document and are not interrupted by formal zones), and then each paragraph of the document receives its label by either module. Recognition is based on merely local information so far.

Our current work aims at improving the results by taking two different routes. For one thing, we are integrating layout information, in particular HTML tags, into the identification of formal zones. To that end, the input to the system will no longer be plain text but a canonical, XML-based representation of the logical document structure, which is produced from HTML. The other line is to make more extensive use of knowledge about zone neighbourhood. To this end, we are revising the rules for formal zones so that they output probabilistic judgements, and these will be combined with a trigram model capturing the zone sequences in our corpus. Thus, all information about zone locations will be removed from the rules and incorporated into a single, separate knowledge source. Finally, we are currently adapting our implemented text summarizer (Stede et al., 2006) to utilize the zone information so that the quality of summaries for the particular genre of film reviews will be improved considerably.

References

P. Chaovalit and L. Zhou. 2005. Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proc. of the 38th Hawaii Int'l Conference on System Sciences.*

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.

Robert C. Miller. 2002. *Lightweight Structure in Text*. Ph.D. thesis, Computer Science Department, School of Computer Science, Carnegie Mellon University.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proc. of EMNLP-02*, pages 79–86.

M. Stede, H. Bieler, S. Dipper, and A. Suryiawongkul. 2006. Summar: Combining linguistics and statistics for text summarization. In *Proc. of ECAI-06*, Riva del Garda.

G. Stegert. 1993. *Filme rezensieren in Presse, Radio und Fernsehen*. München: TR-Verlagsunion.

Peter D. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proc. of the ACL-02*, pages 417–424.

Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. 2004. Learning subjective language. *Computational Linguistics*, 30(3):277–308.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proc. of EMNLP-03*.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proc. of the 15th ACM international conference on Information and knowledge management*.

CHAT To Your Destination

Fuliang Weng[1] Baoshi Yan[1] Zhe Feng[1] Florin Ratiu[2] Madhuri Raya[1] Brian Lathrop[3] Annie Lien[1] Sebastian Varges[2] Rohit Mishra[3] Feng Lin[1] Matthew Purver[2] Harry Bratt[4] Yao Meng[1] Stanley Peters[2] Tobias Scheideck[1] Badri Raghunathan[1] Zhaoxia Zhang[3]

[1]Research and Technology Center, Robert Bosch LLC, Palo Alto, California, USA
[2]Center for the Study of Language and Information, Stanford University, California, USA
[3]Electronics Research Lab, Volkswagen of America, Palo Alto, California, USA
[4]Speech Technology and Research Lab, SRI International, Menlo Park, California, USA

{fuliang.weng,baoshi.yan,zhe.feng,madhuri.raya}@us.bosch.com
{varges,fratiu,mpurver,peters}@csli.Stanford.edu
{brian.lathrop,rohit.mishra}@vw.com
{harry}@speech.sri.com

Abstract

In the past few years, we have been developing a robust, wide-coverage, and cognitive load-sensitive spoken dialog interface, CHAT (Conversational Helper for Automotive Tasks). New progress has been made to address issues related to dynamic and attention-demanding environments, such as driving. Specifically, we try to address imperfect input and imperfect memory issues through robust understanding, knowledge-based interpretation, flexible dialog management, sensible information communication, and user-adaptive responses. In addition to the MP3 player and restaurant finder applications reported in previous publications, a third domain, navigation, has been developed, where one has to deal with dynamic information, domain switch, and error recovery. Evaluation in the new domain has shown a good degree of success: including high task completion rate, dialog efficiency, and improved user experience.

1 Introduction

In the past few years, we have been developing a robust, wide-coverage, and cognitive load-sensitive spoken dialog interface CHAT under a joint NIST ATP project with Bosch RTC, CSLI of Stanford University, ERL of VW of America, and STAR lab of SRI International. The CHAT system is specifically designed to address imperfect speech and imperfect memory of human users, when they use the system to interact with devices and receive services while performing other tasks—typically, these tasks are their primary, and sometimes even critical tasks, such as driving.

Examples of imperfect speech are speech disfluencies, incomplete references to proper names, and phrase fragments, while examples of imperfect memory include very limited number of names memorized or non-exact names memorized. Imperfect speech and memory happen quite often. In one reported Wizard-Of-Oz experiment for the restaurant finder domain [Weng et al 2006], 29% of the proper names used by people were partial names. The imperfect speech and memory issues accompanied with multi-tasking pose a big challenge to the development of a robust dialog system. Over the course of the project, we have developed a number of technologies in various modules of the dialog system to deal with these two issues [Weng et al 2004; Zhang and Weng 2005; Mirkovic and Cavedon 2005; Pon-Barry et al 2006; Varges 2005; Purver et al 2006]. Specifically, in this paper, we describe progress made over the past year when a navigation domain and related use cases are introduced. Evaluation conducted for the navigation domain shows high task completion rates and user satisfaction.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 79–86,
Antwerp, September 2007.

The paper is organized as follows: Section 2 describes the updated CHAT system architecture and its functionality; Section 3 is devoted to approaches used to address the imperfect speech and memory issues; Section 4 gives a description of data collection setup, evaluation scenarios, as well as evaluation results; finally, we conclude with a comparison with other work.

2 The CHAT System and Its Functionality

The CHAT system has adopted many state-of-art technologies and has grown beyond its heritages over the years. This progress is reflected in several core aspects, including the spoken language understanding (SLU) module, the dialog manager (DM), the content optimizer (CO), the knowledge management (KM), the response generation (RG), as well as the overall system architecture.

The SLU module integrates multiple understanding strategies with components such as edit region detection algorithm [Zhang and Weng, 2005; Zhang et al 2006][1], partial name identifier, shallow semantic parser, and deep structural parser. This approach enables understanding at finer levels when faced with imperfect input from the distracted multi-tasking user, and/or from speech recognition errors.

The DM, originated from the CSLI dialog manager [Lemon et al 2002], follows the information-state-update approach [Larsson and Traum 2000]. It uses a dialog move tree to keep track of multiple dialog threads and multiple applications [Mirkovic and Cavedon 2005; Purver et al 2006]. The latest version also supports mixed initiative dialogs for all the three domains.

The KM controls access to knowledge base sources and their updates. Domain knowledge is structured according to domain-dependent ontologies. The current KM makes use of OWL, a W3C standard, to represent the ontological relationships between domain entities.

The CO module acts as an intermediary between the dialog management module and the knowledge management module, controls the amount of content, and provides recommendation to users. It receives queries from the DM, resolves possible ambiguities, and queries the KM. It performs an appropriate optimization strategy based on the returned results [Pon-Barry et al 2006].

The RG module uses a hybrid rule-based and statistical approach. It takes query results from the KM via CO and generates natural language sentences as system responses to user utterances. The query results are converted into natural language sentences using a rule-based bottom-up production system. Finally, a scoring and ranking algorithm is used to select the best generated sentence [Varges 2005].

The architecture of the CHAT system is similar to its previous versions [Weng et al 2004; Weng et al 2006]. However, a couple of enhancements have been made to deal with multiple applications and random events from external devices or services. One enhancement is the introduction of an Application Manager (AP). The AP module isolates the application dependent information and operations from the core dialog system.

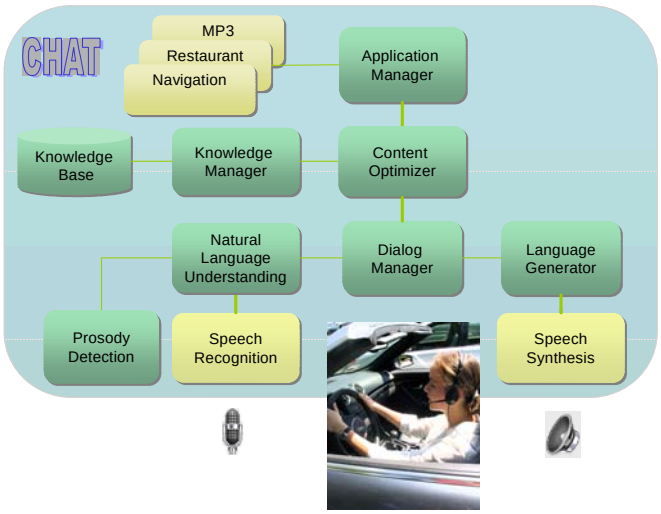


Figure 1 The CHAT System architecture.

Another major improvement is the modularity and configurability. The current version of the CHAT system is highly modularized and configurable. All the modules in Figure 1 are shared across the different domains. Domain specific models or parameters are supplied to the system in a configurable manner. Explicit on-the-fly domain switch becomes very simple – people can just say "switch to X" or other commonly used phrases to switch to the domain X. Implicit domain switch is also possible, where the users do not have to use explicit

[1] Edit region detection algorithms identify disfluent areas in an input utterance, such as hesitation, repeat, or correction. For example, "Get a, hmm, take me to Dave's house".

statements for switching to another domain. For example, having selected a desired restaurant in the restaurant domain, the user may then say “find me a fast route to restaurant XYZ”, without preceding this request with an explicit statement such as “switch to navigation”. However, due to extra burden on the system when all the applications are included, this feature is not set as a default. Additionally, because of the high modularity and configurability, it is much easier to add any new application.

In the added navigation domain, a few new functionalities are provided in the CHAT system:

1. Destination entry. One may either tell the system his or her address or use a selected Point Of Interest (POI) as the destination. In address input mode, different expressions for addresses, even ones with partial street or city names, are allowed.
2. Route selection. One may negotiate with the system with different criteria during the conversation for route selection. The criteria include distance, speed, mid-point, avoidance, or highway preference. For example, the driver can tell the system to find the shortest route to a destination that avoids a specific street. Or, he or she can also ask the system to find the fastest route that goes by a specific location and avoids highway.
3. Dynamic information or random events. While driving in a driving simulator, one can access dynamic information, such as the speed limit and the distance to destination. In addition, the dialog system can process random events, such as instructions from the navigation unit, and communicate the events to the user. In the meantime, the user can ask for a repeat of information if he or she did not hear the message clearly.

The three selected applications for the CHAT system, i.e., MP3 music player, restaurant finder, and navigation, represent three important application categories: entertainment, service, and navigation. When combining the three application domains together, the CHAT system can demonstrate a very useful scenario, which allows people to negotiate a POI (restaurant) or explicitly enter a destination; find the best route; and, on the way to the destination, access dynamic location-based information (“what is my next turn”) or entertain self (“play a song in playlist X”).

To give a concrete impression of what the CHAT system can do, an example dialog is given here:

User: Switch to restaurant.

System: Ok. Switched to the restaurant finder.

User: Can you find me a Chinese restaurant in Palo Alto with good service?

System: There are 11 Chinese restaurants in Palo Alto that have good service. Here are the first few: Little Garden Restaurant China Delight Mandarin Gourmet Restaurant Should I continue?

User: Hmm...tell me more about, Garden Restaurant?

System: Little Garden Restaurant is an expensive Chinese restaurant. Dress code is casual, service is good and rating is three stars. It’s on El Camino Real in Palo Alto.

User: Switch to navigation.

System: Ok, switched to Navigation system.

User: Take me to Little Garden Restaurant, but stop by Bosch Page Mill Office.

System: Ok, I will plan a route to Little Garden Restaurant that goes by Bosch Page Mill Office. Would you like anything else?

User: Get me the fastest route.

System: Ok, I will plan a fast route to Little Garden Restaurant that goes by Bosch Page Mill Office. Would you like anything else?

User: Um...nope.

System: Ok, I will plan a route now. Please wait … I calculated a route to Little Garden Restaurant that goes by Bosch Page Mill Office. It is 4.3 miles long and should take around 12 minutes. Now I am starting guidance...

User: Switch to music player.

System: Ok. Switched to mp3 music player.

User: What songs do you have by Alicia Keys?

System: There are 16 songs by Alicia Keys. Here are the first few: GoodBye, Jane Doe, and Caged Bird

User: Play “Caged Bird.”

System: Playing the song “Caged Bird.”

User: Add it to a playlist.

System: Which playlist do you mean, car or work?

User: What’s on the car playlist?

System: Interlude, A Womans Worth, and Whatever.

User: Add it to the car playlist then.

System: I added the song "Caged Bird" to car.

User: Switch to navigation.

System: Ok. Switched to navigation system.

User: Where am I right now?

System: You are on Page Mill Road in Palo Alto.

User: How far away am I from my destination?

System: You need to travel 2.5 miles to get to destination Little Garden.

During the conversation, click-to-talk is used to start the recognition. To reduce the effect of early speaking or early cutoff, we use a circular buffer to locate the start of an utterance, and use prosody information to identify precisely the ending of an utterance [Shriberg et al 2000]. This mechanism is integrated with the Nuance V8.5 recognizer.

In the next section, we will discuss the additional improvements made to address the issues of imperfect speech and memory.

3 Dealing with Imperfect Input and Memory

Two threads of research have been explored to deal with imperfect input: improve the robustness in the concerned modules; and provide error recovery strategies.

Improving robustness. To accommodate partial names in human utterances, separate ngram name models are trained on name databases of different classes for the SR module. A disfluency model is separately trained and integrated in the Statistical Language Model (SLM) for the recognizer. The partial or full proper names and disfluent regions are then identified by a proper name identifier and edit region detector, respectively. To understand the output from the recognizer, its SLU module adopts multi-component understanding strategies. A deep understanding component provides detailed information for each component in an utterance, which may be used for sophisticated dialogs. This module may also provide the boundary information for unknown proper names. On the other hand, a shallow semantic parser extracts domain-specific information, including flat or structured semantic classes. This provides a backoff strategy in the case the deep understanding module does not produce valid parses. These two components complement each other for better understanding and conversation.

Error recovery strategies. Individual understanding strategies do not always produce the correct interpretation in their 1st candidate. To correct errors, similarly, we experiment and integrate two different approaches: delay the final decision to a late stage; and design dialog strategies to clarify or confirm user's intention. In the first approach, the SLU passes the top n-best alternatives as well as their likelihood scores to DM. The DM makes the final decision based on the n-best output from the SLU module, the possible dialog moves, and the dialog context (active dialog threads) [Purver et al 2006]. To deal with possible misunderstanding, we also developed dialog strategies such as clarification, confirmation, or even rejection when the system is not confident about its understanding. Another way to improve the communication is to convey back implicitly or explicitly the interpreted results and allow user to revise his or her constraint specification when any mismatch is noticed. Revision and addition of constraints onto previously stated ones are realized across all the three domains.

To handle imperfect memory issue, we continue our research in two directions: regulate the amount of information through presentation strategies; and allow the users to ask for the repeat of information already presented.

Regulated information presentation. During the conversation, user utterances are interpreted, and internal queries are constructed based on the constraints extracted from the utterances. These queries are sent to the Content Optimizer and Knowledge Manager for obtaining results that satisfy the constraints. Quite often, the results and their quantity would either overwhelm the user or leave them in a position where he or she does not know how to proceed. This can be a serious distraction or cognitive load problem in our investigation, as the user is occupied by other critical tasks, such as driving. One consequence is that people may not remember all the items enumerated, when the returned result list is long. In such case, the system proposes additional criteria so as to narrow down the results. In the event there is no result from the databases, the system proposes a relaxation of the constraints from the user. This has led to better user satisfaction [Pon-Barry et al 2006].

Information repetition. When the user focuses on other critical tasks, it is not always easy for him or her to remember the statements from the system.

One additional functionality allows the user to ask for the repeat of information just presented. This new functionality is very useful especially in the navigation domain where the navigation instructions occur at random and people may not always pay attention to the instructions at the time of speaking.

In addition, as mentioned earlier, the CHAT system allows the user to use partial names, anaphora, or ordinal references[2], which alleviates the imperfect memory issue and reduces the cognitive load of the user.

After the CHAT system is equipped with the above approaches and strategies, it shows a great improvement in terms of dealing with various phenomena caused by imperfect input and imperfect memory. Since most of these approaches and strategies are very collaborative in nature, they lead to a positive effect on user experience. This is partially reflected in the evaluation results reported in Section 4.

4 Experiments and Evaluation Results

For the navigation domain, the experimental setup is to drive and talk in a driving simulator. Three virtual cities are designed in the simulated environment with different streets, buildings, and businesses. Approximately 50 streets are setup in the tri city virtual environment – a limited number due to the cost of street design in the virtual world. Five different routes are designated to control the experiments and about 2500 restaurant names are included in the database for POI queries. Each restaurant is associated with a street name, a street number, and a city name. There is some duplication between city names and street names in the environment. Conducting experiments in a simulated environment addresses bias concern that arises when real cities are used for the task—some subjects may be more familiar than others in terms of streets and navigation. Using simulated environments also enables us to control the variation of different factors in the experiments, such as traffic.

As in the other two domains, WOZ data collection was used to bootstrap the development of the CHAT system for the navigation domain [Cheng et al 2004]. For the WOZ data collection, 20 subjects were recruited for performing navigation related tasks while driving in the three cities in the driving simulator. In addition, 14 subjects were recruited for dry runs, and 20 subjects were used for evaluation. The scenarios used in dryruns and evaluation are a subset of the scenarios used in the WOZ data collection.

The WOZ data collection gives us insight into how human subjects interact with an ideal dialog system, helps us in selecting research topics we need to address, and provides us data for improving the language coverage in both NLU and NLG modules.

Since the CHAT dialog system is designed as a task-oriented system and is not intended for any general conversation, careful attention was given to the development of the dialog tasks for the subjects to perform in the WOZ data collection, dry runs, and evaluation. Specifically, we developed the following two guidelines:

1. **Task-constrained**. We try to make goals of each task transparent and explicit (to form the intended mental context), so that the collected speech would not become irrelevant, unusable, or very sparse (see an example below).
2. **Language-neutral**. The language used in the instructions for communicating these task goals to the participant and in the scenario descriptions was created in such a way to avoid "copying behavior". One instruction explicitly asks the participants to "try to phrase your requests in your own words, rather than simply repeating the description of the scenarios".

We call this task design approach as *task-constrained and language-neutral*. This approach is used for both the restaurant finder and navigation domains. An example of a task description from the navigation is given here.

Task description: You have just picked up your business clients from the airport and would like to take them out to a reasonably priced lunch. You think that they would prefer Chinese food. Use the Navigation System to (1) find a Chinese restaurant, and (2) plan a route to the restaurant.

Eight task categories are used in the evaluation with examples such as "plan routes to destinations (e.g., restaurant POIs or address input)" and "query about road conditions". Each subject is given a practice trial and three test trials. The purpose of

[2] Examples of the ordinal references include "the second one", or "that last one".

the practice trial is to familiarize the subjects with the procedure and tasks, and to reinforce the language-neutral guideline. A total of 16 tasks from the eight task categories are designed, and they are designated to the three test trials. The evaluation procedure is very similar to the one used for the restaurant finder domain [Weng et al 2006].

Initial comparison of expressions used in the navigation scenario/task descriptions and expressions used by the subjects shows that the copying behavior is largely avoided. We found that only 18.13% of the subject expressions mimic the scenario/task expressions. In quantifying the copy behavior, it is counted as a copy if an expression is used in a task description and a subject repeats this same expression. For example, in the task "get clarification of the most recent route instruction", if the subject says "clarify the most recent instruction", this is counted as a complete copy; if the subject says "clarify the last instruction", this is counted as half of a copy; and if the subject says "repeat the last instruction", this is counted as a non-copy. Certain expressions do not have a clear alternative, such as "the current location". In these cases, we do not count them as a copy, and there are only two of such expressions.

This initial result indicates that our guidelines are effective in the experiments.

Among other metrics, three major measurements are used in the evaluation of CHAT's performance for the navigation tasks: task completion rate, dialog efficiency, and user satisfaction. The task completion rate is defined as the percentage of tasks completed during the evaluation. The CHAT system reaches an overall 98% task completion rate for the navigation tasks. To measure the dialog efficiency, we use the number of turns required to complete a task. Here, one turn was defined as one user utterance to the system during a dialog exchange between the user and the system while attempting to perform a task. The CHAT system is able to complete the tasks with 2.3 turns on average. Although it is not directly comparable between the two different domains, this number is much smaller than the average number of turns needed for the restaurant finder tasks (4.1 turns) reported one year earlier. Using the user satisfaction rating system by CU-Communicator [Pellom et al 2000], we reached a score of 1.98 with 1 indicating "strong agreement" and 5 indicating "strong disagreement" to each of the following statements:

4. It was easy to get the information I wanted.
5. I found it easy to understand what the system said.
6. I knew what I could say or do at each point in the dialog.
7. The system worked the way I expected it to.
8. I would use this system regularly.

We computed a one-sample 2-tailed t-test to see if mean ratings for the navigation system was significantly different from the mean rating of 1.76 for the best of the CU Communicator Systems (i.e., goal user satisfaction rating). Results showed that this difference was not significant ($t(19) = 1.17$, $p > .05$). This suggests that participants were no less satisfied with our navigation system than those participants who evaluated the CU Communicator System.

To get a better understanding of the improvement, we examine the word recognition accuracy for the two domains: for the navigation tasks, the accuracies with and without Out-Of-Vocabularies (OOVs) included are 85.5% and 86.5%, respectively; for the restaurant finder tasks, the accuracies are 85% and 86%, accordingly. Thus, the improvements are more likely a result of the new or refined implemented approaches.

5 Conclusions

Previous dialog applications include travel planning, flight information, conference information, bus information, navigation, hotel reservation, and restaurant finder [Pellom et al 2000; Polifroni et al 2003; Bohus et al 2007]. However, these applications are independently developed using single or completely different frameworks. In our case, we have integrated three representative applications and allow explicit or implicit domain switch with shared dialog contexts. The most related work is the GALAXY-II [Seneff et al 1999]. However, in their work, different applications are managed by different turn managers.

In terms of content presentation, [Polifroni et al 2003] discussed ways of organizing the content based on fully automated bottom-up clustering, while our approach focuses on semi-automated but configurable strategies that make use of the system ontology, and on external domain configurations for content organization and presentation.

More sophisticated dialog management research has recently focused on collaborative aspects of human machine dialogs [Allen et al 2001; Lemon et al 2002; Rudnicky et al 1999]. However, such research on conversational dialog systems has typically focused on dealing with dialogs that users need to pay full attention to. In addition, most of this research only deals with simple expressions where the meanings are mainly embedded in the semantic slots. For research in which elaborated expressions are considered, the coverage is typically small. Another thread of research is targeted at broad coverage but simple dialogs, which is exemplified by the work at AT&T [Gorin et al 1997].

While extending the research on the collaborative aspects, our effort specifically focuses on dealing with the conversational phenomena in multi-tasking and distracting environments, specifically imperfect input and imperfect memory. While dealing with imperfect input can be traced back far in time [Carbonell and Hayes, 1983; Weng 1993; Lavie & Tomita 1993; He and Young 2003], the CHAT system integrates models ranging from disfluency, partial and full proper names, shallow semantic parsing, and deep structural parsing. The interpretation only occurs when all the contextual information and alternatives are gathered. For the imperfect memory issue, we explore information presentation and other strategies to enable the user to access the information comfortably. All these approaches and strategies lead to high task completion rate and dialog efficiency as well as user satisfaction across the three domains, especially for the navigation. Collectively, the CHAT system shows very interesting use scenarios and promising performance.

Acknowledgement

This work is sponsored by a NIST ATP funding, as well as Robert Bosch LLC and VW of America.

References

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. Towards Conversational Human-Computer Interaction. *AI Magazine*, 2001.

Bohus, D., Raux, A., Harris, T., Eskenazi, M., and Rudnicky, A. Olympus: an open-source framework for conversational spoken language interface research, *HLT-NAACL 2007 workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology*, Rochester, NY, 2007.

Carbonell, J. and Hayes, P. Recovery Strategies in Parsing Extragrammatical Language. *American Journal of Computational Linguistics.* Vol 9, No. 3-4, 1983.

Cheng, H., Bratt, H., Mishra, R., Shriberg, E., Upson, S., Chen, J., Weng, F., Peters, S., Cavedon, L., and Niekrasz J. A Wizard-of-Oz framework for collecting spoken human computer dialogs. *Proc. of ICSLP-200*0, Jeju, Korea, 2004.

Gorin, A., Riccardi G., Wright. J., How may I help you? *Speech Communication*, Vol. 23, pp. 113-127, 1997.

He, Y. and S. Young. A data-driven spoken language understanding system. *IEEE Workshop on Automatic Speech Recognition and Understanding*. 2003.

Larsson, S. and Traum, D. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering,* 6(3-4), 2000.

Lavie, A., and Tomita, M. GLR* - An Efficient Noise-Skipping Parsing Algorithm for Context-Free Grammars. In Proceedings of IWPT-1993, Tilburg, The Netherlands, August 1993.

Lemon, O., Gruenstein, A., and S. Peters. Collaborative activities and multi-tasking in dialogue systems. *Traitement Automatique des Langues (TAL),* 43(2), 2002.

Mirkovic D. and Cavedon L., Practical multi-domain, multi-device dialogue management", *PACLING'05: 6th Meeting of the Pacific Association for Computational Linguistics*, Tokyo, 2005.

Pellom, B., Ward, W., and S., Pradhan. The CU Communicator: An architecture of dialog systems. *Proc. of ICSLP*, Beijing, 2000.

Polifroni, J., Chung, G., and Seneff, S. Towards automatic generation of mixed-initiative dialog systems from web content. *Proc. of Eurospeech*, 2003.

Pon-Barry, H. and F. Weng. Evaluation of presentation strategies in a conversational dialog system. *Proc. of ICSLP*, 2006.

Purver, M., Ratiu, F., and Cavedon, L. Robust Interpretation in Dialogue by Combining Confidence Scores with Contextual Features. *Proc. of Interspeech*, Pittsburgh, PA, 2006.

Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu, W., and Oh, A. Creating natural dialogs in the Carnegie Mellon Communicator system. *Proc. of Eurospeech*, 1999.

Seneff, S., Lau R., and Polifroni, J. Organization, communication, and control in the GALAXY-II conversational system . *Proc. of Eurospeech*, 1999.

Shriberg E., Stolcke A., Hakkani-Tur D. and Tur G. Prosody-Based Automatic Segmentation of Speech into Sentences and Topics, *Speech Communication* 32(1-2), 127-154, 2000.

Varges, S. Chart generation using production system. *Proc. of 10th European Workshop on Natural Language Generation*, 2005.

Weng, F. Handling Syntactic Extra-grammaticality. *Proceedings of the 3rd International Workshop on Parsing Technologies*, Tilburg, the Netherlands, August 10-13, 1993.

Weng, F., Cavedon, L., Raghunathan, B. Mirkovic, D., Cheng, H., Schmidt, H., Bratt, H., Mishra, R., Peters, S., Upson, S., Shriberg, E., Bergmann, C., Zhao, L. A conversational dialogue system for cognitively overloaded users. *Proc. of ICSLP*, Jeju, Korea, 2004.

Weng, F., Varges S., Raghunathan, B. Ratiu, F., Pon-Barry, H., Lathrop, B., Zhang, Q., Bratt, H., Scheideck, T., Mishra, R., Xu, K., Purvey, M., Lien, A., Raya, M., Peters, S., Meng Y., Russell, J., Cavedon, L., Shriberg, E., and Schmidt, H. CHAT: A Conversational Helper for Automotive Tasks. *Proc. of Interspeech*, Pittsburgh, PA, 2006.

Zhang, Q. and Weng, F. Exploring features for identifying edited regions in disfluent sentences. *Proc. of 9th IWPT*, Vancouver, Canada, 2005.

Zhang, Q., Weng, F and Feng, Z. A progressive feature selection algorithm for ultra large feature spaces. *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL9th*, Sydney, Australia, 2006.

Commute UX: Telephone Dialog System for Location-based Services

Ivan Tashev, Michael L. Seltzer, Yun-Cheng Ju, Dong Yu, Alex Acero
Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
{ivantash, mseltzer, yuncj, dongyu, alexac}@microsoft.com

Abstract

In this paper, we describe a telephone dialog system for location-based services. In such systems, the effectiveness with which both the user can input location information to the system and the system delivers location information to the user is critical. We describe strategies for both of these issues in the context of a dialog system for real-time information about traffic, gas prices, and weather. The strategies employed by our system were evaluated through user studies and a system employing the best strategies was deployed. The system is evaluated through an analysis of 700 calls over a two month period.

1 Introduction

The availability of online maps and mapping software has led to a dramatic increase in location-based services, such as route planning, navigation, and locating nearby businesses, e.g. (Gruenstein, et al., 2006). While much of the effort has been focused on bringing these applications and services to desktop computer users, there is a demand for these services to be available to mobile users.

A significant portion of the mobile users will utilize these services from a vehicle while driving. The automotive environment is a particularly challenging, because operating a vehicle is a hands-busy and eyes-busy task, making the use of touch screens or pointing devices potentially unsafe. In contrast, using of speech as both an input and output modality is a natural and safe means of interacting with information (Tsimhoni, et al., 2004).

In this paper, we present a telephone dialog system called Commute UX (Commute User Experience) that provides location-based information to in-car commuters. The system provides information about traffic, gas prices, and weather, based on real-time data obtained via web services. It was built as a telephone dialog system to enable any user with an ordinary cell phone to access these services without the need for a data plan or additional hardware or software installed in the car. An example dialog is shown in Figure 1.

The most critical part of a dialog system for location-based services is of course, how well the system understands locations spoken by the user. In the proposed system, the definition of location is varied by a context manager, based on the granularity required for a particular task. For example, some tasks may require only knowing the user's current city or neighborhood while others require the system to know the user's precise location. The system engages with the user to obtain the required location information in the most efficient way. In addition, the system can be personalized to a particular user, so common locations or "personal points of interest" are known by the system.

We have found that reliably *conveying* spoken locations to a user is equally important in dialog systems that provide location-sensitive information. We present a method for automatically converting

S: Welcome to Commute UX. Would you like traffic, gas prices, or weather?
U: I need the closest gas station in Redmond.
S: In that case, I need you to tell me the intersection where you are.
U: I'm on the corner of 148th and 35th.
S: You are at the corner of 148th Avenue NE and NE 35th Street, right?
U: Yes.
S: The nearest gas station is at the corner of NE 24th Street and 148th Avenue NE.
S: What else can I help you with?
U: How's the traffic to Seattle?
S: I'll get the traffic between Redmond and Seattle, right?
U: Yes.
S: The traffic between Redmond and Seattle, via SR-520, is …
S: What else can I help you with?
U: Nothing, thanks.
S: Good-bye.

Figure 1. An example dialog from Commute UX

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 87–94,
Antwerp, September 2007.

addresses into more descriptive locations using intersections and landmarks, in a manner that more closely resembles the way in which humans convey location to each other.

The Commute UX system was deployed to a limited group of real users and we evaluated the performance of the system based on an analysis of approximately 700 calls made to the system over an eight-week period. In addition, users of the system were surveyed in order to obtain a subjective evaluation.

2 System architecture

The architecture of Commute UX is depicted in Figure 2. In each turn, the system receives a voice input from the user, processes the input, and reacts to the user accordingly. Six functional modules are involved in this process: the speech recognizer, the semantic parser, the dialog manager, the context manager, the information retriever, and the response manager.

2.1 Speech recognizer

The task of the speech recognizer is to convert the voice input into text, from which semantic information will be extracted and processed. Its performance directly affects the task completion rate and the user satisfaction. Note that the acoustic model used by the speech recognizer is usually independent of the task. However the language model (LM) is highly task-dependent and its quality usually determines the recognition accuracy of the speech recognizer.

The design of the LM is both a science and an art, where a balance needs to be made between the accuracy of the keyword recognition and the flexibility of the speaking style it can support. In our system, we have used a strategy that trains a statistical LM from the slots (e.g., city name, road name, gas type) and information bearing phrases learned from sample queries (e.g. "… the closest gas station in <City> …") and augments it with a filler word N-gram (Yu, et al., 2006) to model the insignificant words. The filler part of the LM absorbs hesitations, by-talk, and other non-information bearing words unseen in the training sentences. The filler word N-gram is pruned from a generic dictation LM.

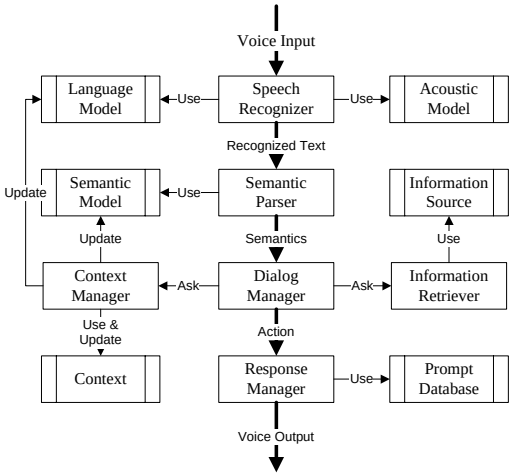


Figure 2. System architecture of Commute UX

2.2 Semantic parser

Semantic parser extracts the semantic information from the recognized text output from the speech recognizer. Converting information into its semantic representation has two benefits. First, semantic representation is more concise and consistent than the phrases. Using semantic representation greatly simplifies the subsequent processing in the later stages. Second, semantic representation is modality independent. By converting information into the same semantic representation, we make the rest of the system isolated from different input modalities. Adding new modalities thus becomes simple and cheap.

Extracting semantic information, however, is not trivial, especially since the output from the speech recognizer contains errors and users may convey multiple semantics in one utterance. The semantic information extracted includes the task classification, which is a generic call-routing problem, e.g. (Kuo, et al., 2002; Carpenter, et al., 1998), and task-specific semantic slots (e.g. origin city, destination city, time of day for weather forecast). Slot labeling is performed using a Maximum Entropy classifier (Berger et al., 1996) trained from the same LM training sentences.

2.3 Dialog manager

The task of the dialog manager is to determine the appropriate actions to take, given the current dialog context and the newly extracted semantic information. Note that both the speech recognizer and the semantic parser are not certain about their results.

The confidence from them needs to be taken into consideration when decision is to be made.

The dialog management is based on a two-level state machine in our system: the turn level and the dialog level. The turn level state machines are pre-built configurable and reusable dialog components such as system-led dialog component and mixed initiative dialog component. These state machines define the basic behaviors of a turn. For example, what to do when the confidence is low, medium, and high, and what to do when silence or mumble is detected. The dialog level (inter-turn) state machine defines the flow and strategy of the top level dialog. For example, what to do if the system cannot get what the user has said after trying twice. In our system, the top level state machine is designed so that it supports both free-form mixed initiative and strict system-led dialog. If the system cannot decipher some of the semantic slots in users' free-form utterances, the system will fall-back to the system-led dialog and guide the user step by step to achieve the user's goal. The user can also yield to the system-led dialog from the very beginning.

The dialog manager gets context information from the context manager and the information requested by the user through the information retriever. The information and prompts are delivered to the user through the response manager.

2.4 Context manager

The context manager plays a key role in Commute UX. Contexts in our system include the user information (e.g., user registered places, user's name, and past requests), the dialog history, and the semantic information confirmed so far. By maintaining current and accurate context information, the context manager can resolve semantic conflicts and make the system synchronous to the user's perceived state.

One important task of the context manager is to update the LM and the semantic model based on the context. By choosing the context dependent LM and the semantic model, the system can greatly reduce the perplexity and achieve higher recognition accuracy and lower number of turns.

2.5 Information Retriever

The information retriever provides an interface between the dialog manager and the backend information sources. In our system, the information is from three major sources: the relatively stable geographical database, which contains information such as cities, streets, intersections, and points of interest (POI); the rapidly changing real time information such as gas prices, traffic conditions, and weather conditions; and the user's registered information such as telephone numbers and personal points of interest (see Section 3.2).

2.6 Response Manager

The response manager presents information to the user or prompts the user for additional information. In our current system, the only presentation modality is voice and so the task of the response manager is to utilize the prompt database, synthesize the best audio output, and present the audio to the user. The system employs several strategies to decide the best manner in which to speak information to the user, as will be discussed in Section 4.

3 Understanding locations from the user

The crux of any dialog system focused on location-based services, such as Commute UX, is to reliably understand the locations spoken by the user. However, the notion of location and the required granularity of location can vary significantly based on the task. For example, for traffic or weather applications, a broad definition of location, such as neighborhood, city, or zip code, can be adequate, e.g. "How's the traffic between Seattle and Bellevue". However, for other tasks such as finding the nearest gas station, or route planning, the user needs to convey a precise location to the system. Finally, there is another distinction between personal locations that can vary based on the user, e.g. home and work, and geographic entities that have standard names and meanings.

3.1 Recognizing: from regions to points

In order to perform recognition of locations, a geographic database is crawled and the relevant information, such as the entity name, entity type and geolocation (latitude/longitude) or bounding box, is stored in a relational database. The database structure enables us to hierarchically categorize locations in a given state: zip codes contain cities, cities contain neighborhoods and points of interest, etc. All of these entities are valid locations in the application and are thus added to the grammar.

When the user makes a query, the parser processes the recognized text and isolates any locations

in the spoken utterance. These locations are then passed to the back-end database to find the location data for that entity. The database is searched from most specific location (personal point of interest) to the most general (city or zip code) in order to determine the user's intended location.

In some cases, the task itself dictates the scope of the location grammar. For example, traffic information is only available on major highways, and not local roads. Because we cannot provide a user with traffic information on local roads, a traffic query does not require the same precision in origin and destination as a task such as route planning. As a result, we simplify the task and allow users to make traffic queries only on the roads themselves ("How's the traffic on I-5 north?"), or between cities, neighborhoods, or personal points of interest ("How's the traffic between Bellevue and Seattle?"). This enables the dialog to be much more concise (the user does not have to convey two exact addresses) and because the grammar is more constrained, the accuracy is higher.

There are cases where the user's query can lead to ambiguities. For example, suppose the user asks for the traffic between two cities, and there are two common routes between the origin and destination. Our system will choose the most common route, and attempt to resolve the ambiguity by informing the user of the route it has chosen:

U: How's the traffic between Bellevue and Seattle?
S: The traffic between Bellevue and Seattle, **via I-90** is light, with an average speed of …

In this case, the system informed the user that traffic information provided was for the route taking Interstate 90. The user, who presumably knows both routes, can then query for the other route, by asking, "How about via 520?" The context manager maintains the origin and destination cities from the previous query and adds Highway 520 as a road to be included in the route between Bellevue and Seattle. The routing engine will then determine the route between these two cities that takes this highway, and then the corresponding traffic information can be retrieved and delivered to the user.

There are many instances where the user needs to convey an exact location to the system, not simply a city or neighborhood region. For example, if the user needs to find the closest gas station, or would like directions between two places. The most obvious way to convey an exact location is using an address. However, users often do not know a valid address for their current location, especially while they are driving. Even if an address were known, recognition errors make the use of addresses inefficient in conveying location. This was confirmed in (Venkataraman et al., 2003), where an iterative multi-pass approach using a class-based language model was proposed to improve the recognition of spoken addresses. The difficulty is even more apparent when one considers that state-of-the-art recognition accuracy for a five digit number in noise conditions that are realistic for mobile scenarios is about 90%. This means that one out of ten house numbers or zip codes will be misrecognized.

In (Seltzer et al., 2007), we proposed the use of intersections as a convenient and reliable means of conveying location. While the use of intersections alleviates some of problems found in address recognition, it is still a challenging problem. For example, there are over 3500 unique street names, and over 20,000 intersections in the city of Seattle. In addition, streets and intersections are highly acoustically confusable and often spoken informally, with incomplete specifications. For example a user might say "the corner of Third and Denny" rather than "the corner of Northeast Third Avenue and Denny Way".

To reliably recognize intersections, we employ an information retrieval approach. We construct a database of streets and intersections in a particular city. The intersections are treated as documents in a database, and phonetic-level features are derived from the word stings comprising these "documents". When the user utters an intersection, the recognized text is parsed into two street names and the phonetic level features are extracted each street name. Intersection classification is then performed using a vector space model with TF-IDF features. This approach allows the system to reliably recognize intersections in the presence of recognition errors and incomplete street names. Details about this method and an evaluation of its performance can be found in (Seltzer et al., 2007).

3.2 Personal Points of Interest as Locations

One key feature of the Commute UX system is an optional website registration for users. Users can create an account where they provide their phone number and specify any number of personal

points of interest (PPOI). These PPOI are specified by a friendly name (e.g. “Jane’s school”), an optional formal name (e.g. “Washington Middle School”), and an address. A back-end web service converts this address to a geolocation and this information is stored in the database. By default, the user is prompted to register home and work as personal locations. Users can then add additional PPOI. Each time a user changes some PPOI, the database is updated and the recognition grammars are regenerated to reflect the current list of unique PPOI friendly names and formal names. When a user calls the system, caller ID is performed as grammar entries corresponding to that user’s PPOI are activated. The caller’s phone number and the recognized PPOI are then used to retrieve the corresponding location form the database.

After a limited internal deployment, we have 276 registered users who created a total of 625 PPOI, but only 97 unique PPOI friendly names in the grammar. The three most popular PPOI were “home”, “work”, and “gym”.

The presence of PPOI also enables the system to assume some default behaviors. For example, if a registered user calls the system during common commuting times, the system will automatically fill the semantic slots with the home and work locations of that user and asks if the user would like the traffic information from home to work (or vice versa).

4 Rendering spoken locations to the user

The ability for the user to understand and remember the locations spoken by the system is as important as the system’s ability to understand the locations input by the user. Conveying locations to users in spoken dialog systems is problematic for several reasons. First, depending on the quality of the TTS voice, understanding a spoken location can be quite difficult, even in optimal conditions. In a vehicle, the environmental noise can make intelligibility even harder. The situation is exacerbated by the high cognitive load required by driving, so the user cannot fully focus on the system’s output speech. In addition, because the user’s hands and eyes are typically busy, s/he cannot write down the location as the system speaks it, and therefore must try to remember the location as closely as possible.

4.1 Automatically rendering locations using intersections and landmarks

To enable users to more easily understand locations, spoken by the system, we modeled the system’s output on the manner in which humans convey locations to each other. For example, a user calling a business to ask its location will often be told by the clerk, “We’re on the corner of 40th and 148th,” rather than “We’re located at 14803 40th Street.” Similarly, humans will often use landmarks, such as “We’re on Main Street near the Shell Station” or “We’re on the corner of Fifth and Mercer, near the Space Needle.

To create a similar capability in our system, we crawled a geographic database containing all streets and intersections along with their latitude/longitude coordinates in a particular city. In addition, we also crawled a database of points of interest (POI), also labeled with their geographic coordinates. These points of interest included a variety of entities, such as schools, libraries, parks, and government buildings. The information about streets, intersections and POI was stored in a database.

Using this information, locations that we want to convey to users, for example the location of a gas stations, are processed as follows. The address of the entity is converted to geographic coordinates. Using these coordinates, the intersections database is queried to find all intersections within 0.05 miles (approximately half a block). If multiple intersections are returned, they are ranked according to an intersection importance metric, defined as the sum of the total number of other intersections of which each constituent street in the given intersection is a member. The top ranked intersection is selected. Following the intersection search, the POI database is queried to identify any POI within 0.1 miles (one block) from the entity of interest.

After this process, each location we can return to the user is represented by its original address, as well as the nearest intersection and/or landmark, if either was found. For those locations that do have a nearby intersection and landmark, we have various ways to present the location to the user summarized in Table 1.

4.2 User preferences for spoken locations

We performed a user study to determine which of these four methods of rendering an address was

Address only	14803 Northeast 51st Street
Address & POI	251 Rainier Avenue North, near Renton Chamber of Commerce
Intersection only	The corner of East Madison Street and 17th Avenue
Intersection & POI	The corner of NE Woodinville Road and 131st Avenue, near City Hall

Table 1. Address representations in Commute UX

preferred by users of a spoken dialog system. Users of the study ran a program on their desktop PCs. Each trial of the study was as follows. The system randomly selected an address from our database of gas station locations. This location was rendered in one of the four styles described in the previous section. The user listened to a TTS engine speak the location. Once the location was spoken, the user was asked to type in as much of location as they could remember. The user could not start typing until the TTS output was complete. The system then randomly chose another address from the database, and rendered it in a style randomly selected from other the three remaining methods. The user again listened to the TTS engine speak the location and had to type in as much of the location as they could remember. After the user completed these two locations spoken in different ways, s/he was asked which, if any, of the two styles was preferred. This completed a single trial of the study. Each user performed a minimum of three trials.

Preferences for location rendering were evaluated based on 40 users who completed a total of 133 trials. The users' data was hand-scored and analyzed in terms of accuracy and user preference. Users' ability to accurately remember spoken locations in these different styles was scored as follows. Addresses and intersections both contain two critical elements (the number and the street name in the former, the two street names in the latter). For locations spoken as addresses or intersections, each element the user correctly identified (within a tolerance of 0.1 miles) is given 0.5 points. Correct recognition of both elements therefore received 1 point. Correct recognition of a spoken POI received 1 point regardless of whether the other elements are correct. Thus, each address transcribed by the user was scored from zero to one in the following way:

$$r = \max\left(\frac{r_1}{2} + \frac{r_2}{2}, r_{POI}\right) \qquad (1)$$

Question type	Number	Sum	Accuracy (%)
Address only	67	57.5	85.82
Address & POI	65	53.5	82.31
Intersection only	65	54.0	83.08
Intersection & POI	69	47.7	69.13

Table 2. Recognition rate for various address representations.

where r_1, r_2 and r_{POI} are either 0 or 1 and are the recognition of the first element, second element, and POI.

The averaged recognition results for each one of the four address representations are shown in Table 2. While the first three representations have approximately the same recognition rate, it is substantially lower for "Intersection & POI". This representation was typically the longest and is therefore the most difficult to remember.

The user preferences are evaluated as follow. For each trial, the preferred representation receives one point. If the user had no preference between the two styles, both are assigned 0.5 points. The final score is weighted with the recognition rate – we weight more these preferences which are properly recognized:

$$p_k = \frac{\sum_i p_i^{(k)} r_i^{(k)}}{\sum_i r_i^{(k)}} \quad k \in [1,4] \qquad (2)$$

where $p_i^{(k)}$ is the preference score of the *i*-th session, where the address is represented in *k*-th way; $r_i^{(k)}$ is the recognition result for the same session, computed by equation (1). Both the non-weighted and weighted average preference scores are shown in Figure 3. Rendering a spoken location using the intersection is clearly preferred, followed closely by the combination of intersection and POI. Because the combination of intersection & POI resulted in the lowest recognition accuracy, we set the system to refer to locations using the nearest intersection whenever possible. In feedback solicited from the users after this study, several participants stated that POI helped only when they were familiar with the area. Otherwise, it was not helpful and added confusion. This indicates that location-based services targeting commuters and residents may want to use POI in describing locations

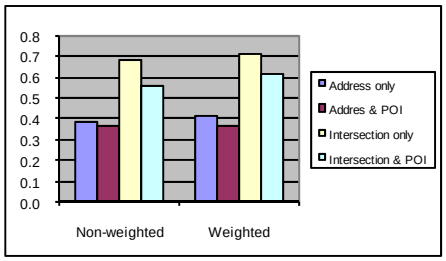


Figure 3. User preferences for address conveying.

to users, while those targeting tourists or business travelers should not.

5 Initial Deployment and Evaluation

The initial version of the Commute UX dialog system can process requests for information about traffic, cheapest and nearest gas stations, and weather in Washington State. The system was demonstrated to approximately 800 Microsoft employees in Redmond, WA campus at the beginning of March, 2007. It was made available to all Microsoft employees but no additional effort was made to actively recruit users. The results presented in this paper are based on an eight week period between March 12, 2007 and May 6, 2007. During this time, a total of 276 users enrolled at the Commute UX website, specifying a phone number and PPOI.

5.1 Analysis of calls

The system received 698 calls during this time period, or 12.5 calls per day. Of these calls, 62.2% were from registered users, while 37.8% were from non-registered users. There were calls from 214 unique phone numbers, of which 55% were registered users. This translates to approximately 3.3 calls per user. However, the distribution of calls per user is not uniform, a 40 users accounted for 50% of the calls during this time period.

From these calls, there were total of 927 tasks that users tried to perform. A task is defined as the user's attempt to obtain a piece of information from the system. In our system, the possible tasks are obtaining a traffic report, the location of the cheapest or nearest gas station, or a weather report. The traffic is the most frequently called with 55% of all queries, followed by the gas prices with 27%, and weather with 17%.

Table 3 shows the average number of turns for each of the three tasks and across all tasks. The results are shown for all users as well as for registered and non-registered users alone. Non-registered users use 0.7 more turns than registered users. The only difference between registered and non-registered users from the system's point of view is the presence of PPOI. We believe that the use of PPOI enables users to obtain the information they want efficiently with fewer dialog turns.

Task Type	All	Registered	Non-registered
Traffic	3.56	3.33	4.08
Gas Prices	3.73	3.54	4.14
Weather	3.80	3.61	4.41
Total	**3.65**	**3.44**	**4.14**

Table 3. Averaged number of turns per task type.

This theory is further validated when we examined the task completion rate. Figure 4 shows the task completion rates for the various tasks as a function of all users, registered and non-registered users. Overall, there is a 65.6% task completion rate. It is interesting to note, however, that registered users obtain a consistent task completion rate of about 70% across all tasks, while the task completion rate of non-registered users varies dramatically from 48% for the traffic task to 64% for the weather task. The traffic application is the only application that requires multiple locations: both an origin and destination. Coincidentally, traffic is also the application that is most likely to use PPOI as many users query the system for traffic information during their commutes between home and work. For calls made during these times, the registered users have only to confirm that they would like the traffic report between home and work, while non-registered users have to convey two locations to the system for the same request. Thus, the use of PPOI results in fewer turns in the dialog, and leads to a significantly higher task-completion rate for registered users.

5.2 User evaluation

To obtain a more subjective evaluation of the Commute UX system, we sent out a web-based survey to users of Commute UX who had made at least one call to the system and those who participated in the user study discussed in Section 4.2, whether they were registered or not. From this solicitation, we received 23 responses.

The survey asked the users to state their level of agreement to a series of statements, using a five-step scale that ranged from Strongly Agree to

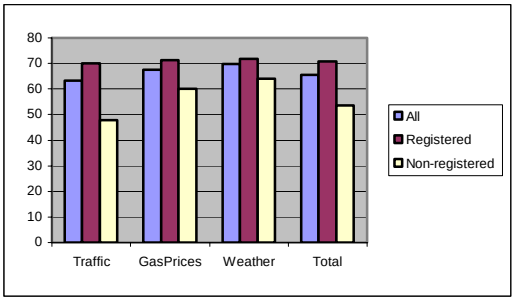


Figure 4. Completion ratio (%) per task.

Strongly Disagree. The questions and the responses are shown in Figure 5. As the results show, a majority of the respondents find the system useful and believe the system understands their speech. It is interesting to note that most users believe they are speaking in a natural manner, yet a similar number claim to only answer the questions the system asks. This contradicts our usual notion that system initiated dialog is not perceived as natural.

The other interesting conclusions from this data concern personalization. We note that several people use PPOI but most do not use PPOI other than the default "home" or "work" locations. Finally, we note that there are a significant number of users that always ask for the same information from the system. This indicates that there is a large opportunity for further improvement in task completion with additional personalization and user-specific grammar adaptation in this domain.

6 Discussion

In this paper, we presented a telephone dialog system for location-based services. It utilizes several key technologies for both recognizing and rendering spoken locations. We performed a user study to evaluate the users' response to various ways of describing a spoken location in terms of addresses, intersections, or points of interest, and designed our system to operate in the manner that both provided the best accuracy and was most preferred by users. The system also enables users to improve their experience with personal points of interest. The use of these personal locations resulted in dialogs with a higher task completion rate and fewer turns per task. A subjective user evaluation of the system revealed that most users had a positive experience with the system, but that there were opportunities for additional improvement through further personalization and user adaptation.

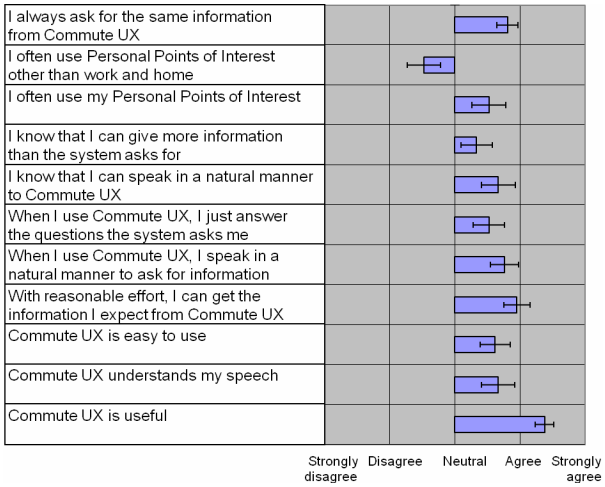


Figure 5. User survey results.

References

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, Vol. 22.

Bob Carpenter and Jennifer Chu-Carroll. 1998. Natural Language Call Routing: A Robust, Self-organizing Approach. *Proc. ICSLP*. Sydney, Australia

Alexander Gruenstein, Stephanie Seneff, and Chao Wang. 2006. Scalable and Portable Web-based Multimodal Dialogue Interaction with Geographical Databases. *Proc. ICSLP*, Pittsburgh, PA.

Hong-Kwang Jeff Kuo, Chin-Hui Lee, Imed Zitouni, Eric Fosler-Lussier, Egbert Ammicht 2002. Discriminative Training for Call Classification and Routing. Proc. ICSLP. Denver, CO.

Michael L. Seltzer, Yun-Cheng Ju, Ivan Tashev, and Alex Acero. 2007. Robust Location Understanding in Spoken Dialog Systems Using Intersections. Submitted to *Interspeech 2007*, Antwerp, Belgium.

Omer Tsimhoni, Daniel Smith, and Paul Green. 2004. Address entry while driving: speech recognition vs. a touch-screen keyboard. *Human Factors*, vol. 46, no. 4, pp. 600–610.

Anand Venkataraman, Horacio Franco, and Greg Myers. 2003. An architecture for rapid retrieval of structured information using speech with application to spoken address recognition. *Proc. of ASRU*, USVI.

Dong Yu, Yun-Cheng Ju, Ye-Yi Wang, and Alex Acero. 2006. N-gram based filler model for robust grammar authoring. *Proc. ICASSP*, Toulouse, France.

Corpus-Based Training of Action-Specific Language Models

Lars Schillingmann* and **Sven Wachsmuth** and **Britta Wrede**

Bielefeld University, 33615 Bielefeld, Germany,

{lschilli, swachsmu, bwrede}@techfak.uni-bielefeld.de

http://www.techfak.uni-bielefeld.de/ags/ai/

Abstract

Especially in noisy environments like in human-robot interaction, visual information provides a strong cue facilitating a robust understanding of speech. In this paper, we consider the dynamic visual context of actions perceived by a camera. Based on an annotated multi-modal corpus of people who verbally explain tasks while they perform them, we present an automatic strategy for learning action-specific language models. The approach explicitly deals with the asynchrony of actions and verbal descriptions and includes an automatic parameter optimization based on a perplexity measure. Results show that a significant improvement of the word accuracy can be achieved using a dynamic switching of action-specific language models.

1 Introduction

While speech recognition is an easy task for humans even under difficult acoustic conditions, current ASR systems still cannot compete with humans (Potamianos et al., 2003). This is especially true in human-robot interaction, where one has to deal with spontaneous speech effects, noisy environments, communicative gestures, and a frequent referencing to visual objects and events. In this case, speech recognition and understanding becomes a multi-modal issue. This has also been emphasized by several psychological studies that suggest a very early interaction between vision and speech processing (Spivey et al., 2001). For the practical development of speech understanding components for robotic interfaces, there are three implications. First, there is a need for multi-modal corpora in order to train and evaluate more sophisticated speech recognition models. Secondly, visual and acoustic speech events need to be synchronized and aligned with regard to semantic content for learning as well as interpretation. Thirdly, new strategies for the early integration of visual information into the speech recognition process need to be developed. In this paper, we focus on the first and second issues and show first results for the third.

The integration of speech and visual context can be treated on different levels of processing that depend on the kind visual information considered. Motivated by the McGurk effect (1976) audiovisual speech recognition (AVSR) systems have been developed. These systems integrate acoustic features with those extracted from the speakers face. This is an approximately synchronous process during speech production. In AVSR, typically Hidden Markov Models (HMMs) are used for modelling the acoustic and visual features. The approaches mostly differ in the handling of slight asynchrony between the two feature streams. The methods range from simple feature concatenation which does not allow asynchrony at all up to more flexible HMM architectures (e.g. Product-HMMs) allowing ca. 100 ms of asynchrony in practice (Potamianos et al., 2003).

Other systems proposed integrate features from a static visual scene into speech recognition. Knowledge inferred from a visual scene can be used to generate grammars for object descriptions (Naeve et al., 1995). These grammars are used as language model to improve speech recognition. Deb Roy (2005) reports a system, which fuses knowledge of the visual semantics of language and the specific contents of a visual scene during speech processing. Based on

*Partially supported by the Federal Ministry of Education and Research Germany (Joint Project DESIRE)

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 95–102,
Antwerp, September 2007.

the current scene layout the system generates possible word sequences for object descriptions from a probabilistic grammar. These are weighted by a likelihood associated with each object in the scene. The result is a bi-gram model, which is dynamically updated using a visual attention mechanism incorporating the partially processed utterance. This model is used to bias speech recognition. Both approaches have in common that the scene information remains static during speech processing. Thus, the synchronization problem can be neglected and the integration is done on the level of utterances. In this case also late integration schemes are possible that infer a joint multi-modal meaning after a word sequence has been recognized (Wachsmuth and Sagerer, 2002).

The timing and synchronization becomes relevant when dynamic visual events are considered as visual context. Two different cases can be distinguished. On the one hand, communicative gestures like pointing provide information that is directly related to the syntactic structure of the sentence. As a consequence, these are approximately synchronized with the corresponding noun phrases and partially marked in the wording. In this area, different research groups have started to collect multimodal corpora (Green et al., 2006; Wolf and Bugmann, 2005; Maas and Wrede, 2006). However, in these settings, the scene environment is still static and the kind of visual information provided is of limited use in speech recognition.

On the other hand, human actions or action sequences that are verbally commented are the most informative but also most flexible case. Usable corpora for speech recognition training as well as evaluation are still rare. Integrating this information into speech recognition broaches two problems. First, humans do not execute actions synchronously while describing a task verbally. The degree of asynchrony lays in a range of several seconds as reported in (Wolf and Bugmann, 2006). Hence, it is not possible to integrate this information using HMM architectures as used in AVSR. Second, the actions change in the course of an utterance. Thus, the contextual information is not static as in the previous systems utilizing visual scene contents.

In this paper, we present a corpus-based method for training and optimising action-specific language models. The goal is to improve recognition accuracy by using these models during speech processing. Training data for the language models is collected using a scenario described in section 2. Section 3 describes our method of associating utterance parts to actions. The resulting action-specific training data is used in an automated language model training and optimisation process. The results of this process are discussed in section 4.

2 Scenario and data collection

Figure 1: A test subject describes a task while performing it.

Our scenario resembles a situation in which a user teaches a new task to a robotic system. A test subject sits in front of a table with several objects (e.g. a cup and a plant) on it that can be utilized for different manipulative actions (Figure 1). Only a subset of the objects is relevant for the following demonstration. The subject is instructed to explain some simple tasks to the system while performing the corresponding action sequence. In order to suppress deictic gestures and too complex descriptions they have to imagine, that their communication partner is intelligent and knows the setup. The tasks are watering a plant, preparing tea and preparing coffee. In order to generate more varying utterances the test subjects have to perform each task twice with three different object layouts. The second time they are additionally instructed to name colours and object relationships if possible. The utterances are recorded using a headset microphone and the scene is recorded by video. A corpus is collected containing the utterance transcriptions and time intervals, which annotate the actions. The actions performed are annotated in the video based on an abstraction hierarchy

as depicted in Figure 2). The choice of the compositional granularity was based on two reasons. First, the corresponding primitives can be detected using a pre-trained trajectory based action recogniser (Li et al., 2006). Secondly, the verbalization happened on that level due to the instructions given.

The resulting corpus consists of 195 utterances from 11 test subjects (17.7 utterances per person). The overall length is about 38 minutes. The average utterance length is about 12.7 s with about 33 words per utterance. The entire corpus includes 6 429 words with a lexicon size of 288 different words. The videos are annotated with 11 different actions. The average length of an action interval is 1.75 s. All in all 999 intervals with an overall length of about 29 minutes have been annotated. Each utterance contains 5.5 actions in average.

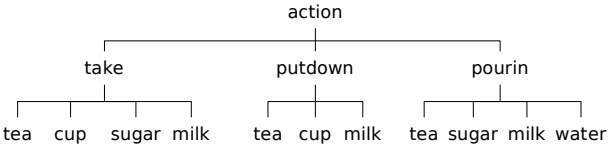


Figure 2: Hierarchic structure of actions used for annotation.

The following section describes how action-specific language models are created using this corpus.

3 Action-Specific Language Models

Speech recognition models are typically formulated distinguishing acoustic and language models. The standard technique for language models are n-grams that have proven their effectiveness over many years (Rosenfeld, 2000). For acquiring realistic language models, n-grams need to be trained using a representative sample. In the present approach, we assume that the wording will be biased by the action, which the speaker performs and describes in parallel. Thus, we aim at the estimation of action-specific language models. In order to gain corresponding action-specific samples two problems need to be solved. First, a method is required, which is able to associate speech with action intervals in order to extract action-specific parts from an utterance. Secondly, our approach requires temporal information (word intervals) for both the actions and the speech. The utterance transcriptions from the above-described corpus are not annotated with temporal information in contrast to the video annotation. Manual annotation on that level of detail is expensive. Thus, we use an automated approach, which is described in the next section. Afterwards we elaborate on our approach to the first problem.

3.1 Gaining Time Information

The temporal information of an utterance with a known transcription can be gained by using a so-called forced alignment. Our speech recogniser (Fink, 1999) uses Hidden Markov Models (HMMs) as acoustic models. Existing models trained on a speech corpus are used. Words not in the lexicon are defined by new compound models based on phoneme HMMs. In a forced alignment, the model topology is restricted in accordance with each utterance transcription. This means the order of word models is fixed for each transcription ensuring a correct alignment although the acoustic quality varies depending on the speaker. Since the transcription does not contain pauses or spontaneous speech effects, the model topology needs to be adapted accordingly. An "<other>" model for these effects is optionally allowed between words. Figure 3 shows a schematic diagram of the model topology. For

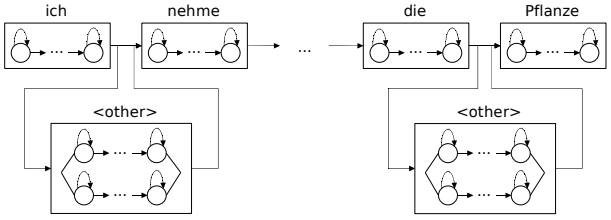


Figure 3: Schematic diagram of a HMM topology with fixed word model order and optional "<other>" models.

each utterance, a sequence of MFCC feature vectors is extracted following standard speech recognition techniques. The Viterbi algorithm is used to calculate the state sequence $\boldsymbol{s}$ through the model topology which produces the feature vector sequence $\boldsymbol{o}$ with the maximum probability given the HMM λ:

$$\boldsymbol{s}^* = \underset{\boldsymbol{s}}{\operatorname{argmax}}\, P(\boldsymbol{o}, \boldsymbol{s}|\lambda) \qquad (1)$$

After the Viterbi alignment, the resulting state sequence can be used to calculate the time interval for

each word since the frame length used during feature extraction is known. After this step, the temporal information is available for both the utterance transcription and the action annotation. The following section explains the next step where the temporal information is used to associate utterance parts with actions.

3.2 Pairing of Speech and Actions

The main problem when speech has to be associated with action intervals is that the utterance parts semantically belonging to actions are asynchronous on the time-line (Wolf and Bugmann, 2006). Thus, a distance measure $d(w_i, a_j)$ is calculated between each word w_i and action a_j. A set of tolerance parameters is used to decide if a word is assigned to an action. By choosing these parameters appropriately, the asynchrony between speech and actions can be respected. Since the time shift is not longer than several seconds this procedure is suitable. Multiple cases have to be handled when calculating with temporal intervals, which are systematically structured by Allen's calculus (Allen, 1983). Our method uses a subset of these relationships. Each type of action uses independent tolerance parameters to the left h_j^l and the right h_j^r. They are used depending if w_i is before or after a_j respectively. Pauses detected during the forced alignment give hints about the change of an action. Thus, silence is weighted additionally using a penalty parameter g_j so that silence between an action and a word further increases the temporal difference. Figure 4 illustrates the distance measure when silence has to be considered.

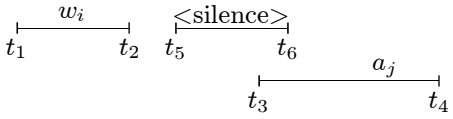


Figure 4: The distance function between two word intervals under the above constellation is defined as $d(w_i, a_j) = t_3 - t_2 + g_j \cdot (t_3 - t_5)$.

A word is associated with an action if the following condition is true:

$$-h_j^r < d(w_i, a_j) < h_j^l \qquad (2)$$

Figure 5 gives a simple example about the assignment strategy. The tolerance parameters are determined automatically and individually for each language model using an optimisation method, which is described in section 3.4.

3.3 Language Model Training

The objective of the language model training is to create a n-gram-model for each action type, which predicts the action-specific utterance parts most accurately. These models could directly be trained with the results of the above assignment strategy but it is likely that these models become too specific. Therefore, the training data is structured using the hierarchy defined in figure 2. The top level refers to the complete utterance. The second level addresses utterance parts on a more general action level e.g. "take" or "put". The third level reaches the highest level of granularity with action-object specific utterance parts. During training each level can be weighted using an individual factor (see figure 6). The set of weighting factors is specific for each language model. Thus, each language model has an individual degree of specialisation depending on these factors. The training data required in this process is generated using the speech and action pairing process with an individual parameter set. Both the pairing parameters and the weighting factors are optimised specifically for each language model using a method described in the following section.

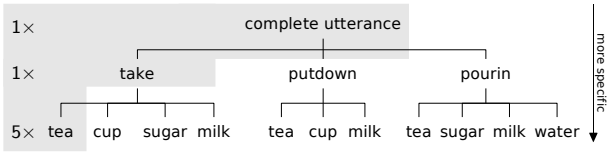


Figure 6: Structure of the training data using the action hierarchy. The highlighted path shows by example, which parts are used and weighted to train one language model.

During model estimation, absolute discounting and backing-off are used to handle unseen events. The counts $c(\boldsymbol{y}z)$ of a word z with history $\boldsymbol{y}$ are modified with an absolute value β in order to gain probability mass for unseen events so that the relative frequencies are defined as:

$$f^*(z|\boldsymbol{y}) = \frac{c(\boldsymbol{y}z) - \beta}{c(\boldsymbol{y}\cdot)} \qquad \forall \boldsymbol{y}z \; c(\boldsymbol{y}z) > \beta \qquad (3)$$

Where $c(\boldsymbol{y}\cdot)$ denotes all events with history $\boldsymbol{y}$.

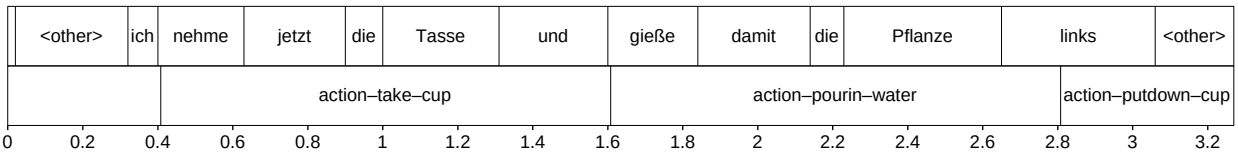


Figure 5: Augmented utterance transcription and action annotation on one time axis ($t[s]$). Assuming pourin-water has a tolerance of 0.5 s to the left and 0 s to the right the part "Tasse und gieße damit die Pflanze links" is assigned to this action.

3.4 Parameter Optimisation

In the above sections, we have introduced several parameters. The tolerance parameters and the penalty factors for silence sum up to 33 in total considering all 11 action types. In addition, the weighting factors in the training data structure count 33 in total. This large number of free parameters cannot efficiently be determined manually. Thus, we use an optimisation method, which uses the perplexity to measure the quality of the action-specific models. We firstly describe the method in general and go into detail in the next paragraph.

In order to compute the perplexity a test sample is required. Since our corpus is relatively small, the choice of the test sample has large influence on the perplexity. Therefore the perplexity is computed using a leave-one-out cross validation (Kohavi, 1995). The utterances of one person are used as testing data on each run; the others are used for training. Firstly, a parameter set with the above parameters is generated. This parameter set is used to train language models with the method described in the last two sections. The testing data is gained using the same parameter set. Secondly, the perplexity is computed for each excluded test subject. The average perplexity regarding an action-specific language model is the final measurement of this model and the underlying parameter set. Thus, a parameter optimisation also finds the tolerance parameters for speech action assignment. The asynchrony between speech and actions is respected this way. This method depends on the assumption that actions frame semantic units, which are verbalised similarly. Therefore, a correct assignment of speech to actions results in a better perplexity rating.

In detail, the optimisation is realised by evaluating a large number of parameter sets automatically. The tolerance parameters to the left and the right are varied in a range from 0 to 3 seconds using an increment 0.5. The silence penalty is varied in a range from 0 to 2 analogously. The training data is weighted zero or once on utterance level. The action-level weighting is varied between 0 and 5. On the action-object level, weighting factors from 1 to 10 have been explored. We have chosen 12 sets of these factors in order to evaluate models with different degrees of specialisation. All combinations of these parameters result in 2 892 different sets. Each one is used to generate a complete set of action-specific bi-gram language models. Unseen events are handled using absolute discounting with $\beta = 0.8$. Due to the large number of parameter sets and the resulting complexity, this factor has not been made subject to optimisation. Furthermore, the discounting factor has insignificant influence regarding this method as informal tests have shown.

After the action-specific language models have been created the perplexity is computed so that each combination of language model and the underlying parameter set is associated with one. This way the perplexity can be used as optimisation criterion to find the best language model for each type of action. In the following section we present first results gathered using these models during speech processing.

4 Results

The language models' quality is evaluated by assessing the corresponding speech recognition performance. Our speech recogniser uses a standard time synchronous integrated search strategy to weight hypotheses generated by the acoustic model additionally with the language model. We have implemented a strategy, which enables the speech recogniser to switch language models during speech processing

	W_{ACC} %		W_{CORR} %
Action-Specific	65.98	±1.1	68.77
Base Model	69.39	±1.1	71.96
Difference	−3.41		−3.19
Random Usage	48.61	±1.2	51.36

Table 1: Recognition results (expand strategy) using optimised action-specific language models, trained with utterance parts on action-object level only.

Action	Base perp.	Model perp.	Diff
take-cup	20.84	16.55	4.29
take-tea	34.90	16.97	17.93
take-sugar	24.17	14.04	10.12
take-milk	22.68	19.28	3.40
putdown-tea	28.39	9.83	18.56
putdown-cup	23.01	15.11	7.90
putdown-milk	30.48	12.03	18.45
pourin-tea	41.21	11.95	29.27
pourin-sugar	20.39	12.50	7.89
pourin-milk	36.32	12.54	23.78
pourin-water	34.51	16.10	18.41

Table 2: Comparison of the perplexity regarding the action-specific models against the perplexity using a standard bi-gram trained on the whole utterances. The language models are trained with utterance parts on action-object level only.

using a set of switch points. In our case these switch points are generated from the action annotation. Two strategies have been implemented. The *stick* strategy uses exactly the interval borders and a default model when no annotation is available e.g. between two intervals. The *expand* strategy expands each action interval as far as possible so that an action-specific model is always used. All results are computed using a leave-one-out cross validation as described in section 3.4. The audio data belonging to the excluded test subject for each run is used for evaluating the speech recognizer. Afterwards the word accuracy W_{ACC} and the word correctness W_{CORR} are calculated.

In order to see how the degree of specialisation affects the recognition results it is possible to apply restrictions during optimisation. In the following, we

	W_{ACC} %		W_{CORR} %
Action-Specific	70.56	±1.1	73.20
Base Model	69.39	±1.1	71.96
Difference	**1.17**		**1.24**
Random Usage	69.22	±1.1	71.97

Table 3: Recognition results (expand strategy) using optimised action-specific language models, trained using the utterance level always once. Weighting factors have been made subject to optimisation.

Action	Base perp.	Model perp.	Diff
take-cup	20,43	17,59	2,84
take-tea	26,59	25,15	1,44
take-sugar	23,36	18,98	4,38
take-milk	22,68	21,63	1,05
putdown-tea	26,36	20,57	5,80
putdown-cup	22,51	20,91	1,60
putdown-milk	30,46	21,95	8,51
pourin-tea	27,27	22,51	4,77
pourin-sugar	20,33	15,40	4,93
pourin-milk	31,34	25,46	5,88
pourin-water	29,53	24,62	4,91

Table 4: Comparison of the perplexity regarding the action-specific models against the perplexity using a standard bi-gram trained on the whole utterances. The language models are trained using the utterance level always once.

Action	Tolerance [s] left	right	Silence-penalty
take-cup	2.00	1.00	2.00
take-tea	3.00	3.00	0.00
take-sugar	0.00	3.00	1.00
take-milk	3.00	2.50	0.00
putdown-tea	2.50	0.00	0.50
putdown-cup	3.00	0.50	0.50
putdown-milk	0.50	0.00	1.00
pourin-tea	0.50	2.50	1.00
pourin-sugar	0.50	1.00	1.50
pourin-milk	0.00	2.00	0.00
pourin-water	2.50	1.50	0.00

Table 5: Tolerance parameters found by the optimisation process (cp. table 4). The language models are trained using the utterance level always once.

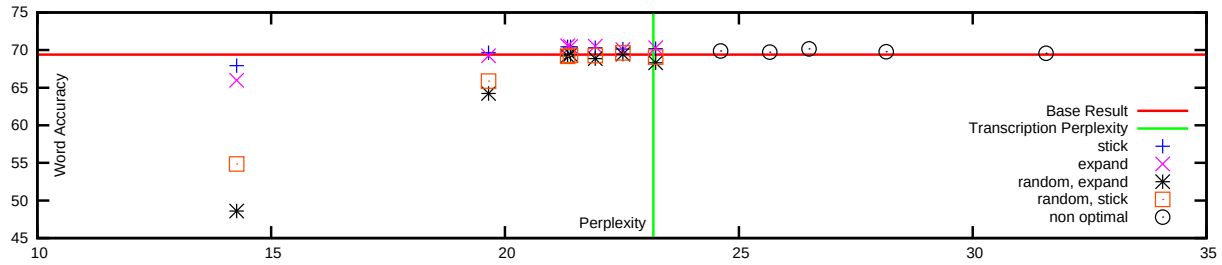


Figure 7: Overview of the average perplexity against word accuracy for all evaluation results. Models that are more specific have a lower perplexity. The difference between random and correct usage is larger for models that are more specific. The optimal results are slightly more specific than the standard bi-gram. Non-optimal models with up to 80 % of the top rated models thrown away do not reach this result. The keywords expand and stick denote the switching strategy where expand means each action interval is expanded as much as possible.

Action	Weighting Factors		
	Utt.	Ac.	Ac.-Obj.
take-cup	1	0	3
take-tea	1	0	1
take-sugar	1	0	3
take-milk	1	0	3
putdown-tea	1	0	5
putdown-cup	1	0	1
putdown-milk	1	1	10
pourin-tea	1	1	5
pourin-sugar	1	1	5
pourin-milk	1	0	5
pourin-water	1	0	3

Table 6: Weighting factors determined during parameter optimisation (cp. table 4).

present detailed results using very specialised models on the one hand and results where the degree of specialisation has also been made subject to optimisation on the other hand. The results are compared against recognition results using a standard bi-gram model trained on the complete utterance level (base result). Another comparison is made against results where an action-specific model is randomly selected for each action interval during speech recognition in order to evaluate their level of specialisation.

Table 1 shows results using very specific models trained with utterance parts on action-object level only. The models are too specific since the results are less good than using a standard bi-gram model. The perplexity difference in table 2 shows that these models are much more specific to the action context than the standard bi-gram model. The random usage result confirms that parts not belonging to the corresponding action context are not well described by the model.

Since very specific models with a low perplexity do not improve recognition results restrictions are applied during optimisation. The results in table 3 are generated using language models, which have been trained using the utterance level always once. The other weighting factors have been made subject to optimisation. The results are significantly better in comparison to the standard model. In contrast to the very specific models, the perplexity difference to the base model is smaller (see table 4). The random usage results emphasise the high level of generalisation. Table 5 shows the optimised tolerance parameters. The according weighting factors are shown in table 6. As one can see, the action-level seems to be of less importance to the specialisation and is therefore rarely used.

We have evaluated more action-specific models optimised under different restrictions. These results are summarized in figure 7. In order to verify that our method actually finds action-specific models which have better results than others trained during the optimisation process we have additionally evaluated non-optimal action-specific models with a lower perplexity. These models are selected by leaving different percentages (from 10 % up to 80 %) of the top rated models unconsidered during the opti-

misation process. The figure shows that these models indeed create worse recognition results than the fully optimised ones.

5 Outlook

We have demonstrated an approach to include visual context into speech recognition realised by means of action-specific language models, which are automatically trained and optimised. The action-specific utterance parts required for training are gained using an automatic associating method between actions and speech. The method only requires manual annotation on a level of low detail. The perplexity is used as optimisation criterion for the training parameter sets and a detailed analysis shows the adequacy of this approach. In order to ensure a certain level of generalisation the complete utterance level has to be always used. The optimisation under this restriction delivers the best results, which are significantly improved in comparison to speech processing with a standard bi-gram model.

Although this approach is able to improve speech recognition, the pairing of speech and actions happens on a heuristic level. Further research has to show in how far this association delivers semantically correct results. In contrast to knowledge-based methods, our approach can easily be transferred to other domains due to the automated pairing and training process.

Further applications of action-specific language models could make it possible that action hypotheses are extracted during speech recognition. In order to realise that, multiple models could be matched against each other during speech processing.

References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November.

G. A. Fink. 1999. Developing HMM-based recognizers with ESMERALDA. In Václav Matoušek, Pavel Mautner, Jana Ocelíková, and Petr Sojka, editors, *Lecture Notes in Artificial Intelligence*, volume 1692, pages 229–234, Berlin Heidelberg. Springer.

A. Green, H. Hüttenrauch, E. A. Topp, and K. S. Eklundh. 2006. Developing a contextualized mulimodal corpus for human-robot interaction. In *Proc. of Int. Conf. on Language Resources and Evaluation (LREC)*, Genua.

Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence*, pages 1137–1145.

Zhe Li, Jannik Fritsch, Sven Wachsmuth, and Gerhard Sagerer. 2006. An object-oriented approach using a top-down and bottom-up process for manipulative action recognition. In *DAGM06*, volume 4174 of *Lecture Notes in Computer Science*, pages 212–221, Heidelberg, Germany. Springer-Verlag.

Jan F. Maas and Britta Wrede. 2006. BITT: A corpus for topic tracking evaluation on multimodal human-robot-interaction. In *Proceedings of the international conference on Language and Evaluation (LREC)*, Genoa, Italy.

Harry Mcgurk and John Macdonald. 1976. Hearing lips and seeing voices. *Nature*, 264(5588):746–748, Dezember.

U. Naeve, G. Socher, G. A. Fink, F. Kummert, and G. Sagerer. 1995. Generation of language models using the results of image analysis. In *European Conference on Speech Communication and Technology*, pages 1739–1742, Madrid.

G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior. 2003. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326.

R. Rosenfeld. 2000. Two decades of statistical language modeling: where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, Aug.

Deb Roy and Niloy Mukherjee. 2005. Towards situated speech understanding: visual context priming of language models. *Computer Speech & Language*, 19(2):227–248, April.

M. J. Spivey, M. J. Tyler, K. M. Eberhard, and M.K. Tanenhaus. 2001. Linguistically mediated visual search. *Psychological Science*, 12(4):282–286, July.

S. Wachsmuth and G. Sagerer. 2002. Bayesian Networks for Speech and Image Integration. In *Proc. of 18th National Conf. on Artificial Intelligence (AAAI-2002)*, pages 300–306, Edmonton, Alberta, Canada.

J. C. Wolf and G. Bugmann. 2005. Multimodal corpus collection for the design of user-programmable robots. In *TAROS 2005 Towards Autonomous Robotic Systems Incorporating the Autumn Biro-Net Symposium*, September.

J. C. Wolf and G. Bugmann. 2006. Linking speech and gesture in multimodal instruction systems. In *IEEE International Symposium on Robot and Human Interactive Communication*, pages 141–144, Hatfield, UK, September.

Negotiating Spatial Goals with a Wheelchair

Thora Tenbrink
I5-[DiaSpace]
SFB/TR 8 Spatial Cognition
University of Bremen
tenbrink@sfbtr8.uni-bremen.de

Shi Hui
I5-[DiaSpace]
SFB/TR 8 Spatial Cognition
University of Bremen
shi@informatik.uni-bremen.de

Abstract

We present our iterative approach to enabling natural dialogic interaction between human users and a wheelchair, based on the alternation of empirical studies and dialogue modelling. Our approach incorporates empirically identified conceptual problem areas and a dialogue model designed to manage the available information and to ask clarification questions. In a Wizard-of-Oz experiment employing the first version of the model, we test how verbal robotic reactions can enable users to provide the information needed by the wheelchair to carry out the spatial task. Results show that the output must be extraordinarily coherent, temporally well-placed, and aligned with the user's descriptions, as even slightly deviating reactions systematically lead to confusion. The dialogue model is improved accordingly.

1 Introduction

Most advanced work on dialogue systems focuses on human-computer interaction scenarios in which either the user requires information from an expert system (e.g., Kruijff-Korbayová et al. 2002), or the user and the system negotiate a joint task such as making reservations (Rieser & Moore 2005), or the system engages in tutoring the user within a specific area of interest (Clark et al. 2005). In such tasks, there are typically no particular complications with respect to time or space: Although the dialogue takes place in real time, there are no fundamental context-related effects of temporal delay or spatial mismatch. Complementing this research, there is a growing interest in dialogue systems employed in real time in spatially embedded interaction scenarios, such as situated human-robot dialogue. Such scenarios typically employ robots designed to accomplish service tasks for users instructing them by using natural language. Work in this area often focuses on a number of specific techniques designed to overcome the particular complexity of such a situation (e.g., Lemon et al. 2003, Spexard et al. 2006, Kruijff et al. 2007). Our own work fits into this latter endeavour by focusing on the spatiotemporal matching problems that are typical for a dynamic setting. Our users are involved in the process of reaching a spatial goal together with the robot in a wayfinding setting. The particular challenge in our framework lies in reaching mutual agreement in relation to the actual surroundings in spite of the fact that humans' and robots' spatiotemporal concepts differ in crucial respects.

Related work also focusing on route descriptions is addressed, for example, by the Instruction-Based Learning group (e.g, Bugmann et al. 2004), and by MacMahon et al. (2006). Our current focus is on a detailed qualitative analysis of the discourse flow between human and robot, using a realistic interaction scenario with uninformed users that is tailored to the actual technological requirements. This particular approach is not to our knowledge adopted elsewhere (though see Gieselmann & Waibel 2005 for a different scenario), but is specifically needed to establish and improve the relationship between implemented functionalities and humans' intuitive reactions at being confronted with an autonomous transportation device. In this paper, we first describe our approach including earlier empirical re-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 103–110,
Antwerp, September 2007.

sults and a sketch of the first version of our dialogue model. Then we present the results of another empirical study testing the model, discuss the ensuing improvements, and conclude by outlining the next steps in this iterative process.

2 Previous work

One of the prominent aims in the SFB/TR 8 Spatial Cognition (Bremen/Freiburg)[1] is to enable smooth and efficient spatiotemporally embedded language-based interaction between humans and robots. For this purpose we explore uninformed users' natural preferences in tasks resembling the future functionalities of our robots in basic respects, coupling technological development with empirical investigations. In the long run, our system will implement ontological knowledge as described in Hois et al. (2007), the development of which is also based on our targeted empirical findings, in addition to a careful examination of the existing literature on spatial language semantics and usage (Tenbrink 2007). Our dialogue system architecture is described in Ross et al. (2005). While the system itself is not restricted to application in a particular robot, we focus here on an application with the autonomous wheelchair "Rolland" (Lankenau & Röfer 2000). In Shi & Tenbrink (forthc.), we describe the first steps in adapting the system for an indoor route description scenario. The main focus in that work is on matching the users' spontaneous utterances with the robot's implemented *conceptual route graph* (Krieg-Brückner & Shi 2006). In the following we summarize the results.

2.1 Empirical results

Our first empirical study was designed to collect spontaneous utterances and examine users' generalized strategies in a scenario resembling the targeted robotic task. Our users were told to move with the robotic wheelchair through an office environment and describe a range of places and locations to the robot. After that, they were asked to instruct the robot to move to one of these places.

From the collected natural language data, we extracted the following potentially problematic consequences of our users' linguistic choices and strategies. Most typically, the utterances may contain a reference to an object or location in the real world that the robot is incapable of resolving. This may be due to the vocabulary available to the robot, to the name tags attached to objects and locations in the robot's internal map, to the user's employment of a different expression than that expected by the robot, or to the robot's inability to establish the exact spatial relationship that the user refers to. The latter point is enhanced by the fact that natural spatial utterances are typically underspecified (Tversky & Lee 1998); they only point to a vague spatial direction that needs to be matched to other knowledge sources, and they often lack information about a required ingredient (such as the relatum). Since the robot's perceptual abilities differ from the human's, there is a high potential for mismatches especially in the (normal) case of underspecification. On top of that, the utterances we collected in our scenario reflect a high degree of uncertainty on the part of the users.

A different problem is that users are unsure about the level of granularity or detail suitable for the instruction. Some instructions directly refer to the goal location, while others only give directional information such as "straight on – to the left". Since the robot has access to higher-level information, this method is not efficient as it leads to a continuous need for interaction. Also, to match instructions with the implemented conceptual route graph, the robot needs information about spatial boundaries of the route segments, which is often not provided, at least not explicitly. The information provided by the users is also often too vague to be matched to the robot's knowledge.

2.2 Dialogue system

The first version of our proposed dialogue model was designed to deal with each of the identified problem areas. In the case of reference resolution problems, underspecification, and missing boundaries, the robot asks for more information. If a conflict between the description and the robot's internal map is detected, the robot makes an assertion to inform the user about this disparity. In case of ambiguities, the robot may provide a suggestion to the user. These ideas were integrated within a dialogue model based on the COnversational Roles model (COR) (Sitter & Stein 1992). Figure 1 shows a depiction of a clarification subdialogue *ask(robot, user),* initiated by the robot, a part of the dialogue

[1] Funding by the DFG is acknowledged. Also, special thanks to Kerstin Fischer who was crucially involved in the preparation of the empirical work reported here.

model. In the diagram *request, reject, accept, suggest and assert* are dialogue acts, while *instruct (user,robot)* is another subdialogue within the dialogue model. Circles represent dialogue states; the marked one is the final state. The subdialogue *instruct(user,robot)* may involve iterative processes such as those described by Clark & Wilkes-Gibbs (1986), in which the agreement on a particular kind of reference may take several turns.

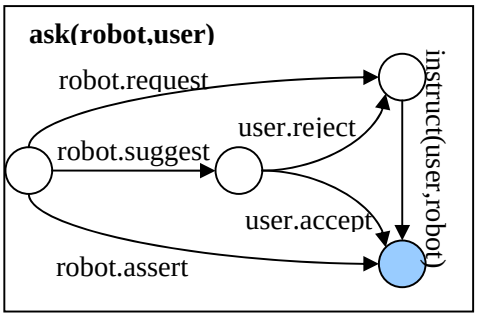


Figure 1 Clarification subdialogue

Our examination of the collected data shows that our formal model should theoretically capture the majority of the potential communication problems identified on the basis of the (monological) first study. In order to account for dynamic dialogue processes, and to put the dialogue model to the test, we conducted a second study in which the robot reacted verbally to the users' utterances. This is particularly important since our cases of clarification relate neither directly to the semantic nor the pragmatic level of understanding (cf. Schlangen 2004), but rather, to the cognitive domain: the robot needs to know precisely how the users' cognitive representation should be matched to its own internal conceptual map. Therefore, standard clarification mechanisms such as various forms of *reprises* or *clausal*, *constituent*, or *lexical* clarifications (Purver et al. 2003) do not readily apply in this particular situationally embedded domain of interaction. Our second study is presented next.

3 Empirical investigation

17 German and 11 English native speakers participated in this experiment. The setting in this second study exactly matched that of the first, except that in this case, in the second (route instruction) phase a human "wizard" was seated behind a screen who triggered prefabricated robotic utterances following a certain schema based on the dialogue model developed before. The schema was devised based on our knowledge about the range of variability in the users' spontaneous utterances, as gained from the first study. The wizard's instructions were as follows: If the user simply states the goal location or reference to a room without providing further information, the robot informs the user that this location is unknown to it, and *requests* further information. If the user provides an underspecified spatial direction such as "then left", the robot *suggests* a precise location to turn according to its internal knowledge, or *requests* clarification in a number of predefined ways, formulated so as to induce the speaker to provide the relevant information on a suitable level of granularity. These reactions account for those cases in which boundaries cannot be inferred from probable interpretations of combined utterances (which should often be possible at least to a certain degree). The wizard could also assume a representation mismatch and react by letting the robot *assert*: "Sorry, this does not match with my internal map". Thus, using a range of preformulated utterances, the wizard was able to produce a reasonably natural dialogue with the user without natural language generation while sounding "automatic" as suitable for the robot. The design was intended to presume a high amount of mismatch and need for clarification (Fischer 2003).

As before, the linguistic data were recorded, transcribed, segmented into TCUs, ("turn constructional units", cf. Selting 2000),[2] and analyzed using the methodology of a detailed qualitative discourse analysis. In particular, since we are interested in the cognitive elements and spatial information content, we categorized the route instruction data with respect to whether each TCU contained:

1. a **directional** or motion-based term, such as "straight on" or "turn left"
2. a reference to a spatial **location**: either a landmark (or sub-goal) or the goal itself, e.g., "go to the office"
3. a reference to a **path entity** ("the hallway").

These distinctions were further examined with respect to whether the landmarks or (sub-)goals in 2. as well as the path entities in 3. were spatially anchored as in "the office *on the left*" or "the first

[2] TCUs are defined on the basis of interactionally relevant completion, taking syntactic, semantic, pragmatic linguistic evidence as well as activity-related factors into account.

hallway *on the left*", and whether they occurred together with a path-describing term such as "*past* the office" or "*down* the hallway". These aspects reflect insights on basic elements of route descriptions (e.g., Denis et al. 1999, Gryl et al. 2002). A specific spatial segment could be described in full by combining all three categories: "go straight on down the hallway in front of you towards the third office on your right". However, most TCUs contain only parts of this information. Other parts may be expressed in or inferable from adjacent TCUs. The component analysis serves here for a first evaluation of the data, though they cannot capture the intricate diversity of the users' distinctions (cf. Klippel et al. in press). More detailed annotations are possible and desirable for our subsequent work, for instance, integrating qualitative and quantitative distance and orientation information (which plays a minor role for the present analysis). In addition to the component-related analysis (section 3.1), we pursued a procedural approach by analysing the development of particular stretches of discourse in detail. In sections 3.2 through 3.4 we present the generalized results of this analysis in relation to the utterance categories presented in 3.1 along with a number of examples.

3.1 Component analysis

In the 11 English and 17 German dialogues, we identified 123 English and 244 German TCUs produced by the user and directed to the robot. In the following, we do not differentiate further between the languages since we do not focus on language differences here. 103 (less than one third) of the TCUs were non-spatial (such as answering "Yes" to a robot's question).

Of the remaining 264 TCUs, about one third (31.44%) were purely direction-based (category 1). Almost half of the 264 TCUs (45.08%) contained mention of a location (category 2); one-third of these were neither spatially anchored nor part of a path-description; rather, they presupposed knowledge of the mentioned place, as in "go to the stugaroom". Almost one-third (29.55%) of the 264 TCUs contained a path entity (category 3), half of these occurring together with a path-describing term, and one-third specifying the path entity spatially. Thus, as with landmark entities, the identity of path entities is typically presupposed. We now turn to a closer look at a number of samples in order to investigate the success of our first dialogue model reflected by our wizard's reactions. Here we did not establish abstract categories for measuring dialogic success, as we felt this would not do justice to the flexibility of the discourse processes.

3.2 Smooth dialogue with clarifications

Some of the dialogues turned out to be entirely unproblematic, they appeared to be completely covered by the dialogue model and did not exhibit any communication problems. Here is an example (D1) (numbers in brackets refer to pauses in seconds; the examples segment the utterances here into idealized turns rather than TCUs).

ROBOT I'm ready.
USER please go to the stugaraum
ROBOT Oh, I'm sorry, I must have missed that location. Can you please explain to me how to get there?
USER leave this room (1) then (2) turn left (2) follow the hallway (2) and then take (1) another left
ROBOT Is this the first possibility to turn?
USER yes (2) then continue left continue down that hallway
ROBOT Up to which point do I go straight?
USER you go straight until the last room on the left side
ROBOT Okay, I think I got it. I am taking you to the Stugaroom now.
USER okay go
ROBOT I am sorry, my batteries are too low. Please recharge them first.

This example shows how the speaker first assumes full knowledge on the part of the robot, by simply referring to the goal location (category 2). The clarification question induces the speaker to provide more specific information structured in segments according to relevant decision points, employing categories 1 and 3. Then the robot suggests that the left turn is the first possibility to turn, which is confirmed. Utterances of category 2 (here: "continue down that hallway") are underspecified with respect to the end boundary, motivating another clarification question by the robot. The user gives the information, and the robot is satisfied.

3.3 Shift of granularity level

Some utterances do not reflect a level of granularity that is suitable for the robot. Our preformulated utterances in some cases successfully motivated a shift in this respect as intended by the form of the robot's requests, as shown in the following example D2.

USER to go to the stugaroom (1) turn (4) 180 degrees (2) exit (1) room three four four zero. turn (1) left (3) go straight (1) ahead

ROBOT Up to which point do I go straight?
USER to the - I don't know (5) go straight (1) turn left to the (3) to the copy room (3)
ROBOT When I'm left of the room with the photocopying machine, what do I do?
USER you go straight until (1) you get to the stugaroom

The user starts out with category 1 descriptions accompanied by hesitations and pauses, omitting information about the boundaries of actions. After the robot's first question the user starts to mention places (category 2) but seems to be uncertain about the correct spatial relations. The robot's second question then seems to encourage the user so that the last part of the route instruction is provided fluently and on a level of granularity more directly matching the robot's internal spatial knowledge (combing categories 1 and 2). – In the following example D3, the robot asserts confusion with respect to its internal map:

ROBOT tut mir leid, aber das passt jetzt gar nicht zu meiner internen Karte. (I'm sorry, but this does not match with my internal map).
USER hm
ROBOT bis wohin fahre ich geradeaus? (Up to which point do I go straight?)
USER bis zur zweiten Linkskurve (until the second left turn)

Here, the robot's request for a spatial goal serves to remedy the confusion and is readily taken up by the user.

3.4 Spatiotemporal congruity

As exemplified by the previous examples, our preformulated answers turned out to be successful whenever they fit sufficiently to the user's descriptions by precisely filling in gaps or underspecifications. However, these clarifications need to be precisely temporally adjusted; as Rieser & Moore (2005:245) point out, "clarification should not be postponed, and immediate, local management of uncertainty is critical." Our example D4 illustrates how a very slight delay may cause confusion and non-clarification of the underspecified item.

USER you turn left, go straight, turn…
ROBOT Up to which point do I go straight?
USER after the copyroom continue straight

The question about when exactly to cease going straight remains unanswered; the user essentially blends the next route segment (which involves going straight) with the robot's question. This confusion is due to the users' choice of rapidly sequencing category 1 utterances that leave boundaries underspecified, which according to the dialogue model triggers the robotic reaction of explicitly asking for boundaries.

Getting back on track is extremely difficult once the discourse flow has been interrupted in an unsuitable way. This may lead to confusions in the represented spatiotemporal sequence, as in the following example D5.

USER wenn wir aus dem Raum A 3440 heraus fahren biegen wir links ab fahren dann geradeaus (coming out of the room A 3440 we turn left and then go straight)
ROBOT wo soll ich abbiegen? (where should I turn?)
USER links. dann an der nächsten Abbiegemöglichkeit nach links biegen wir dort ab (left. then at the next possibility to turn left we turn).

Here, the robot's question is probably intended to refer to the user's description of *going straight*. But the user mentally goes back to the previous expression *turn left*, and then returns immediately to the point where he was interrupted. Matching this kind of non-sequential information to the robot's internal map is certain to cause severe problems. In this case, the user's interpretation of the robot's clarification question could probably have been avoided if the robot had acknowledged the user's description so far (by saying "Okay" prior to asking the question), so that the user knows the question refers to the current route segment rather than a previous one.

3.5 Discussion

Our analysis of utterance components shows that a substantial amount (one third) of speakers' spontaneous route descriptions towards the robot were based purely on spatial directions, rather than providing information about the boundaries of a route segment or the location of a spatial goal or subgoal (landmark). Taken by itself, this result is similar to our monologic study reported in Shi & Tenbrink (forthc.) where the proportion of purely directional TCUs is nearly 40%. Such instructions are informative when given together with additional information in adjacent turns (Tversky & Lee 1998). However, the robot may not always be able to integrate this information suitably, given the implemented features of the conceptual Route Graph. Also, some of our participants relied entirely on underspecified directional information, leading to the need to infer implicit actions (MacMahon et al. 2006). In both cases, a sophisticated dialogue model can support the inference processes by filling in missing information with

respect to both the implemented spatial model, and the real world in which the interaction takes place.

The present Wizard-of-Oz study was purposively designed to assume more mismatches than would normally be the case using any sophisticated spatial language understanding system. Nevertheless, the need for conceptual clarification questions will remain, particularly with increasing spatiotemporal complexity. Such procedures are well known also from human-human interaction (which may be assumed as a "gold standard" for our research), e.g., Filipi & Wales (2004). In the present study, the clarification attempts by the robot worked best for the discourse flow when they could be integrated into the user's current mental representation of the spatial as well as the discourse situation. In other cases, clarification questions could induce spatiotemporal distortions not encountered in our previous monological experiments (Shi & Tenbrink forthc.), thus complicating the dialogue rather than enhancing it.

Robotic requests that include a new starting point, such as "When I am left of the room with the photocopying machine, what do I do?" were taken up easily by the users especially in cases of earlier confusion. To generalize this idea, it is important that the robot informs the user about its current state of knowledge in as much detail as possible, and suggests a solution concerning how to proceed further. This will be specifically helpful in the case of spatiotemporal sequencing confusions. Also, it is important that the robot acknowledges what it has understood so far, to let the user know where exactly there is an information gap that needs to be filled in, and to align the spatiotemporal concepts that the interactants are currently referring to. These results are related to Rieser & Moore's (2005) finding that it is better for systems to ask for confirmation of a hypothesis than to merely signal non-understanding.

In general, our brief investigation of a situated dialogic interaction in which a robot's reactions were simulated shows that requesting clarification about spatial representations is a non-trivial endeavour in which even slight deviations in timing or in confirming common ground may lead to severe distortions (see also Stoia et al. 2006). With a real robotic system, speech recognition problems will complicate the situation considerably (Moratz & Tenbrink 2003), although more standard clarification procedures (Purver et al. 2003, Schlangen 2004) are then applicable to cover some of the problems.

4 Improvement of the dialogue model

Regarding the results of our analysis, the dialogue model used as motivation for the empirical studies (cf. section 2.2) needs to be extended. This concerns, in particular, an improvement of the clarification procedures, the amount of feedback provided by the robot, and a more precise matching process between system knowledge and the linguistic input by the user. Specifically, the precise discourse history is important since specific requests providing information about successfully integrated knowledge are more useful than generic clarification questions, as motivated above. Moreover, the robot's internal map represented as a Conceptual Route Graph and the robot's current position on the map should be used for informing the user in detail about current disparities, in order to classify various requests, and to make precise suggestions (see below). In the former version, this information was only used to detect mismatches, not to inform the user within the clarification subdialogues. To achieve an effective and natural dialogue with users, the dialogue model needs to take account of information from both dialogic and internal sources. Consequently, the first extension of the dialogue model augments it by integrating the dialogue history as well as the internal map with the robot's current position (denoted as [H, M]). The COnversational Roles model is a generic situation-independent dialogue model. Dialogue models based on the COR model cover discourse patterns that are independent of the dialogue context. By integrating the dialogue history as a parameter in the extended dialogue model we add a crucial element from the well-known information state approach (Traum & Larsson 2003) into the dialogue modelling process. As a result the model benefits from both approaches: the flexibility of the information state approach and the well defined structure of the COR based modeling approach.

With respect to the mapping of user utterances to the robot's internal map, the general utterance "this does not match with my internal map" did not seem to be helpful for the users but rather caused confusion (cf. D3). Precise suggestions such as "Is this the first possibility to turn?" seemed more promising (cf. D1). In our improved model, we

substitute the three simple dialogue acts, *request*, *assert* and *suggest* (see Fig. 1 above) by subdialogues. Each subdialogue uses the discourse history and the internal map representation to support detailed classifications. Figure 2 represents the sample subdialogue *request(robot,user)*. First, the robot acknowledges the part of the instruction that it has understood, based on [H, M]. The user can react by rejecting this account and providing a further instruction, in which case the robot does not formulate the request in the intended way. However, if the user does not react or reacts by accepting the robot's description, the robot continues by requesting information about entities, boundaries, or orientations, depending on the current requirements, in a way that is aligned to the users' descriptions as much as possible (using the dialogue history). The dialogue will then continue with the user providing the missing information.

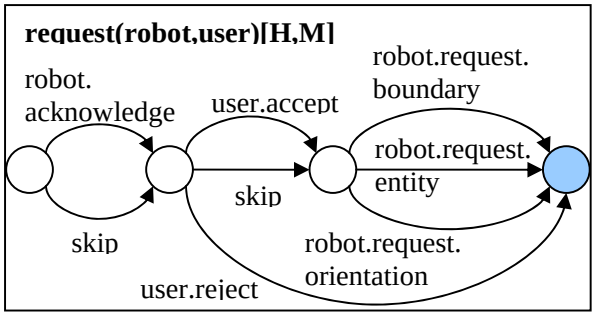


Figure 2 'Request' subdialogue

5 Conclusion and Outlook

We presented a detailed qualitative analysis of a Wizard-of-Oz study specifically tailored to the intended functionalities of the robotic wheelchair Rolland, employing the first version of our dialogue model. Results show that the model is successful in encouraging the user to provide missing information and to use a suitable level of granularity. However, clarification questions from the robot need to be formulated and placed with specific care, as even slight confusions and temporal misplacements of the robot's utterances can lead to severe communication problems and distortions of the user's spatiotemporal representation. Our proposed solution is to let the robot inform the user about its internal state of knowledge in as much detail as possible, and to formulate requests and suggestions in a way that is aligned to the user's descriptions. The next step in our iterative approach is to test this revised model empirically.

The construction of dialogue models is the first step towards the development of dialogue systems based on empirical findings. We are now developing a general approach to specify straightforwardly Recursive Transition Networks in a formal specification language, using the model-checker technique to analyse features, complexity and coverage of dialogue models. Then, dialogue models will be constructed from empirical data by extracting the discourse patterns from annotated dialogues, and analysing the relations between discourse patterns and dialogue models. This procedure will enable us to assert how many dialogues fall into a given dialogue model, which may serve as a basis for evaluating a dialogue's success and efficiency and comparing various instances of dialogue systematically. This approach also supports the mechanical comparison of dialogue models and can thus be used in the dialogue model evaluation process in future iterations.

References

Bugmann G., Klein E., Lauria S. and Kyriacou T. 2004. Corpus-Based Robotics: A Route Instruction Example. In *Proc. IAS-8*, Amsterdam, pp. 96-103.

Clark, B., O. Lemon, A. Gruenstein, E. Owen Bratt, J. Fry, S. Peters, H. Pon-Barry, K. Schultz, Z. Thomsen-Gray, and P. Treeratpituk. 2005. A General Purpose Architecture for Intelligent Tutoring Systems. In J. C. J. van Kuppevelt, L. Dybkjær and N.O. Bernsen (eds.), *Advances in Natural Multimodal Dialogue Systems.* Berlin, Heidelberg: Springer.

Clark, H.H. and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

Denis, M., F. Pazzaglia, C. Cornoldi, and L. Bertolo. 1999. Spatial discourse and navigation: an analysis of route directions in the city of Venice. *Applied Cognitive Psychology* Vol. 13/2, pp. 145 – 174.

Filipi, A. and R. Wales. 2004. Perspective-taking and perspective-shifting as socially situated and collaborative actions. *Journal of Pragmatics, Volume 36, Issue 10*, 1851-1884.

Fischer, K. 2003. Linguistic Methods for Investigating Concepts in Use. In Stolz, T. and Kolbe, K. (eds.), *Methodologie in der Linguistik*. FfM: Lang, 39-62.

Gieselmann, P. and A.Waibel 2005. What makes human-robot dialogues struggle? In *Proc. DIALOR.*

Gryl, A., B. Moulin and D. Kettani. 2002. A conceptual model for representing verbal expressions used in route descriptions. In K. Coventry and P. Olivier (eds.), *Spatial Language: Cognitive and Computational Perspectives.* Dordrecht: Kluwer, pp. 19-42.

Hois, J., M. Wünstel, J.A. Bateman, and T. Röfer. 2007. Dialog-Based 3D-Image Recognition Using a Domain Ontology. In T. Barkowsky, M. Knauff, G. Ligozat, and D. Montello (eds.), *Spatial Cognition V: Reasoning, Action, Interaction*. Berlin: Springer.

Klippel, A., T. Tenbrink, and D. Montello (in press). The Role of Structure and Function in the Conceptualization of Directions. In E. van der Zee and M. Vulchanova (eds.), *Motion Encoding in Language and Space*. Oxford University Press.

Krieg-Brückner, B. and Shi, H. 2006. Orientation Calculi and Route Graphs: Towards Semantic Representations for Route Descriptions. In Raubal, M., Miller, H.J., Frank, A.U., & Goodchild, M.F., *Proc. GIScience 2006, Münster*. Berlin: Springer, pp 234–250.

Kruijff, G.-J., H. Zender, P. Jensfelt, and H.I. Christensen. 2007. Situated Dialogue and Spatial Organization: What, Where... and Why? *International Journal of Advanced Robotic Systems, Special Issue on Human and Robot Interactive Communication* 4, 2.

Kruijff-Korbayová, I., E. Karagjosova, and S. Larsson. 2002. Enhancing collaboration with conditional responses in information-seeking dialogues. In Bos, Foster and Matheson (eds): *EDILOG 2002,* 4-6 September 2002, Edinburgh, UK, pp. 93-100.

Lankenau, A., and Röfer, T. (2000). The Role of Shared Control in Service Robots - The Bremen Autonomous Wheelchair as an Example. In: Röfer, T., Lankenau, A., Moratz, R. (Eds.): *Service Robotics - Applications and Safety Issues in an Emerging Market. Workshop Notes, ECAI 2000*. 27-31.

Lemon, O., A. Bracy, A. Gruenstein, and S. Peters. 2003. An information state approach in a multimodal dialogue system for human-robot conversation. In P. Kühnlein, H. Rieser and H. Zeevat (eds.): *Perspectives on Dialogue in the New Millennium*. Amsterdam: Benjamins, pp 229-242.

MacMahon, M., B. Stankiewicz, and B. Kuipers. 2006. Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions. In *Proc. AAAI-2006*, Boston, MA.

Moratz, R. and Tenbrink, T. 2003. Instruction modes for joint spatial reference between naive users and a mobile robot. Proc. *RISSP 2003*, Special Session on New Methods in Human Robot Interaction, October 8-13, 2003, Changsha, Hunan, China.

Purver, M., Ginzburg, J. & Healey, P. 2003. *On the means for clarification in dialogue*. In Smith, R. and van Kuppevelt, J. (eds.), *Current and New Directions in Discourse and Dialogue*. Kluwer, pp. 235-255.

Rieser, V. and J.D. Moore. 2005. Implications for Generating Clarification Requests in Task-oriented Dialogues. *Proceedings of the 43rd Annual Meeting of the ACL*, pp. 239–246, Ann Arbor.

Ross, R.J., H. Shi, T. Vierhuff, B. Krieg-Brückner, and J.A. Bateman. 2005. Towards Dialogue Based Shared Control of Navigating Robots. In Freksa, C., M. Knauff, B. Krieg-Brückner, B. Nebel, and T. Barkowsky (eds.), *Spatial Cognition IV: Reasoning, Action, Interaction*. Berlin: Springer, pp. 479-500.

Schlangen, D. 2004. Causes and strategies for requesting clarification in dialogue. In *Proc. SIGdial04.*

Selting, M. 2000. The construction of units in conversational talk. *Language in Society 29*, 477–517.

Shi, H. and T. Tenbrink (forthc.) Telling Rolland where to go: HRI dialogues on route navigation. In K. Coventry, T. Tenbrink, and J.A. Bateman (eds.), *Spatial Language and Dialogue.* Oxford University Press.

Sitter, S. and A. Stein. 1992. Modelling the Illocutionary Aspects of Information-Seeking Dialogues. *Information Processing and Management 28*, 124-135.

Spexard, T., S. Li, B. Wrede, J. Fritsch, G. Sagerer, O. Booij, Z. Zivkovic, B. Terwijn, and B. Kröse. 2006. BIRON, where are you? Enabling a robot to learn new places in a real home environment by integrating spoken dialog and visual localization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

Stoia, L., D.K. Byron, D. Shockley, E. Fosler-Lussier. 2006. Sentence Planning for Realtime Navigational Instruction. *Proc. HLT-NAACL 2006*, 157-160.

Tenbrink, T. 2007. *Space, time, and the use of language: An investigation of relationships*. Berlin: Mouton de Gruyter.

Traum, D. and Larsson, S. 2003. The Information State based Approach to Dialogue Management. In R. Smith and J. van Kuppevelt (eds.), *Current and New Directions in Discourse and Dialogue*. Dordrecht: Kluwer, pp. 325-353.

Tversky, B. and P. Lee. 1998. How Space Structures Language. In: C. Freksa, C. Habel & K.F. Wender (eds.), *Spatial Cognition. An Interdisciplinary Approach to Representing and Processing Spatial Knowledge*, pp. 157-175.

Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms

Alexander Gruenstein **Stephanie Seneff**
Spoken Language Systems Group
M.I.T. Computer Science and Artificial Intelligence Laboratory
32 Vassar St, Cambridge, MA 02139 USA
{alexgru,seneff}@csail.mit.edu

Abstract

We present *City Browser*, a web-based platform which provides multimodal access to urban information. We concentrate on aspects of the system that make it compelling for sustained interaction, yet accessible to new users. First, we discuss the architecture's portability, demonstrating how new databases containing *Points of Interest* (*POI*s) may easily be added. We then describe two interface techniques which mitigate the complexity of interacting with these potentially large databases: (1) context-sensitive *utterance suggestions* and (2) *multimodal correction* of speech recognition hypotheses. Finally, we evaluate the platform with data collected from users via the web.

1 Introduction

Multimodal dialogue interfaces, which provide a graphical input and output modality in addition to speech, do not currently tend to be available to the wide audience of users that can be found for more traditional, telephone-based speech-only dialogue systems. At the moment, most development and testing of such systems occurs in the laboratory, under controlled experimental conditions. In this paper, we focus on efforts to convert our restaurant-guide multimodal dialogue system previously described in (Gruenstein et al., 2006; Gruenstein and Seneff, 2006) into *City Browser*, a full-fledged platform for providing urban information multimodally via the world wide web. Because *City Browser* is available via the web, it has millions of potential users on all sorts of Internet-connected devices, which may or may not have keyboards. However, it is a major challenge to actually reach out to these users with an interface that is compelling and capable enough to afford a sustained interaction, yet accessible and intuitive enough to be usable by people who likely have no past experience with multimodal dialogue systems.

In this paper, we identify a core set of capabilities which make *City Browser* compelling as a generic platform for presenting geographic information. The platform provides capabilities to support multimodal exploration of databases containing *Points of Interest*. Exploration is enhanced by allowing users to access information about public transportation, obtain driving directions, and locate addresses on the map. However, over the course of developing the system, it has become apparent that, even as the platform becomes more useful, it also tends to become more difficult to use – a trend often noted by dialogue system designers.

We present two novel user-interface components which are intended to make multimodal dialogue systems more usable in the face of growing complexity. The first is a *suggestions module* which takes advantage of the visual modality to provide high-quality, context-sensitive suggestions to the user about what she can say or do next. The second is a *multimodal error correction framework*, which provides the user with an interactively correctable N-best list of recognizer hypotheses.

Finally, because our interest is in understanding how real users interact with multimodal dialogue

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 111–119,
Antwerp, September 2007.

systems outside of the laboratory environment, we describe our nascent, web-based data collection efforts in which users interact with *City Browser* from their own computers. In particular, we focus our analysis on the response of naive users to the presence of the suggestions module and correctable N-best list.

2 A Platform for Accessing Urban Information

The *City Browser* platform grew out of our work with a multimodal dialogue system which was initially restricted to information about restaurants. The system's overall client-server architecture for speech recognition, linguistic processing, and gesture interpretation has previously been described in detail (Gruenstein et al., 2006). The interface is web based and users need only a web browser equipped with the Java plug-in to access the system. The interaction is centered around a map, as pictured in the screenshot in Figure 5 (in the appendix). In addition to speech input, users can draw on the map, as well as click on displayed *Points of Interest* (*POI*s). An example of a dialogue with the system is given in Figure 1. *City Browser* currently has knowledge of 10 major U.S. metropolitan areas, typically encompassing 20-50 miles around a major city. Harvesting new data from the web for additional metropolitan areas is a semi-automatic process [see (Gruenstein and Seneff, 2006)].

As we observed users interact with our prototype containing only restaurant information, we realized that, while perhaps of some interest, the system was not useful enough for a genuinely compelling interaction. Several frequently requested features were unavailable. In particular, users wanted to find an address on the map and then search near it, obtain driving directions between points of interest, and find out about available public transportation options.

City Browser expands on our previous prototypes by providing these new features. Users can now plan an outing by finding a restaurant of interest and then figuring out how to drive or take public transportation to get there; or they can look for an address, find out how to get there, and look for restaurants nearby. Moreover, when using an address to anchor a search,

S_0:	Welcome to City Browser
U_1:	Show me 32 Vassar Street in Cambridge.
S_2:	Here is 32 Vassar Street in Cambridge. [*shown on map*]
U_3:	Are there any Indian restaurants near here?
S_4:	There are 10 Indian restaurants near 32 Vassar Street in Cambridge. [*results displayed*]
U_5:	What is the nearest subway stop to India Pavilion?
S_6:	Central square station is located at Massachusetts Avenue and Western Avenue, Prospect Street, and Magazine Street. It is on the Red Line.
U_7:	Are there any museums in this area? [*circles an area of the map*]
S_8:	The Museum of Fine Arts is located at 465 Huntington Avenue in Boston.
U_9:	How do I get to there from 77 Massachusetts Avenue in Cambridge?
S_{10}:	Here are driving directions from 77 Massachusetts Avenue in Cambridge to the Museum of Fine Arts. [*gives link to directions*]

Figure 1: An example interaction. U_i: User; S_i: System. Gestures and system actions are bracketed. Some system remarks were shortened for brevity.

it need not be fully specified, as in *32 Vassar Street, Cambridge, Massachusetts*. Instead, it might only be a street name (*Vassar Street in Cambridge*), or just a city (*Cambridge*).

In addition to these core map-based functionalities, it was also apparent that users wanted to be able to access POIs besides restaurants: they were especially interested in POIs such as tourist attractions, banks, parking garages, and gas stations. In order to support this, we have moved from providing access to a restaurant database, to creating a more generic platform for accessing multiple types of POI databases at once. Given a small amount of metadata and a new database of POIs, the language processing components of *City Browser* can easily be updated to support the new database. In particular, support is provided for databases with some or all of the following attributes: (1) *Name* The name of the POI (*e.g.* Museum of Fine Arts), to be used for natural language generation. (2) *Aliases* Alternative names for the POI, for the language model. (3) *Address or Position* The address of the POI, or a location expressed as a latitude and longitude. (4) *Phone Number* The POI's phone number. (5) *URL* Link to a webpage with more information about the object. (6) *Description* A brief description of the POI.

Our currently deployed version of *City Browser* uses these generic database capabilities to provide access to a database of museums. The architecture

also accommodates the subway station databases for providing public transportation information, the geographical database of cities, streets, and neighborhoods, as well as the existing restaurant database.

2.1 Comparison to Similar Systems

The most similar system we are aware of is MATCH (Johnston et al., 2002), which provided extensive multimodal capabilities for accessing urban information. There is significant overlap between *City Browser* and MATCH. For instance, both provide multimodal access to restaurant and public transit information. A major feature of the MATCH system which is lacking in *City Browser* is handwriting recognition; we have not concentrated on this modality, as we do not currently assume our users will have access to a pen-based interface. Another similar interface is AdApt (Gustafson et al., 2000), which provides apartment rental information in downtown Stockholm.

To the best of our knowledge, *City Browser* stands out in that it provides support for POI databases containing thousands of entries, extending throughout a metropolitan area; in particular, the restaurant databases are comparable in size to those of commercially available, web-based restaurant databases. Moreover, *City Browser* supports a multitude of metropolitan *areas*, rather than just one or two *cities*. As we have just described, it also supports the arbitrary addition of new databases of POIs. *City Browser* provides links to driving directions and supports the recognition of arbitrary addresses with any street name in the metropolitan area. Finally, as noted, *City Browser* is fully web-based; and beyond a web browser, requires only the standard Java plug-in to operate. It is the combination of these factors which make *City Browser* uniquely accessible to a potentially large audience, even as a prototype.

3 Suggestions Module: *What Can I Say?*

City Browser is designed to be a highly *user*-driven interface. The task is generally exploratory in nature, rather than transactional, as tends to be more typical for dialogue systems. In testing earlier iterations of the system, we observed that users often had trouble formulating queries "out of thin air," given their lack of experience using such a system. However, given the large bounds of the system's capabilities, it is difficult to imagine a system-directed dialogue, as there are many paths of exploration.

Natural interaction with increasingly complex and intelligent systems is a fundamental challenge in dialogue system research. As capabilities increase, systems often become much more difficult to use. Users can't easily distinguish an error in which an in-domain phrase is misrecognized, from one in which an out-of-domain phrase is spoken. We utilize *City Browser*'s multiple modalities to gain leverage in attacking this problem, by designing a suggestions module which visually provides users with contextually-specific suggestions as to what they might say next at the current point in the dialogue.

On the right-hand side of the GUI, as shown in Figure 5 of the appendix, we show a list of suggested utterances labeled *What Can I Say?*. In fact, these suggestions extend beyond simply what a user can *say*, by indicating gestures that can be made to accompany certain utterances. As in any dialogue system, particular utterances and actions may only be relevant at a given point in the dialogue; to address this, we have created a module which dynamically produces a relevant set of suggested utterances at each new turn in the dialogue. This serves two purposes. First, it allows us to offer relevant suggestions given the current state of the dialogue, tailored specifically to the current context. Second, even as the same templates are used, their content words (such as city names, street names, and cuisine types) are continually changed, giving the user a general impression of the range of the system's knowledge. For instance, a user might be surprised to see a city 20 miles away from the center of the metropolitan area mentioned, indicating that the system has knowledge of many surrounding suburbs.

Dynamic suggestions, which are dependent on the current dialogue state, are instantiated from handcrafted templates and filled in using the current metropolitan region's POI databases. Suggestions are also tailored to any POIs of interest currently visible on the map. Finally, appropriate follow-up queries are inferred from the user's previous utterance. Figure 2 gives an overview showing how the list of suggestions is generated. The different categories of suggestions generated include the following:

Previous Utterance:	*Show me cheap Indian restaurants in Cambridge*
Key-Value Semantics:	clause=*request*, topic=*restaurant*, cuisine=*indian*, price_range=*cheap*,city=*cambridge*
Matching DB entry (subset of attributes shown):	`{q restaurant :name "india castle" :phone "(617) 864-8100" :streetnum "928" :street "massachusetts avenue" :city "cambridge" :state "ma" :cuisine ( "indian" ) :recommendation "recommended" :price_range "low" :neighborhood "harvard square" }`
Random DB entry:	`{q restaurant :name "dakshin" :phone "(508) 424-1030" :streetnum "672" :street "waverly street" :city "framingham" :state "ma" :cuisine ( "indian" ) :recommendation "*none*" :price_range "low" }`

TEMPLATE	***REALIZATION***
Global	
I'm looking for $PRICE_RANGE $CUISINE restaurants on $STREET in $CITY.	I'm looking for *cheap Indian* restaurants on *Waverly* street in *Framingham*.
What is the nearest $SUBWAYNAME station to $ADDRESS?	What is the nearest *T* station to *672 Waverly Street* in *Framingham*?
Are there any $CUISINE restaurants here? [outline a region with the mouse]	Are there any *Indian* restaurants here? [ouline a region with the mouse]
Subsetting	
Show me the $ATTRIBUTE ones.	Show me the *recommended* ones
Tell me about these. [Circle a few $ENTITY_TYPEs with the mouse]	Tell me about these. [Circle a few *restaurants* with the mouse]
Anaphoric	
What's the phone number of $NAME?	What's the phone number of *India Castle*?
Give me driving directions to $NAME from $ADDRESS	Give me driving directions to *India Castle* from *672 Waverly Street* in *Framingham*
Are there any subway stops near $NAME	Are there any subway stops near *India Castle*?
Contrastive	
What about in $CONTRAST_CITY?	What about in *Framingham*?

Figure 2: This figure shows inputs to the suggestions module, examples of each type of template used to create suggestions, and the actual suggestions which are *realized* by combining each template and the input shown at the top. The inputs to the module are (1) the previous utterance and its key-value semantic representation, (2) the database entries which matched that query, and (3) other randomly selected database entries. This information is used to fill in values in each type of template on the left, yielding the realizations of those templates on the right.

Globally relevant suggestions These are utterances which always apply, such as map commands (*pan right* and *zoom in*), queries about addresses, driving directions, public transportation, and points of interest. The POI databases in the current metropolitan region are used to fill in the templates, as shown in Figure 2. The database entries are used in such a way as to guarantee that each suggested utterance, if uttered (and correctly recognized), will actually yield one or more results. This is very important, since, as some users get to know the system, they read the suggestions verbatim. This helps them to verify that the system is working, and to become more comfortable using it. Figure 2 shows examples of different types of suggestions which might be rendered from a single database entry.

Subsetting suggestions There are two forms for specifying *subsetting* suggestions. First, *multimodal* ones, such as *Tell me about these [Circle a few restaurants with the mouse]*, allow the user to zero in on a smaller set. Second, suggestions which subset by *attribute* show how POI properties can be used to narrow down the set, as in, *Show me the highly rated ones*. A rank-ordered list of properties for each POI type is used for this type of narrowing down; for restaurants we use the *price_range* and *recommendation* properties. Any of these properties which were not mentioned in the user's previous utterance are used to create novel suggestions.

Anaphoric suggestions A user will often want to get more information about a particular attribute of either a single POI in focus or a focus set. We produce two types of suggestions for these cases. If a

single entity is currently salient, we offer *anaphoric* suggestions relating to an attribute of that entity, such as *Tell me its phone number*. For a set of entities, we offer suggestions about the properties of individual members, such as *Can you tell me the address of the Museum of Fine Arts?* In addition to querying about a particular property, users may also use one of the in-focus entities as a reference point for searching for something else, as in *Are there any subway stops close to the Royal East?*

Contrastive suggestions A nice aspect of using natural language to access this type of information is that it is quite easy and natural to build on a dialogue by retaining some attributes of a search query and replacing others. For example, if a user has just said *Can you show me the subway stations in Cambridge*, it is quite natural to follow up with a query such as *What about in Brookline?*. We again use the key-value representation of the user's previous utterance, but this time we look for keys which *were* explicitly mentioned by the user. We then produce suggestions in which one or more of these keys is changed to a different value (which, as usual, is drawn from actual database items). In addition, we offer multimodal contrastive suggestions, such as *What about near here? [Click on a point on the map]*.

Our suggestions system resembles somewhat the multimodal help system developed for MATCH (Hastie et al., 2002). MATCH relied on the user explicitly asking for help, while we offer newly updated suggestions at every turn unobtrusively along the side of the screen. While both the MATCH system and our suggestions system are sensitive to the dialogue context, we are more aggressive about actively incorporating information from the various databases used in the system. We are also more sensitive to the semantic content of previous queries, allowing our module to offer more targeted *subsetting* suggestions. On the other hand, the MATCH system's capability to actually demonstrate how to draw or write during a multimodal command is quite useful, and we hope to incorporate a similar capability in the future.

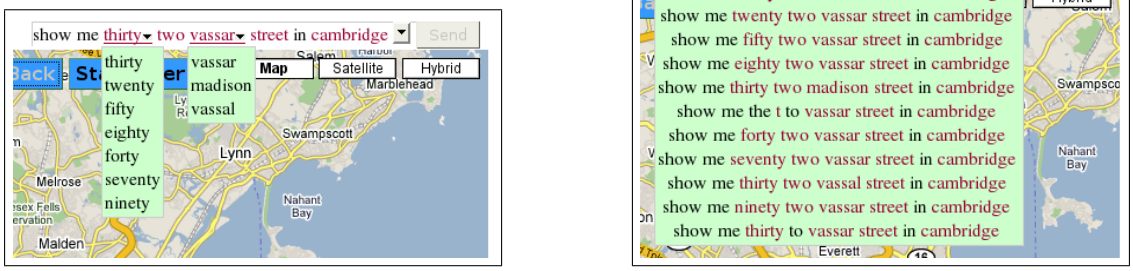
The system can also be seen as providing similar functionality to targeted help systems like those described in (Hockey et al., 2003) and (Gorrell, 2003). However, while these algorithms provide help prompts based on an out-of-domain utterance which was not correctly recognized, the suggestions module described here makes use of the visual modality to try to avoid out-of-domain utterances in the first place. The two approaches could likely be beneficially paired.

4 Multimodal Error Correction

One of the most potentially frustrating aspects of interacting with a dialogue system like *City Browser* is inaccurate speech recognition. Our previous research in this area has focused on dynamic language modeling mechanisms which aim to minimize errors involving proper nouns. Nonetheless, errors arising from the misrecognition of proper nouns are still quite common in *City Browser*, as well as errors having to do with numbers (*e.g.* "thirty" v.s. "fifty"). Other dialogue system designers working in domains with large sets of proper names have also noted this difficulty (Weng et al., 2006).

While extensive research has been performed on multimodal error correction techniques for dictation systems [*e.g.* (Suhm et al., 2001)]– especially with regard to techniques which display alternative hypotheses – we are not aware of dialogue systems which make use of alternatives-based multimodal error correction techniques. Extensive arguments have been made, however, for the potential of multimodal interaction to decrease understanding error rates (Oviatt, 1999).

For *City Browser* we have currently deployed a straightforward mechanism for alternatives-based multimodal error correction, which utilizes the fact that a class n-gram is used as the recognizer's language model – a common mechanism for dialogue system language modeling. Our corrections mechanism presupposes that a large number of errors arise from the misrecognition of content words, rather than the structure of the utterance itself. We display a correctable N-best list which uses semantic knowledge derived from the class n-gram to create alternatives lists. *City Browser* displays the recognizer's top hypothesis, which it has taken to be correct and already responded to, and allows users to correct it in two ways. First, a drop-down menu is available which allows the user to replace the top hypothesis with any of up to 15 of the top hypotheses




```
show me thirty<TENS> two<DIGITS> vassar<STREET> street<STREET_T> in cambridge<CITY>
show me twenty<TENS> two<DIGITS> vassar<STREET> street<STREET_T> in cambridge<CITY>
show me fifty<TENS> two<DIGITS> vassar<STREET> street<STREET_T> in cambridge<CITY>
show me thirty<TENS> two<DIGITS> madison<STREET> street<STREET_T> in cambridge<CITY>
show me the t<SUBWAYNAME> to vassar<STREET> street<STREET_T> in cambridge<CITY>
show me forty<TENS> two<DIGITS> vassar<STREET> street<STREET_T> in cambridge<CITY>
show me seventy<TENS> two<DIGITS> vassar<STREET> street<STREET_T> in cambridge<CITY>
show me thirty<TENS> two<DIGITS> vassar<STREET> street<STREET_T> in cambridge<CITY>
show me twenty<TENS> two<DIGITS> vassal<STREET> street<STREET_T> in cambridge<CITY>
show me ninety<TENS> two<DIGITS> vassar<STREET> street<STREET_T> in cambridge<CITY>
show me thirty<TENS> to vassar<STREET> street<STREET_T> in cambridge<CITY>
...
```

Figure 3: Correctable N-best list. We show a portion of the N-best list generated from the utterance *Show me 32 Vassar Street in Cambridge* along with the drop-down menus available on the user's output. The image on the top-left corner shows what the user sees momentarily during active processing.

which appear on the N-best list. Second, the classes of the language model are leveraged to create potential confusion sets for the members of each class. In particular, whenever a recognition hypothesis is generated by the recognizer, any word or word sequence in the hypothesis which was chosen from one of the language model classes is tagged as such. A separate list is constructed from all words that appear in each class in the top 50 hypotheses on the N-best list. If a class member appears in the top hypothesis, a drop-down menu allows the user to change the value of this class member to that of any other, and then resubmit the altered hypothesis to *City Browser* for processing. Figure 3 shows an N-best list generated by the recognizer, and the resulting drop-down menus which are then available to correct this recognition result.

Typically we expect that this capability would be used primarily to make a single *token replacement* in which one misrecognized class member is replaced with another. We expect that, with less frequency, users will examine the N-best list itself to choose a new *candidate hypothesis*, as this is a more cognitively demanding task. By combining these methods, more complex corrections are possible: a user may first choose a candidate hypothesis with the correct syntactic form, but incorrect class members. They can then perform token replacements to change these class members. This is potentially easier than examining a deep N-best list, as the top-left screenshot in Figure 3 shows. Currently, users can only modify the recognition hypothesis using the provided drop-down menus; though in future work we hope to develop mechanisms which allow the user to type and/or speak to correct parts of the initial hypothesis. However, users are currently free to ignore the correction mechanism by speaking a new utterance.

In our in-lab pilot testing, we realized that users often did not realize that this corrections capability existed, despite help tooltips which point it out. To better advertise the capability, *City Browser* briefly displays each of the available *token replacements* for 1.2 seconds as soon as the recognition hypotheses are available. The benefit here is two-fold. First, it allows the user to easily see if the correct alternative

exists, without having to activate the drop-down list with their mouse. Second, it provides feedback that the system is working even as the input is still being processed and the GUI updated. This both increases the *perceived* responsiveness of the system, and puts the user in a position to detect the error and make the correction more quickly.

5 Preliminary Data Collection Results

We have previously evaluated earlier iterations of the system on several small sets of users using a tablet computer in the laboratory (Gruenstein et al., 2006; Gruenstein and Seneff, 2006). After developing new capabilities, we are now collecting data from users via the web, using their own hardware. We hope that this methodology will enable us to collect a large corpus of data from a wide variety of users, and will allow us to identify issues involved in deploying live dialogue systems.

Subjects are currently being recruited via email lists with an incentive of a $20 Amazon.com gift certificate. Subjects are led through one warm-up task to ensure that their audio set-up is functional, then through 10 scenario-based tasks of generally increasing complexity. The tasks are worded in such a way as to make it difficult to simply "read back" the task description to the system. Several of the tasks are designed to be potentially frustrating if users simply read them back, mentioning concepts that the system does not understand (*e.g.* "highway 93"). This allows us to gather data about how users react when the system encounters out-of-vocabulary words, or concepts the system can't parse or understand. In some cases, it also allows us to collect data about how users might want to interact with the system, if capabilities involving these concepts were available. Figure 6 (in the appendix) shows one of the scenarios used to collect data.

We have transcribed and begun to annotate the data collected from the first 25 users who interacted with the system, and *attempted all*, or *almost all*, scenarios. A total of 1,277 recorded utterances led to recognition hypotheses from these users. The word error rate across all users was 26.0%, similar both to our previous results, and those obtained for small sets of users interacting with MATCH (Bangalore and Johnston, 2004; Johnston et al., 2002) and AdApt (Hjalmarsson, 2002) systems. In order to coarsely gauge the system's performance, we have manually labeled each utterance according to whether the system's response was *entirely correct*, *partially correct* (*e.g.* contained a subset of the information requested), or *incorrect*.

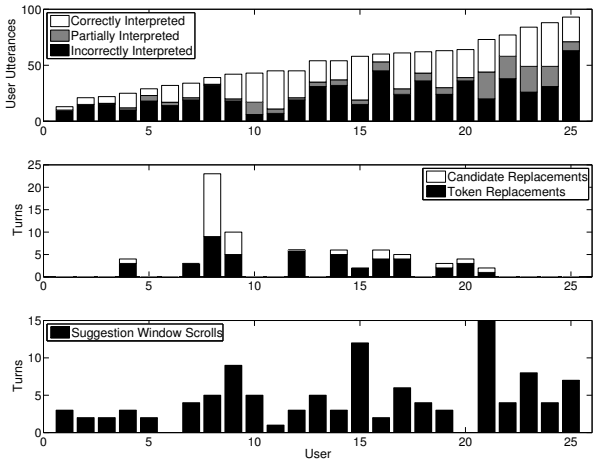


Figure 4: Per-user interaction analysis. Top: correctly, partially correctly, and incorrectly interpreted utterances. Middle: turns with token or candidate corrections. Bottom: turns where suggestions window was scrolled.

Figure 4 shows the number of utterances per user, broken down by the appropriateness of the system's response. Quality of interaction varied quite a bit among users, with some having much more successful interactions than others. We observe that system performance is far from perfect, and are currently further analyzing the causes of the errors. A preliminary analysis shows that audio problems such as inappropriate microphone input level and endpointing errors are responsible for a significant portion of the errors. These types of errors are to be expected when reaching out to a wide range of users using their own hardware, as many users have limited experience using their computer microphone.

We also used log data to glean some knowledge of users' awareness of the N-best corrections capabilities and the suggestions interface. Figure 4 also shows how many times each user used the corrections framework. This is broken down into *token* replacements, in which an individual token (such as a city or street name) was replaced and *candidate* replacements, in which an entirely different candidate

hypothesis was chosen. We found that about half the users (12 of 25) used the corrections capability at least once. In fact, all of these 12 used it more than once.

Finally, to get a very rough idea of whether or not users were at least noticing the suggestions offered by the system, we counted turns in which a user scrolled the suggestions window. The suggestions window can usually fit more than 10 suggestions – depending on screen resolution – when the system first starts. As results are returned, it shrinks to accommodate showing the list of these results, and only the top 5 or so suggestions are usually shown. Users can scroll the window to see all of the currently available suggestions, and this action is logged by the system. Almost all (23 of 25) users scrolled this window at least once; most of them scrolled it during at least several turns. Figure 4 graphs this data. We are encouraged that users are interested enough to scroll the suggestions window, and note that they are likely looking at these suggestions more often than indicated by scrolling, as the top few suggestions (which can be seen without scrolling) are usually intended to be the most relevant to the current context.

6 Summary and Future Work

We have presented *City Browser*, a web-based platform for developing multimodal interfaces which give users access to POI databases. We have shown how *City Browser* can easily accommodate new POI databases. In addition, we have described two aspects of the system which make it easier for users to interact with the unfamiliar technology: a suggestions module and a multimodal error correction interface technique. Finally, we present a preliminary evaluation of these features using data collected from users via the web, using their own computer hardware. We show that users generally do discover and make use of the suggestions feature, while about half use the correctable N-best list.

In the future, we plan to expand the capabilities of *City Browser* based on observations of user interactions and their feedback. We are particularly interested in improving both the suggestions generating and multimodal error correction modules. For example, we believe that a full-blown semantic representation of utterances could be incorporated to allow users to correct structured representations of *City Browser* interpretations rather than text strings.

Acknowledgements

Thanks goes to Chao Wang for input on a draft, Sean Liu for transcription and GUI work, and Liz Murnane for GUI work. This research is sponsored by the T-Party Project, a joint research program between MIT and Quanta Computer Inc., Taiwan.

References

S. Bangalore and M. Johnston. 2004. Robust multimodal understanding. In *Proc. of ICASSP*.

G. Gorrell. 2003. Recognition error handling in spoken dialogue systems. In *Proc. of 2nd International Conference on Mobile and Ubiquitous Multimedia*.

A. Gruenstein and S. Seneff. 2006. Context-sensitive language modeling for large sets of proper nouns in multimodal dialogue systems. In *Proc. of IEEE/ACL 2006 Workshop on Spoken Language Technology*.

A. Gruenstein, S. Seneff, and C. Wang. 2006. Scalable and portable web-based multimodal dialogue interaction with geographical databases. In *Proc. of INTERSPEECH*.

J. Gustafson, L. Bell, J. Beskow, J. Boye, R. Carlson, J. Edlund, B. Granström, D. House, and M. Wirén. 2000. AdApt a multimodal conversational dialogue system in an apartment domain". In *Proc. of ICSLP*.

H. Hastie, M. Johnston, and P. Ehlen. 2002. Context-sensitive Help for Multimodal Dialogue. In *Proc. of ICMI*, pages 93–98.

A. Hjalmarsson. 2002. Evaluating AdApt, a multi-modal conversational dialogue system using PARADISE. Master's thesis, KTH, Stockhom, Sweden.

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 user's performance. In *Proc. EACL*.

M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. Walker, S. Whittaker, and P. Maloor. 2002. MATCH: An architecture for multimodal dialogue systems. In *Proc. of ACL*.

S. Oviatt. 1999. Mutual disambiguation of recognition errors in a multimodel architecture. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 576–583.

B. Suhm, B. Myers, and A. Waibel. 2001. Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8(1):60–98.

F. Weng et al. 2006. CHAT: A conversational helper for automotive tasks. In *Proc. of INTERSPEECH*.

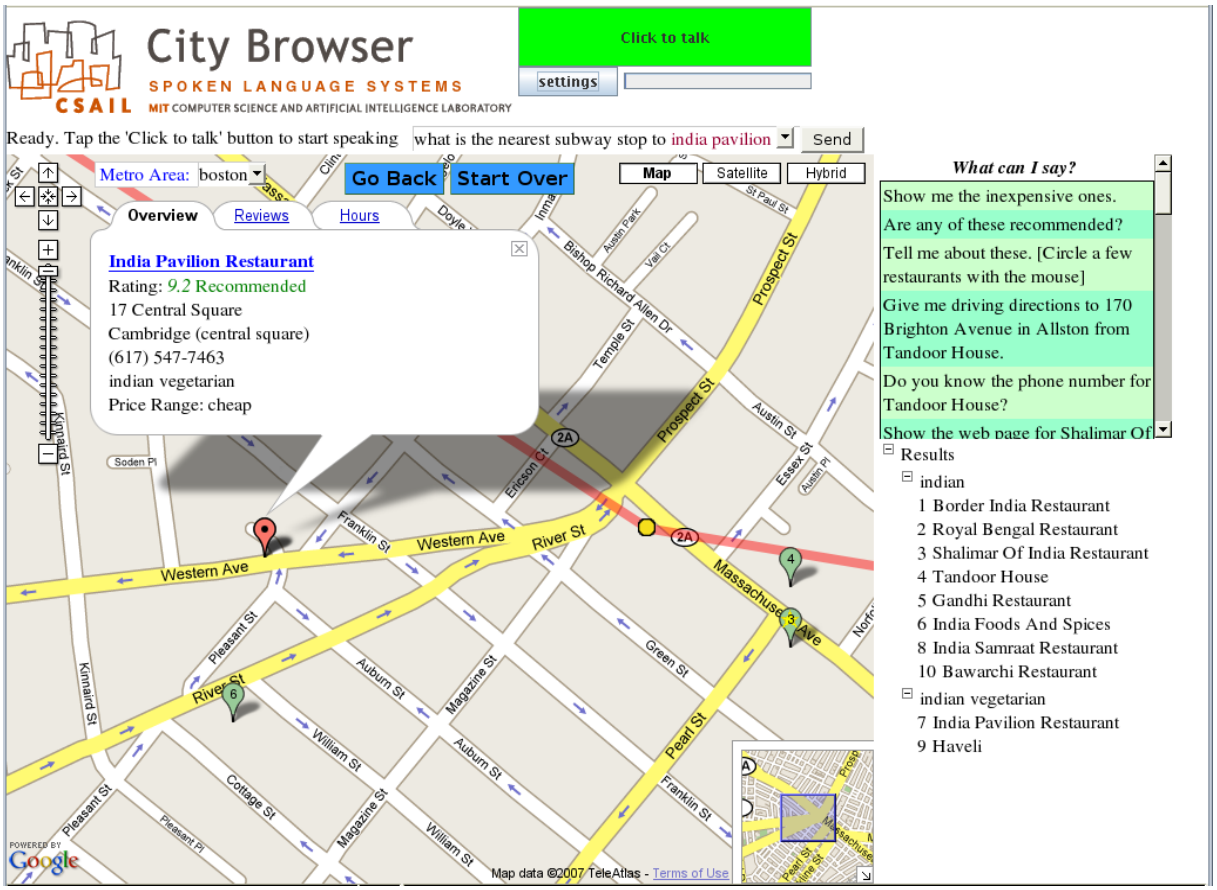


Figure 5: Screenshot of the *City Browser* interface running inside a web browser. At the top, there is a large button that the user presses to start speaking, with a bar underneath which moves as users speak. Immediately below the bar is the top recognition hypothesis for the user's previous utterance, shown as a correctable N-best list. In the upper right corner are the current suggestions of what to say next; below that is a list of restaurants recently returned in an earlier query. These restaurants are shown as the numbered markers on the map at the center. There is also a portion of the overlayed subway map, shown as the line passing through the shaded circle, which has been displayed in response to the user's current query. The shaded circle on that line marks the nearest subway station to the restaurant under discussion, and can be clicked for more information. In the top left corner of the map is a control which allows the user to change the current metropolitan area. To the right of it, are buttons which allow the user to *go back* (undo the previous utterance) and *start over*. The standard Google Maps controls are also overlayed on the map for zooming, panning, and switching to satellite or hybrid view.

You have a friend visiting who wants to go to a couple of different museums in Boston while she's here. She's a sports nut, so you plan to take her to the Sports Museum near the Fleet Center in the morning. Then, you'd like to take her to the Museum of Fine Arts in the afternoon. You are planning on taking the subway to get around starting in Kendall Square. Figure out a plan for doing this. Also, you'd like to find a nice place to eat lunch within walking distance of the Sports Museum, and an Italian place for dinner that is not too far from the Museum of Fine Arts.

Figure 6: Example data-collection scenario

Analysis of User Reactions to Turn-Taking Failures in Spoken Dialogue Systems

Mikio Nakano* **Yuka Nagano**** **Kotaro Funakoshi*** **Toshihiko Ito****
Kenji Araki** **Yuji Hasegawa*** **Hiroshi Tsujino***
*Honda Research Institute Japan Co. Ltd.
8-1 Honcho, Wako, Saitama 359-0188, Japan
**Graduate School of Information Science and Technology, Hokkaido University
Kita-14, Nishi-9, Kita-ku, Sapporo, Hokkaido 060-0814, Japan
{nakano, funakoshi, yuji.hasegawa, tsujino}@jp.honda-ri.com
{calico,t-itoh,araki}@media.eng.hokudai.ac.jp

Abstract

This paper presents the results of an analysis of user reactions towards system failures in turn-taking in human-computer dialogues. When a system utterance and a user utterance start with a small time difference, the user may stop his/her utterance. In addition, when the user utterance ends soon after the overlap starts, the possibility of the utterance being discontinued is high. Based on this analysis, it is suggested that the degradation in speech recognition performance can be predicted using utterance overlapping information.

1 Introduction

Many kinds of spoken dialogue systems have been developed in the last two decades. Most previous systems employed a fixed turn-taking strategy, that is, they take a turn when the user puts a certain length of pause after his/her utterances, and they release the turn immediately when the user barges in on a system utterance. In order to improve the usability of spoken dialogue systems, the turn-taking strategy needs to be more flexible.

Thus far, there have been several approaches to this problem. Some methods try to decide when to take a turn based on not only the length of pause but also the content and prosody of the user utterance [e.g., (Sato et al., 2002; Ferrer et al., 2003; Schlangen, 2006)]. Other methods try to decide how to appropriately react to the user barge-in utterances, not just simply stopping whenever a barge-in utterance is detected [e.g., (Ström and Seneff, 2000; Rose and Kim, 2003)].

Despite these efforts, achieving appropriate turn-taking is still difficult. The features used by these methods are not always perfectly obtained. In addition, even humans cannot sometimes decide whether the system should take a turn or not (Sato et al., 2002).

Consequently, in addition to efforts towards improving turn-taking, we need to find a way to make the system cope with turn-taking errors. As a first step, we investigated how users behave when the system made mistakes in turn-taking. We have found that users tend to stop their utterances in certain situations. We expect this to be useful in avoiding misunderstanding caused by speech recognition errors of such discontinued utterances.

2 Analysis of User Reactions to Turn-Taking Failures

2.1 Dialogue Data

We analyzed two sets of human-system dialogue data using the following two different dialogue systems in Japanese. One was a car-rental reservation dialogue system in which the user could make a reservation for renting a car by specifying the date, hour, and locations for rental and return, along with the car type. The other was a video recording system in which the user could set the date, time, channel, and recording mode (long play or short play) for recording a TV program.

Both systems performed frame-based dialogue management. They employed the Julian speech rec-

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 120–123,
Antwerp, September 2007.

ognizer directed by network grammars (Kawahara et al., 2004) with its attached acoustic models. The vocabulary size for speech recognition was 225 words for the car-rental reservation system and 198 words for the video recording system. These systems also employed NTT-IT Corporation's FineVoice speech synthesizer. When collecting the data, a microphone and headphones were used. For each dialogue, the microphone input and the system output were recorded in a stereo file.

The contents of the data sets are as follows:

- Set C: (Car-rental reservation)

  Each of the 23 subjects (12 males and 11 females) engaged in 8 dialogues (total 184 dialogues). In each dialogue, users tried to make one reservation. 134 dialogues were successfully finished within 3.5 minutes, 38 failed, and 12 were aborted because of a system trouble.

- Set V: (Video recording reservation)

  This consists of 117 dialogues (9 dialogues by each of the 13 subjects (9 males and 4 females)). These subjects are different from the subjects for Set C. In each dialogue, the user tried to set the timer to record two programs. In 41 dialogues, the user successfully set up the recordings for two programs within 3 minutes. In 36 dialogues, the user set up only one of the programs. In 34 dialogues, the user could not set up the recordings, and 6 were aborted.

Both systems had variations in dialogue and turn-taking strategies so that a variety of dialogues were recorded. Thresholds for confidence scores for generating confirmation requests were changed, parameters for speech interval detection were changed, and whether the system stopped its utterances when the user barged in was changed. For each subject, different strategies were used for different dialogues. We will not explain these variations in detail since, as we will explain later, we focused on the phenomena of turn-taking failures rather than the causes of them.

After collecting data, both user and system utterances were transcribed as pronounced. Utterance segmentation was done manually based on pauses longer than 300ms, by using an annotation tool.

set \ case	(o1)	(o2)	(o3)	total
C	67	446	7	520
V	46	202	1	249

(o1) The start time of the user utterance is between the start and end times of a system utterance.
(o2) The start times of one or more system utterances are between the start and end time of the user utterance.
(o3) Both (o1) and (o2) occur.

Table 1: Frequencies of user utterances overlapping with system utterances.

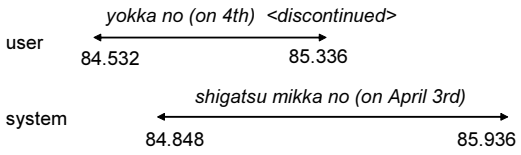


Figure 1: Example discontinuation with overlap.

The timestamps of each speech segment indicate the points in time from the start of the stereo file. Below we simply call these speech segments *utterances*. The total numbers of the user utterances and system utterances in Set C are respectively 3,364 and 5,157 and, in Set V they are 2,521 and 4,522.

2.2 Utterance Overlaps

As Raux et al. (2006) reported, there are several kinds of system turn-taking failures. The system sometimes barges in to a user utterance, and sometimes fails to take a turn. These failures are caused by several reasons, such as errors in speech interval detection, and misrecognitions of the user's intention to release a turn.

In this paper, we focus only on failures that result in overlaps between user and system utterances. We have not investigated the reason for the failure; but instead of that, we analyzed the overlapping phenomena that often occurred when the system made mistakes in turn-taking, because the goal of the analysis is not to improve turn-taking, but to find a way to recover from turn-taking failures. Table 1 shows the frequencies of user utterances overlapping system utterances.

2.3 Discontinuations

In this paper, we call utterances stopped in the middle for any reason *discontinuations*. We found that user utterances overlapping with system utterances

set	all utterances			discontinuations		
	IG	OOG	ALL	IG	OOG	ALL
C	2,662	702	3,364	9	78	87
	22.75	74.05	40.23	12.00	66.97	63.13
V	1,599	922	2,521	2	46	48
	13.08	73.89	39.69	0.00	90.43	87.39

IG means in-grammar utterances, and OOG means out-of-grammar utterances. (upper: # of utterances, lower: word error rate (%))

Table 2: Speech recognition results for all utterances and discontinuations.

d (s)	$-\infty$ – -0.4	-0.4 – -0.2	-0.2 – 0.0	0.0 – 0.2	0.2 – 0.4	0.4 – 0.6	0.6 – 1.0	1.0 – ∞
C	2/45	0/7	4/22	15/43	11/56	3/29	4/34	22/284
V	0/17	0/9	10/21	16/57	6/48	3/27	1/12	2/58

(# of discontinuations)/(# of overlapped user utterances)

Table 3: Frequency of discontinuations depending on the start time difference d.

c (s)	0.0 – 0.1	0.1 – 0.2	0.2 – 0.3	0.3 – 0.4	0.4 – 0.5	0.5 – 0.6	0.6 – 0.8	0.8 – 1.0	1.0 – ∞
C	1/50	7/44	10/67	12/66	15/52	4/36	4/75	4/45	4/85
V	1/19	4/19	9/30	13/28	2/17	3/16	2/22	0/17	4/81

(# of discontinuations)/(# of overlapped user utterances)

Table 4: Frequency of discontinuations depending on c (the length of user utterance after the overlapping starts)

are more likely to be discontinuations. Discontinuations are expected to be difficult for speech recognition mainly because they are not grammatical and include word fragments. So detecting and ignoring them would improve speech understanding. We therefore focus on analyzing discontinuations. Figure 1 shows an example of discontinuations in a car-rental reservation dialogue.

We annotated discontinuations by listening to only the user-speech channel of the stereo files. In set C, 87 utterances are discontinuations, and, in set V, 48 are discontinuations. Of these, 61 and 38 have overlaps with system utterances.

To investigate the speech recognition performance on the discontinuations, we used the same network grammar as the spoken dialogue system used in the data collection. Note that, since user speech segments are made from the timestamps in the transcriptions, they are different from those recognized at the time of data collection. As shown in Table 2, discontinuations include out-of-grammar utterances, so the word error rates are very high.[1]

2.4 Relationship between Discontinuations and Turn-Taking

One way to detect discontinuations that might be effective is to use prosodic information (Liu et al., 2003). Since prosody recognition is not yet perfect, however, it is worth exploring other methods. We therefore investigated in which turn-taking situations discontinuations are likely to exist.

Discontinuations are likely to occur when the start time of the user and system utterances are close. Table 3 shows the relationships of the frequencies of discontinuations in the overlapping user utterances depending on the start time difference d. Here, the start time difference d is defined as follows:

$$d = st(u) - st(s),$$

where $st(i)$ means the start time of utterance i, u is a user utterance and s is the first system utterance among the system utterances overlapping u. We found that people tend to stop their own utterances when d is between -0.2s to 0.4s. When d is larger than 0.4s, the user has already spoken for a while so he/she might try to finish the utterance.

Next, we investigated the end time of the overlapped user utterances, because discontinuations can be expected to occur soon after the overlapping starts. Table 4 shows the frequencies of discontinuations depending on the length of the user utterance after the overlapping starts. This is defined as c in the following formula:

$$c = \begin{cases} et(u) - st(u) & (\text{cases (o1) and (o3) in Table 1}) \\ et(u) - st(s) & (\text{case (o2) in Table 1}), \end{cases}$$

where $et(i)$ means the end time of utterance i. As we expected, when c is between 0.1s and 0.6s, the

[1] The word error rates for the out-of-grammar utterances is very high for the following reason. We transcribed the user utterances without word boundaries because it is not easy to consistently determine word boundaries for Japanese. We used a morphological analyzer to split these transcriptions into words to obtain references for computing speech recognition accuracy. This process tended to produce one-syllable out-of-vocabulary words. Therefore the references include a greater number of out-of-vocabulary words.

Set C

d (s) \ c (s)	0.0 – 0.1	0.1–0.6	0.6 – ∞
$-\infty$ – -0.2	0/0	2/12	0/40
-0.2 – 0.4	1/6	**24/62**	5/53
0.4 – ∞	0/44	22/191	7/112

Set V

d (s) \ c (s)	0.0 – 0.1	0.1–0.6	0.6 – ∞
$-\infty$ – -0.2	0/0	0/11	0/15
-0.2 – 0.4	1/2	**26/52**	5/72
0.4 – ∞	0/17	5/47	1/33

(# of discontinuations)/(# of overlapped user utterances)

Table 5: Frequency of discontinuations depending on c and d.

	Situation S			Other overlapping utterances		
set	IG	OOG	ALL	IG	OOG	ALL
C	20	42	62	285	173	458
	16.67	107.89	78.57	12.72	66.31	35.36
V	13	39	52	97	100	197
	9.52	122.73	86.15	8.44	75.06	43.14

(upper: # of utterances. lower: word error rate (%).)

Table 6: Speech recognition performance for utterances in Situation S and other cases.

user utterances are more likely to be discontinuations than other cases.

From the above analysis, the possibility that a discontinuation occurs is high when d is between -0.2s and 0.4s and c is between 0.1s and 0.6s. We call this situation, *Situation S*. Table 5 shows the frequencies of discontinuations depending on the combinations of d and c.

2.5 Predicting Speech Recognition Performance Degradation

Since discontinuations occur more frequently in Situation S than other cases, speech recognition performance would be degraded in Situation S. Table 6 shows these results. This suggests that the overlapping information can be used for predicting speech recognition performance degradation.

3 Concluding Remarks

This paper presented our preliminary analysis on user reactions to system failures in turn-taking in human-computer dialogues. We found that discontinuations are likely to occur more frequently at the overlapping utterances caused by turn-taking failure. We specified situations where user discontinuations frequently occur. It is suggested that the degradation in speech recognition performance can be predicted using utterance overlapping information. This is expected to be useful for avoiding misunderstanding.

We are planning to conduct more detailed analyses on discontinuations, such as their relationship with the subjects and the dialogue and turn-taking strategy of the system. We also plan to investigate changes in speech recognition performance when statistical language models are employed.

References

Luciana Ferrer, Elizabeth Shriberg, and Andreas Stolcke. 2003. A prosody-based approach to end-of-utterance detection that does not require speech recognition. In *Proc. ICASSP-2003*.

Tatsuya Kawahara, Akinobu Lee, Kazuya Takeda, Katsunobu Itou, and Kiyohiro Shikano. 2004. Recent progress of open-source LVCSR engine Julius and Japanese model repository. In *Proc. Interspeech-2004 (ICSLP)*, pages 3069–3072.

Yang Liu, Elizabeth Shriberg, and Andreas Stolcke. 2003. Automatic disfluency identification in conversational speech using multiple knowledge sources. In *Proc. Eurospeech-2003*, pages 957–960.

Antoine Raux, Brian Langner, Dan Bohus, Alan W. Black, and Maxine Eskenazi. 2006. Doing research in a deployed spoken dialog system: One year of let's go! public experience. In *Proc. Interspeech-2006 (ICSLP)*, pages 65–68.

R.C. Rose and Hong Kook Kim. 2003. A hybrid barge-in procedure for more reliable turn-taking in human-machine dialog systems. In *Proc. ASRU-03*, pages 198–203.

Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyoaki Aikawa. 2002. Learning decision trees to determine turn-taking by spoken dialogue systems. In *Proc. 7th ICSLP*, pages 861–864.

David Schlangen. 2006. From reaction to prediction: Experiments with computational models of turn-taking. In *Proc. Interspeech-2006 (ICSLP)*, pages 2010–2013.

Nikko Ström and Stephanie Seneff. 2000. Intelligent barge-in in conversational systems. In *Proc. 6th ICSLP*.

Comparing Spoken Dialog Corpora Collected with Recruited Subjects versus Real Users

Hua Ai[1], **Antoine Raux**[2], **Dan Bohus**[3*], **Maxine Eskenazi**[2], **Diane Litman**[1,4]
[1] Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260, USA
[2] Language Technologies Institute, Carnegie Mellon University, Pittsburgh PA, 15213, USA
[3] Computer Science Department, Carnegie Mellon University, Pittsburgh PA, 15213, USA
[4] Dept. of Computer Science & LRDC, University of Pittsburgh, Pittsburgh, PA 15260, USA
hua@cs.pitt.edu, {antoine,dbohus,max}@cs.cmu.edu, litman@cs.pitt.edu

Abstract

Empirical spoken dialog research often involves the collection and analysis of a dialog corpus. However, it is not well understood whether and how a corpus of dialogs collected using recruited subjects differs from a corpus of dialogs obtained from real users. In this paper we use Let's Go Lab, a platform for experimenting with a deployed spoken dialog bus information system, to address this question. Our first corpus is collected by recruiting subjects to call Let's Go in a standard laboratory setting, while our second corpus consists of calls from real users calling Let's Go during its operating hours. We quantitatively characterize the two collected corpora using previously proposed measures from the spoken dialog literature, then discuss the statistically significant similarities and differences between the two corpora with respect to these measures. For example, we find that recruited subjects talk more and speak faster, while real users ask for more help and more frequently interrupt the system. In contrast, we find no difference with respect to dialog structure.

1 Introduction

Empirical approaches have been widely used in the area of spoken dialog systems, and typically involve the collection and use of dialog corpora. For example, data obtained from human users during Wizard-of-Oz experiments (Okamoto et al., 2001), or from interactions with early system prototypes, are often used to better design system functionalities. Once obtained, such corpora are often then used in machine learning approaches to tasks such as dialog strategy optimization (e.g. (Lemon et al., 2006)), or user simulation (e.g. (Schatzmann et al., 2005)). During system evaluation, user satisfaction surveys are often carried out with humans after interacting with a system (Hone and Graham, 2000); given a dialog corpus obtained from such interactions, evaluation frameworks such as PARADISE (Walker et al., 2000) can then be used to predict user satisfaction from measures that can be directly computed from the corpus.

Experiments with *recruited subjects* (hereafter referred to as *subjects*) have often provided dialog corpora for such system design and evaluation purposes. However, it is not well understood whether and how a corpus of dialogs collected using subjects differs from a corpus of dialogs obtained from *real users* (hereafter referred to as *users*). Selecting a small group of subjects to represent a target population of users can be viewed as statistical sampling from an entire population of users. Thus, (1) a certain amount of data is needed to draw statistically reliable conclusions, and (2) subjects should be randomly chosen from the total population of target users in order to obtain unbiased results. While we believe that most spoken dialog subject experiments have addressed the first point, the second point has been less well addressed. Most academic and many industrial studies recruit subjects from nearby resources, such as college students and colleagues, who are not necessarily representative of the target

[*] Currently at Microsoft Research, Redmond, WA, USA

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 124–131,
Antwerp, September 2007.

users of the final system; the cost to employ market survey companies to obtain a better representation of the target user population is usually beyond the budget of most research projects. In addition, because subjects have either volunteered or are compensated to participate in an experiment, their motivation is often different from that of users. In fact, a recent study comparing spoken dialog data obtained in usability testing versus in real system usage, found significant differences across conditions (e.g., the proportion of dialogs with repeat requests was much lower during real usage) (Turunen et al., 2006).

Our long term goal is to understand the differences that occur in corpora collected from subjects versus users, and to see, if indeed such differences do exist, their impact on empirical dialog research. In this paper we take a first step towards this goal, by collecting and comparing subject and user dialogs with the Let's Go bus information system (Raux et al., 2005). In future work, we plan to investigate how differences found in this paper impact the utility of using subject corpora for tasks such as building user simulations to optimize dialog strategies.

Because there are no well-established standards regarding best practices for spoken dialog experiments with subjects, we first surveyed recent approaches to collecting corpora in laboratory settings. We then used these findings to collect our subject corpus using a "standard" laboratory setting, by adopting the practices we observed in a majority of the surveyed studies. To obtain our user corpus, we collected all dialogs to Let's Go during its deployed hours, over a four day period. Once collected, we quantitatively characterized the two collected corpora using previously proposed measures from the spoken dialog literature. Our results reveal both similarities and differences between the two corpora. For example, we find that while subjects talk more and speak faster, users more frequently ask for help and interrupt the system. In contrast, the dialogs of subjects and users exhibit similar dialog structures.

In Section 2, we describe the papers we surveyed, and summarize the common practices we observed for collecting dialog corpora using subjects. In Section 3, we introduce the Let's Go spoken dialog system, which we use to collect both our subject and user corpora. In Section 4, we describe the specific in-lab experiment we conducted with recruited subjects. We then introduce the evaluation measures used for our corpora comparisons in Section 5, followed by a presentation of our results in Section 6. Finally, we further discuss and summarize our results in Section 7.

2 Literature Review

In this section we survey a set of spoken dialog papers involving human subject experiments (namely, (Allen et al., 1996), (Batliner et al., 2003), (Bohus and Rudnicky, 2006), (Giorgino et al., 2004), (Gruenstein et al., 2006), (Hof et al., 2006), (Lemon et al., 2006), (Litman and Pan, 2002), (Möller et al., 2006), (Rieser et al., 2005), (Roque et al., 2006), (Singh et al., 2000), (Tomko and Rosenfeld, 2006), (Walker et al., 2001), (Walker et al., 2000)), in order to define a "standard" laboratory setting for use in our own experiments with subjects. We survey the literature from four perspectives: subject recruitment, experimental environment, task design, and experimental policies.

Subject Recruitment. Recruiting subjects involves deciding who to recruit, where to recruit, and how many subjects to recruit. In the studies we surveyed, the number of subjects recruited for each experiment ranged from 10 to 72. Most of the studies recruited only native speakers. Half of the studies clearly stated that the subjects were balanced for gender. Most of the studies recruited either college students or colleagues who were not involved in the project itself. Only one study recruited potential system users by consulting a market research company.

Experimental Environment. Setting up an experimental environment involves deciding where to carry out the experiment, and how to set up this experimental environment. The location of the experiment may impact user performance since people behave differently in different environments. This factor is especially important for spoken dialog systems, since system performance is often impacted by noisy conditions and the quality of the communication channel. Although users may call a telephone-based dialog system from a noisy environment using a poor communication channel (e.g., by using a cell phone to call the system from the street), most experiments have been conducted in a quiet in-room lab setting. Subjects typically talk to the system directly

via a high-quality microphone, or call the system using a land-line phone. Among the studies we looked at, only 2 studies had subjects call from outside the lab; another 2 studies used driving simulators. One study changed the furniture arrangement in the lab to simulate home versus office scenarios.

Task Design. Task design involves specifying whether subjects should use the dialog system to accomplish specific tasks, and if so, defining those tasks. All except one study asked subjects to finish a set of fixed tasks in a predefined order. In one study, subjects were asked to do 2 open tasks after a series of 7 fixed tasks. In another study, where the system provided restaurant information, the researchers asked the subjects to ask about information for at least 4 restaurants, but did not specify the restaurant names. The number of tasks in these studies ranged from 2 to 10.

Experimental Policies. Experimental policies involves specifying additional procedures for running subjects during the course of the experiment. None of the studies mentioned that they controlled their experiments by setting any time limits for the subjects. Only 2 studies clearly declared that subjects were told to read some instructions before the experiment started. While two studies motivated subjects by offering a bonus upon task completion, the majority of studies paid subjects on the basis of their participation alone.

In summary, a standard way to carry out human subject experiments with spoken dialog systems (where we use standard to mean that the practice occurred in a majority of the papers surveyed), is as follows: (1) Recruit at least 10 subjects who are college students or colleagues who are native English speakers, trying to balance between genders; (2) Ask the subjects to come to the lab to generate their dialogs with the system; (3) Set up several tasks for the subjects, and ask them to complete these tasks in a certain order; (4) Pay the subjects for their participation, without a bonus. As will be seen in Section 4, we follow these practices when designing our own experiment.

3 System Description

The study described in this paper was conducted in the Let's Go Lab which uses the Let's Go bus information system, a telephone-based dialog system that provides schedule information for buses in the Pittsburgh area (Raux et al., 2005). The Lab is a service run by the creators of Let's Go to allow other researchers access to their numerous users to run experiments. When the customer service line of the Port Authority of Allegheny County (which manages buses in Pittsburgh) is not staffed by operators (i.e. from 7pm to 6am on weekdays and 6pm to 8am on weekends), callers are redirected to Let's Go. In the Let's Go Lab, experimenters typically run offline and/or in-lab experiments first, then evaluate their approach using the live system.

S: Welcome to the CMU Let's Go bus information system. To get help at any time, just say Help or press zero. What can I do for you?
U: UH I NEED TO GET FROM FIFTH AND BIGELOW
S: Leaving from Fifth and Bigelow . Did I get that right?
U: YES
S: Right. What is your destination?
U: AIRPORT
S: Going to airport . Did I get that right?
U: YES
S: Okay. When are you going to take that bus?
U: NINE P UH... SEVEN P M
S: Leaving at 7 p.m. . Is this correct?
U: YES
S: Right. Just a minute. Let me check that for you. There is a 28X leaving Fifth Avenue at Bigelow ...

Figure 1: Example Dialog with Let's Go.

An example dialog with Let's Go (obtained from a subject) is shown in Figure 1. The interaction with the system itself starts with an open prompt ("What can I do for you?") followed by a more directed phase where the system attempts to obtain the missing information (origin, destination, travel time, and optionally route number) from the user. Finally, the system provides the best matching bus number and time, at which point the user has the possibility of asking for the next/previous buses.

Let's Go is based on the Olympus architecture developed at CMU (Bohus et al., 2007). It uses the RavenClaw dialog manager (Bohus and Rudnicky, 2003), the PocketSphinx speech recognition

High-level dialog features	
number of turns	turn
duration of dialog	dialogLen
total words per user turn	U_word
number of dialog acts per system/user turn	U_action, S_action
ratio of system and user actions	Ratio_action
Dialog style/cooperativeness	
dialog acts	S_requestinfo, S_confirm, S_inform, S_other, U_provideinfo, U_yesno, U_unknown
Task success/efficiency	
average goal/subgoal achievement rate	success%
Speech recognition quality	
non-understanding rate	rejection%
average ASR confidence score	confScore
User dialog behavior	
requests for help	help%
touch-tone	dtmf%
barge-in	bargein%
speaking rate	speechRate

Figure 2: Evaluation Measures (and abbreviations).

engine (Huggins-Daines et al., 2006) and a domain-specific voice built with the Festival/Festvox toolkit (Black and Lenzo, 2000) and deployed on the Cepstral Swift engine (Cepstral, LLC, 2005). As of April 2007, the system has received more than 34,000 calls from the general public, all of which are recorded with logs and available for research.

4 Experimental Setup

Our experiment involves collecting, then comparing, two types of dialog corpora involving human users and Let's Go. Here we describe how we collected our *subject corpus* and our *user corpus*, i.e., our two experimental conditions. The same version of Let's Go was used by the users and the subjects.

To collect our subject corpus we used a "standard" laboratory experiment, following typical practices in the field as summarized in Section 2. We recruited 39 subjects (19 female and 20 male) from the University of Pittsburgh who were native speakers of American English. We asked the subjects to come into our lab to call the system from a land-line phone. We designed 3 task scenarios[1] and asked the subjects to complete them in a given sequence. Each task included a departure place, a destination, and a time restriction (e.g., going from the University of Pittsburgh to Downtown, arriving before 7PM). We used map representations of the places and graphic representations of the time restrictions to avoid influencing subjects' language. Subjects were instructed to make separate calls for each of the 3 tasks. As shown in Figure 1, the initial system prompt informed the users that they could say "Help" at any time. We did not give any additional instructions to the subjects on how to talk to the system. Instead, we let the subjects interact with the system for 2 minutes before the experiment, to get a sense of how to use the system. Subjects were compensated for their time at the end of the experiment, with no bonus for task completion. Although we set a time limit of 15 minutes as the maximum time per task, none of the subjects reached this limit.

For our user corpus, we used 4 days of calls to Let's Go (two days randomly chosen from the weekday hours of deployment, and two from the weekend hours of deployment) from the general public. Recall that during nights and weekends, callers to the Port Authority's customer service line are redirected to Let's Go.

5 Evaluation Measures

To examine whether differences exist between our two corpora, we will use the evaluation measures shown in Figure 2. All of these measures are adopted from prior work in the dialog literature.

Schatzmann et al. (2005) proposed a comprehensive set of quantitative evaluation measures to compare two dialog corpora, divided into the following three types: high-level dialog features, dialog style/cooperativeness, and task success/efficiency.

[1] It should be noted that one of these tasks required transferring to another bus, which was not explicitly handled by the system. This task was therefore particularly difficult to complete, especially for subjects not familiar with the Port Authority network. However, because this task represented a situation that users might face, we still included this task in the study.

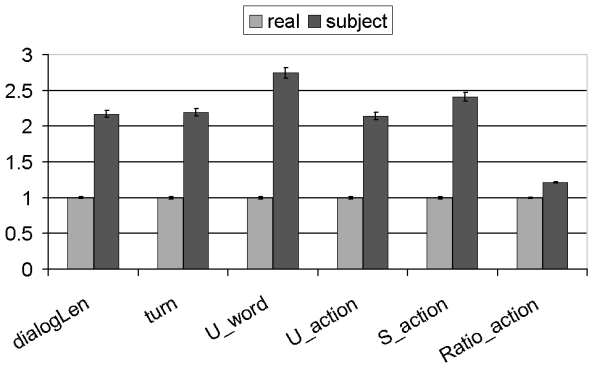


Figure 3: Comparing High-level Dialog Features.

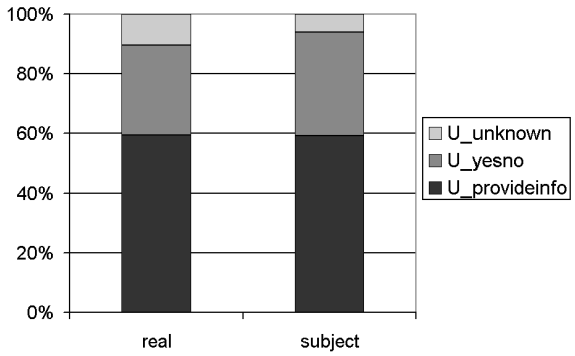


Figure 4: Comparing User Dialog Acts.

We adapt these measures for use in our comparisons, based on the information available in our corpora. For high-level dialog features (which capture the amount of information exchanged in the dialog) and dialog style, we define and count a set of system/user dialog acts. On the system side, **S_requestinfo**, **S_confirm**, and **S_inform** indicate actions through which the system respectively requests, confirms, or provides information. **S_other** stands for other types of system prompts. On the user side, **U_provideinfo** and **U_yesno** respectively identify actions by which the user provides information and gives a yes/no answer, while **U_unkown** represents all other user actions. Finally, **S_action** (resp. **U_action**) represents any of the system (resp. user) actions defined above, and **Ratio_action** is the ratio between **S_action** and **U_action**.

We also define a variety of other measures based on other studies (e.g., (Walker et al., 2000; Turunen et al., 2006)). Two of our measures capture speech recognition quality: the non-understanding rate (**rejection%**) and the average confidence score (**confScore**). In addition, we look into how frequently the users ask for help (**help%**), how often they use touchtone (**dtmf%**), how often they interrupt the system (**bargein%**), and how fast they speak (**speechRate**, number of words per second).

All of the features used to compute our evaluation measures are automatically extracted from system logs. Thus, the user dialog acts and dialog behavior measures are identified based on speech recognition results. For **success%**, we consider a task to be completed if and only if the system is able to get enough information from the user to start a database query and inform the user of the result (i.e., either specific bus schedule information, or a message that the queried bus route is not covered by the system).

6 Results

Our subject corpus consists of 102[2] dialogs, while our user corpus consists of 200 dialogs (90 obtained during 2 weekdays, and 110 obtained over a weekend). To compare these two corpora, we compute the mean value for each corpus with respect to each of the evaluation measures shown in Figure 2. We then use two-tailed t-tests to compare the means across the two corpora. All differences reported as statistically significant have p-values less than 0.05 after Bonferroni corrections.

As a sanity check we first compared the weekday and weekend parts of the user corpus with respect to our set of evaluation measures. None of the measures showed statistically significant differences between these two subcorpora.

Figure 3 graphically compares the means of our high-level dialog features, for both the user and subject dialog corpora. In the figures, the mean values of each measure are scaled according to the mean values of the user corpus, in order to present all of the results on one graph. For example, to plot the means of **dialogLen**, we treat the mean **dialogLen** of the user corpus as 1 and divide the mean **dialogLen** of the subject corpus by the mean of the user corpus. The error bars show the standard er-

[2]Some subjects mistakenly completed more than one task per dialog. Such multi-task dialogs were not included in our analysis, because our evaluation measures are calculated on a per-dialog basis

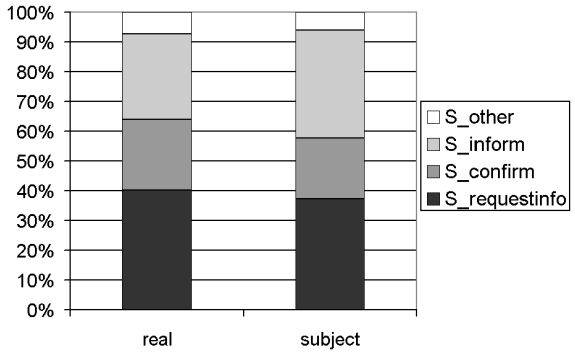


Figure 5: Comparing System Dialog Acts.

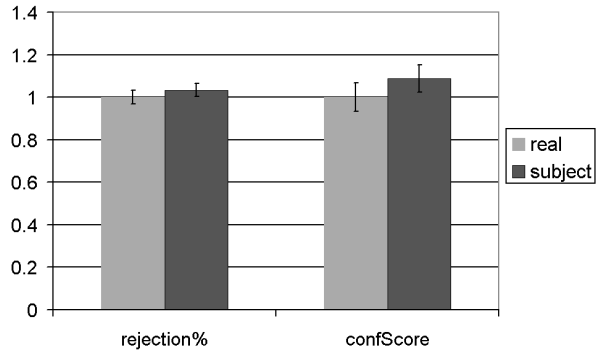


Figure 6: Comparing Speech Recognition Quality.

rors. Using t-tests on the unnormalized means (described above), we confirm that the user dialogs and the subject dialogs are significantly different on all of the high-level dialog features. Subjects talk significantly more than users in terms of number of words per utterance; the number of turns per dialog is also higher for subjects. **U_action** and **S_action** show that both the system and the user transmit more information in the subject dialogs. **Ratio_action** shows that subjects are more passive than users, in the sense that they produce relatively less actions than the system.

Figure 4 (resp. Figure 5) shows the distribution of the user (resp. system) actions in both the user and subject corpora. Subjects give more yes/no answers and produce fewer unrecognized actions than users (these differences are statistically significant). On the other hand, there is no significant difference in **U_provideinfo** between users and subjects. The system provides significantly more information (**S_inform**) to the subjects than to the users, which is consistent with the fact that the task completion rate is higher for subjects. Using automatic indicators to estimate task completion as discussed in Section 5, we find that the completion rate for subjects is 80.7%, while for users it is only 67%. There are also significantly more **S_other** in dialogs with users than with subjects. We did not find any significant difference in the number of system requests (**S_requestinfo**) or confirmations (**S_confirm**).

Figure 6 shows the results for speech recognition quality, using scaled mean values as in Figure 3. There are no statistically significant differences between the number of rejected user turns or the average confidence scores of the speech recognizer. Recall, however, that these measures are automatically calculated using recognition results. Until we can examine speech recognition quality using manual transcriptions, we believe that it is premature to conclude that our speech recognizer performs equally well in real and lab environments.

Figure 7 shows the normalized mean values and standard errors for our user dialog behaviors. Our results agree with the findings in (Turunen et al., 2006). All four measures show significant differences between user and subject dialogs. Users barge in more frequently, use more DTMF inputs, and ask for more help than subjects, while subjects speak faster than users.

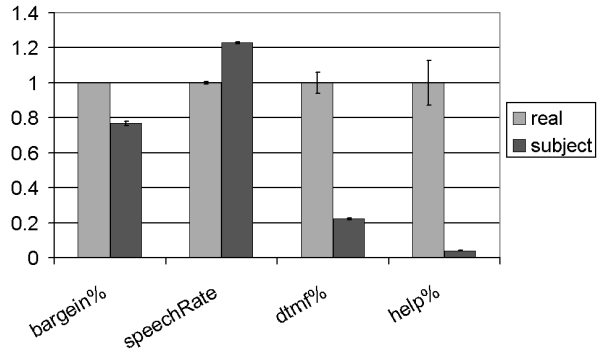


Figure 7: Comparing User Dialog Behaviors.

To summarize, subject dialogs are longer and contain more caller actions than user dialogs, suggesting that subjects are more patient and try harder than users to complete their tasks. In addition, there are less barge-ins and unknown dialog acts in sub-

ject dialogs. Subjects also appear to speak faster than users. This may be because subjects are calling the system in very controlled and quiet conditions, whereas users may experience a higher cognitive load due to their environment (e.g. calling from the street) or emotional state (e.g. concerned about missing a bus).

Finally, in addition to comparing our corpora on the dialog level, we also present a brief examination of the differences between the first user utterances from the dialogs in each corpus. (Because we are only looking at a small percentage of our user utterances, here we are able to use manual transcriptions rather than speech recognition output.) The impact of open system initial prompts on user initial utterances is an interesting question in dialog research (Raux et al., 2006). Most users answer the initial open prompt of Let's Go ("What can I do for you?") with a specific bus route number, while subjects often start with a departure place or destination. Subject queries may be restricted by the assigned task scenarios. However, it is interesting to note that many users call the system to obtain schedule information for a bus route they already know, rather than to get information on how to reach a destination. We also observe that there are only 2% void utterances (when only background noise is heard) in subject dialogs, while there are 20% in user dialogs. This confirms that subjects and users dialog with the system in very different environments.

7 Conclusions and Discussion

In this paper, we investigated the differences between dialogs collected with users in real settings and with subjects in a standard lab setting, and observed statistically significant differences with respect to a set of well-known dialog evaluation measures. Specifically, our results show that subjects talk more with the system and speak faster, while users barge in more frequently, use more touchtone input and ask for more help. Although there are some significant differences in the frequency of particular system/user dialog acts, there is no significant difference in the overall ratios of different dialog acts (i.e., the structure of the dialogs is similar).

Many of the differences we observed suggest that, because users and subjects have different behaviors, a system that is optimal for one population might not be for the other. For instance, the fact that users resort more to system help than subjects and at the same time barge in more often implies different designs for help prompts. Such prompts should be shorter for users to avoid information overload (and early barge-in which prevents them from hearing the message), but might include more information for subjects.

Our results also offer insights for user simulation training. Most current research simulates user behavior on the dialog act level. In this case, training the simulation models from a user corpus or from a subject corpus may not differ much since the dialog act distributions were shown to be similar in our two corpora. At the speech/word level, however, we did see significant differences in user behavior. Thus, simulations trained on subject corpora may be insufficient to train systems that explore problems such as barge-in, switch between modalities, and so on.

Finally, our work can contribute to an understanding of how Let's Go Lab can satisfy the needs of the spoken dialog community. By charting the differences between users and subjects, we can determine how tests carried out on the Lab can translate back to the academic systems of the experimenters.

Acknowledgments

This work is supported by the US National Science Foundation under grants number 0208835 and 0325054. 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 National Science Foundation. We would like to thank the Port Authority of Allegheny County for their help in making the Let's Go system accessible to Pittsburghers.

References

J. F. Allen, B. W. Miller, E. K. Ringger, and T. Sikorski. 1996. *A Robust System for Natural Spoken Dialogue*. In Proceedings of the 34nd Annual Meeting of the Association for Computational Linguistics (ACL).

A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Noth. 2003. *How to Find Trouble in Communication*. Speech Communication, Vol. 40, No. 1-2, pp. 117-143.

A. W. Black and K. Lenzo. 2000. *Building Voices in the Festival Speech System*. http://festvox.org/bsv/

D. Bohus and A. Rudnicky. 2003. *RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda*. In Proceedings of Eurospeech 2003, Geneva, Switzerland.

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

D. Bohus, A. Raux, T. K. Harris, M. Eskenazi, and A. Rudnicky. 2007. *Olympus: an open-source framework for conversational spoken language interface research*. In Proceedings of the HLT-NAACL 2007 workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology, Rochester, NY, USA.

Cepstral, LLC. 2005. *SwiftTM: Small Footprint Text-to-Speech Synthesizer*. http://www.cepstral.com

D. Huggins-Daines, M. Kumar, A. Chan, A. W Black, M. Ravishankar, and A. I. Rudnicky. 2006. *Pocket-Sphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices*. In Proc. of ICASSP 2006.

T. Giorgino, S. Quaglini, and M. Stefanelli. 2004. *Evaluation and Usage Patterns in the Homey Hypertension Management Dialog System*. Dialog Systems for Health Communication, AAAI Fall Symposium, Technical Report FS-04-04

A. Gruenstein, S. Seneff, and C. Wang. 2006. *Scalable and Portable Web-Based Multimodal Dialogue Interaction with Geographical Databases*. In Proc. of ICSLP, 2006.

A. Hof, E. Hagen and A. Huber. 2006. *Adaptive Help for Speech Dialogue Systems Based on Learning and Forgetting of Speech Commands*. In Proc. of 7th SIGdial.

K. S. Hone and R. Graham. 2000. *Towards a tool for the subjective assessment of speech system interfaces (SASSI)*. Natural Language Engineering, 6(3/4), 287-305.

O. Lemon, K. Georgila, J. Henderson. 2006. *Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Evaluation*. In Proceedings of IEEE/ACL Spoken Language Technology.

D. J. Litman and S. Pan. 2002. *Designing and Evaluating an Adaptive Spoken Dialogue System*. User Modeling and User-Adapted Interaction. Vol. 12, No. 2/3, pp. 111-137

S. Möller, R. Englert, K. Engelbrecht, V. Hafner, A. Jameson, A. Oulasvirta, A. Raake, and N. Reithinger. 2006. *MeMo: Towards Automatic Usability Evaluation of Spoken Dialogue Services by User Error Simulations*. In Proc. ICSLP2006.

M. Okamoto, Y. Yang, and T. Ishida. 2001. *Wizard of oz method for learning dialog agents*. Cooperative Information Agents V, volume 2182 of LNAI, pages 20–25.

A. Raux, B. Langner, D. Bohus, A. W Black, M., Eskenazi. 2005. *Let's Go Public! Taking a Spoken Dialog System to the Real World*. In Proceedings of Interspeech 2005 (Eurospeech), Lisbon, Portugal.

A. Raux, D. Bohus, B. Langner, A. W Black, M., Eskenazi. 2006. *Doing Research on a Deployed Spoken Dialogue System: One Year of Let's Go! Experience*. In Proceedings of Interspeech 2006.

V. Rieser, I. Kruijff-Korbayova, and O. Lemon. 2005. *A corpus collection and annotation framework for learning multimodal clarification strategies*. In Proceedings of SIGdial 2005.

A. Roque, A. Leuski, V. Rangarajan, S. Robinson, A. Vaswani, S. Narayanan, and D. Traum. 2006. *Radiobot-cff: A spoken dialogue system for military training*. In Proceedings of International Conference on Spoken Language Processing 2006.

J. Schatzmann, K. Georgila, and S. Young. 2005. *Quantitative Evaluation of User Simulation Techniques for Spoken Dialogue Systems*. Proceedings of 6th SIGdial Workshop on Discourse and Dialogue.

S. P. Singh, M. J. Kearns, D. J. Litman, and M. A. Walker. 2000. *Empirical Evaluation of a Reinforcement Learning Spoken Dialogue System*. Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence.

S. Tomko and R. Rosenfeld. 2006. *Shaping user input in speech graffiti: a first pass*. CHI Extended Abstracts.

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 Proceedings of Interspeech 2006.

M. Walker, J. Aberdeen, J. Boland, E. Bratt, J. Garofolo, L. Hirschman, A. Le, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stallard, and S. Whittaker. 2001. *DARPA Communicator dialog travel planning systems: The June 2000 data collection*. In Proc. EUROSPEECH.

M. A. Walker, C. A. Kamm, and D. J. Litman. 2000. *Towards Developing General Models of Usability with PARADISE*. In Natural Language Engineering, Vol. 6, No. 3.

Dealing with DEAL: A dialogue system for conversation training

Anna Hjalmarsson
Centre for Speech Technology
KTH, Stockholm,
Sweden
annah@speech.kth.se

Preben Wik
Centre for Speech Technology
KTH, Stockholm,
Sweden
preben@speech.kth.se

Jenny Brusk
Department of Game Design, Narrative and Time-Based Media
Gotland University,
Sweden
jenny.brusk@hgo.se

Abstract

We present DEAL, a spoken dialogue system for conversation training under development at KTH. DEAL is a game with a spoken language interface designed for second language learners. The system is intended as a multidisciplinary research platform where challenges and potential benefits of combining elements from computer games, dialogue systems and language learning can be explored.

1 Introduction

There is a growing trend among educational researchers to look at games and game design in order to make education more appealing and effective. A new and challenging domain for spoken dialogue systems is *serious games*, i.e., applications of interactive technology that have purposes other than solely to entertain, including training, advertising, simulation, or education (Iuppa & Borst, 2007). If successful, serious games will engage users motivated by a willingness to be entertained and/or educated. Encouraged by such motivations users will be prepared to talk to dialogue systems because it is fun, repeatedly and for long periods without the need for predefined tasks. This is a tempting scenario.

We present DEAL, a spoken dialogue system for second language learners of Swedish under development at KTH. DEAL is intended as a multidisciplinary research platform where challenges and potential benefits of combining elements from computer games, dialogue systems and language learning can be explored. From a dialogue research point of view a serious game approach contributes with several novel and interesting objectives and challenges. These include how to design dialogues which are fun and natural using a language which suits the vocabulary and language complexity of language learning students on various levels. Since efficiency and task completion are no longer the main objectives, dialogue systems in a serious game context do not have to be predictable, rational or even co-operative. Instead, we need to consider how to build systems which are fun, educational and addictive to talk to.

1.1 Acquiring conversational skills

Language learning can be modelled as a series of developmental steps going from declarative to procedural knowledge. First, an item is noticed in a meaningful contrastive situation, then it occurs repeatedly in meaningful input and is practised in communication until it is internalised, and finally automatised (Ellis, 2006). To automatise these processes when learning a second language we need a meaningful situation where conversational skills can be practised repeatedly. Because of its complexity, learning a language requires substantial effort and the motivation varies both over time and between individuals. To practise conversational skills while playing a game may increase any existing motivation to learn if there is one, and creates a motive to learn if there isn't. Our objective is similar to the Nice project (Gustafson et al., 2004), in that we wish to create a game where spoken dialogue is not just an add-on, but is used as the primary means for game progression.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 132–135,
Antwerp, September 2007.

2 Motivation

The practical motivation of DEAL is to build an application where conversational skills can be practised in a fun and meaningful context. In short, DEAL is a game with a spoken language interface designed for second language learners. A similar approach is used in the tactical language training system (TLTS), a large-scale application that helps people acquire basic conversational skills in Levantine and Iraqi Arabic (Johnson et al., 2005). Our first choice of domain for this work is the trade domain. DEAL sets the scene of a flea market where a talking animated agent is the owner of a shop where used objects are sold. The domain was chosen for several reasons:

- A trading situation is a fairly restricted and universally well-known domain. It is something everyone is conceptually familiar with, regardless of cultural and linguistic background.
- A trading situation is from a language learning point of view a very useful domain to master in the new language
- The objects sold at a flea market can be a diverse set of items which can be tailored to suit the vocabulary mastered by a language learning student.
- A flea market is a place where it is acceptable to negotiate about the price. Negotiation is a complex process which includes both rational and emotional non-rational elements. This opens up for interesting and complex dialogue.

These characteristics combined gives us an application where users can engage in a dialogue situated in a well-known context but which also includes elements of surprise and challenge (i.e., getting a good price).

2.1 Ville

DEAL is developed as a free-standing part of Ville, a framework for language learning developed at KTH (Engwall et al., 2004). Ville is a virtual language tutor helping students to improve their listening and pronunciation skills in a new language. Ville detects and gives feedback on pronunciation errors, and has challenging exercises that are used in order to teach new vocabulary, or to raise the students' awareness of particular perceptual differences between their first and second language. Ville has exercises on phone, syllable, word and sentence level.

DEAL adds the possibility to give conversation training. Whereas Ville is a language tutor who provides the user with feedback on performance, the agent in DEAL does not comment on your performance but acts as your conversation partner in a role-playing fashion. Using DEAL as an integrated part of Ville, the system has knowledge about particular students' acquired vocabulary. This information can be used to tailor the language in DEAL as well as the items being sold.

3 Implementation

DEAL is implemented using components from the Higgins project (Skantze, 2005), an off-the-shelf ASR system, a dialogue manager developed for DEAL purposes and a GUI with an embodied conversational agent (ECA).

3.1 User interface

Our ECA (embodied conversational agent) is developed at KTH (Beskow, 2003), and can use either synthetic or natural, pre-recorded speech. The head is capable of producing lip-synchronized speech as well as extra linguistic signs such as frowning, nodding, and eyebrow movements. Language is multimodal, and in second language learning, visual signals are an important source of information.

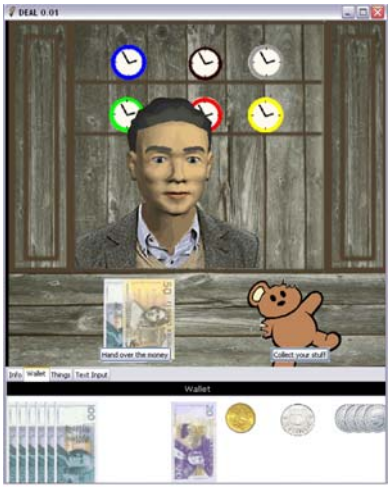


Figure 1: DEAL user interface

Higgins includes modules for semantic interpretation and analysis. Pickering, a modified chart parser, supports continuous and incremental input from a probabilistic speech recognizer. Speech is unpredictable and chunking a string of words into utterances is difficult since pauses and hesitations will likely be incorrectly interpreted as end of utterance markers. This will be even more evident for second language learners whose conversational skills are not yet automatised and whose language contains disfluencies such as hesitations and false starts. Pickering uses context free grammars (CFG) and builds deep semantic tree structures. Grammar rules are automatically relaxed to handle unexpected, ungrammatical and misrecognized input robustly. The discourse modeler, Galatea, interprets utterances in context and keeps a list of the communicative acts (CA) in chronological order. Galatea resolves ellipses, anaphora and has a representation of grounding status which includes information about who added a concept, in which turn a concept was introduced and the concept's ASR confidence score.

4 The DEAL domain

Game designers focus on finding ways to keep players engaged and motivated throughout a game. Nonetheless, dialogues in today's games have a strict way of affecting the continuance of the game. The interaction is typically based on complex tree structures, where one action leads to a set of new choices. Choosing one line or topic has an immediate result and the dialogue traverses a finite branching tree structure. With these types of dialogues it is fairly trivial how to get the desired result, making it less interesting to engage in the interaction. We strive towards an interaction with a less predictable result. Façade is an interactive drama project that introduces a drama manager to make the outcome of a dialogue less predictable (Mateas & Stern, 2003). In Façade the story is divided into beats, an atomic unit of drama, where beats and transitions between beats can unfold in various ways depending on what type of input is provided by the user.

4.1 Dealing with DEAL

DEAL has two actors, one ECA and one human language student. The student is given a mission to buy items at a flea market getting the best possible price from the odd looking shop-keeper. The shop-keeper can talk about objects and their properties and negotiate about the price of the objects. The most challenging part in DEAL, both from a "buyer" (user) point of view and when designing the conversational agent, is negotiating about the price of objects. At first, dealing about price can seem like a fairly rational and straight forward procedure. However, negotiating is a complex multidisciplinary area of research which touches fields such as psychology, economics and political science. Negotiating about a price in a face to face situation involves a number of various parameters which are often affected by non-rational and emotional aspects. Second hand items may have rich interesting characteristics which makes them interesting to talk about. For example the items can be defective, have a personal history or an affection value to the shop-keeper, all of which may have an impact on the negotiation process.

The dialogue can unfold in different ways depending on what the user says (see Figure 2). Negotiation is implemented using a fairly straight forward algorithm and a few heuristics. To introduce elements of gameplay we have integrated a parameter which represents the agent's "willingness" to reduce the price of an item. The willingness parameter is the percentage share of the seller's original price that the ECA is willing to accept, after negotiating, as price for a particular item. The parameter has an initial value which may be affected depending on how the dialogue proceeds. To affect the outcome of the interaction, the player may try to influence the willingness of the shop-keeper to reduce the price.

U1: I'm interested in buying a toy.
S1: Oh, let me see. Here is a doll.
(a doll is displayed)
U2: Do you have a teddy-bear?
S2: Oh, yeah. Here is a teddy-bear.
(a teddy-bear is displayed, see Figure 1)
U3: How much is it?
S3: You can have it for 180 SEK
U4: I give you 1 SEK *(willingness decrease)*
S4: No way! That is less than what I paid for it.
U5: Ok how about 100?
S5: Can't you see how nice it is?
U6: But one ear is missing. *(willingness increase)*
S6: Ok, how about 150?
U7: 130?
S7: Ok, it is a deal!

Figure 2: Dialogue example from DEAL

The outcome of the game is affected by what the user says. For example in utterance U4 the seller is offended by the user's low bid and his willingness to give the user a good price is reduced. However, when the user points out a flaw of the object (the GUI displays a teddy-bear with one ear, see Figure 1) the seller feels obligated to give the user a better price, i.e., his willingness increases.

4.2 Dialogue characteristics in DEAL

Humans who engage in a dialogue tend to coordinate their linguistic behaviour (Pickering & Garrod, 2006), sometimes referred to as entrainment. Research on linguistic entrainment in human-machine interaction has shown that users of spoken dialogue systems also adopt the system's way of speaking (see for example Brennan, 1996). Moreover, research and literature on second language acquisition (SLA) is diverse, with no single theory or model seen as the most appropriate. However, there seem to be a consensus about the value of conversational interactions. The more you talk the better it is.

Consequently, from a second language learning perspective, the language used in DEAL will be crucial. It is important that the agent behaves human-like in a way which motivates the users to talk a lot and not only in short command-like utterances. The goal is not to create a conversational agent which behaves human-like in every sense but which is human enough to make the users suspend their disbeliefs, i.e. make them act as if they were talking to another human being (Cassell, in press). This does not necessarily mean that the agent needs to be cooperative or polite. The seller can actually be rude and try to avoid the users' requests as long as this is done in a way that does not destroy the users' willingness to accept the ECA in DEAL as a character with human-like conversational capabilities.

5 Concluding remarks

Whether DEAL is a fun game or not is yet to be investigated. So far, the scenario, rules and possible actions in DEAL are fairly limited. Much can be added to the system in the long run, but this far our main motivation has been to introduce simple examples of social interaction that affect game progression.

6 Acknowledgements

This research was carried out at Centre for Speech Technology, KTH. The research is also supported by the Graduate School for Language Technology (GSLT). Many thanks to Rolf Carlson, Gabriel Skantze , Joakim Gustafson, Jens Edlund.

References

Beskow, J. (2003). *Talking Heads - Models and Applications for Multimodal Speech Synthesis*. Doctoral dissertation, KTH.

Brennan, S. E. (1996). Lexical entrainment in spontaneous dialog. In *Proceedings of ISSD*.

Cassell, J. (in press). Body Language: Lessons from the Near-Human. In J. Riskin (ed.) The Sistine Gap: History and Philosophy of Artificial Intelligence . Chicago: University of Chicago Press.

Ellis, N. (2006). Selective Attention and Transfer Phenomena in L2 Acquisition: Contingency, Cue Competition, Salience, Interference, Overshadowing, Blocking, and Perceptual Learning. *Applied Linguistics, 27*, 164-194.

Engwall, O., Wik, P., Beskow, J., & Granström, G. (2004). Design strategies for a virtual language tutor. In Kim, S. H. & Young, D. H. (Eds.), *Proc ICSLP 2004* (pp. 1693-1696). Jeju Island, Korea.

Gustafson, J., Bell, L., Boye, J., Lindström, A., & Wirén, M. (2004). The NICE Fairy-tale Game System. In *Proceedings of SIGdial*. Boston.

Iuppa, N., & Borst, T. (2007). *Story and simulations for serious games : tales from the trenches.* Focal Press.

Johnson, W., Vilhjalmsson, H., & Marsella, S. (2005). Serious games for language learning: How much game, how much AI?. In *12: th International Conference on Artificial Intelligence in Education*. Amsterdam.

Mateas, M., & Stern, A. (2003). Façade: An experiment in building a fully-realized interactive drama. In *Game Developer's Conference: Game Design Track*. San Jose, California, US.

Pickering, M., & Garrod, S. (2006). Alignment as the Basis for Successful Communication. *Research on Language and Computation, 4*, 203-228.

Skantze, G. (2005). Galatea: a discourse modeller supporting concept-level error handling in spoken dialogue systems. In *Proceedings of SigDial* (pp. 178-189). Lisbon, Portugal.

Referring under Restricted Interactivity Conditions

Raquel Fernández, Tatjana Lucht, David Schlangen
Department of Linguistics
University of Potsdam, Germany
{raquel|lucht|das}@ling.uni-potsdam.de

Abstract

We report results on how the collaborative process of referring in task-oriented dialogue is affected by the restrictive interactivity of a turn-taking policy commonly used in dialogue systems, namely *push-to-talk*. Our findings show that the restriction did not have a negative effect. Instead, the stricter control imposed at the interaction level favoured longer, more effective referring expressions, and induced a stricter and more structured performance at the level of the task.

1 Introduction

The collaborative process by means of which people coordinate in identifying referents in dialogue has motivated a fair amount of psycholinguistic studies. While most of them experiment with natural, fully interactive conditions (Clark and Wilkes-Gibbs, 1986; Garrod and Anderson, 1987) some, like e.g. (Krauss and Weinheimer, 1966; Clark and Krych, 2004), have investigated how the referring process is affected by non-interactive settings that lack cotemporality (speakers do not receive messages in real time) and simultaneity (speakers cannot communicate at once). This is done by letting speakers talk to a tape recorder for future addressees, which fully precludes any form of interaction.

In the work we report here, we wanted to investigate a condition with *restricted* interactivity, in which cotemporality is allowed but simultaneity is inhibited. This is a setting commonly found in spoken dialogue systems that use a *push-to-talk* turn-taking strategy. To investigate the effects of this restriction in isolation, we conducted an experiment where we let subjects do a referring task either with free turn-taking or with turn-taking controlled by a half-duplex channel managed by *push-to-talk*.

Such restrictions of interactivity are often seen as having negative impact on the efficiency of the dialogue (Whittaker, 2003) as they affect the ability to give immediate and concurrent feedback and hence disturb the grounding process (Clark and Schaefer, 1989). As we have reported in recent work (Fernández et al., 2007), however, we found that subjects in the restricted condition were able to solve the task in roughly the same time, with no loss of efficiency. We hypothesised that one of the reasons behind this was a more cautious strategy whereby subjects proceed by more firmly grounding each step in the task, which was favoured by the turn-taking restriction, In this short paper we extend the analysis to investigate in more detail the effect of the *push-to-talk* restriction on the shape of the referring process. As we shall see, our findings support our previous conclusion that, for some tasks, higher interactivity is not necessarily advantageous.

After briefly describing the experimental procedure in the next section, in Section 3 we summarise the global patterns observed in the dialogues and then focus on the referring process in Section 4. We close with some conclusions and pointers for further work.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 136–139,
Antwerp, September 2007.

2 Experimental Setup

2.1 Task & conditions

In the task to be carried out in our experiment an *instruction giver* (IG) tells an *instruction follower* (IF) how to build up a *Pentomino* puzzle (see Figure 1). The IG has the solution of the puzzle, while the IF has the puzzle outline and the set of loose pieces.

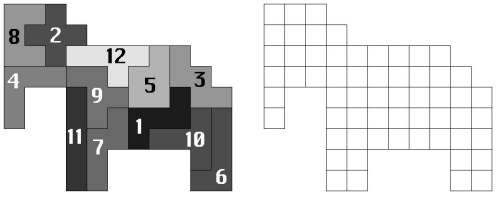


Figure 1: Puzzle and Outline

The IG is asked to tell the IF how to assemble the puzzle following the numbers shown in Figure 1. The pieces that the IF has at her disposal are however not numbered and are all the same colour.

We experiment with two different conditions: a fully interactive *free turn-taking* (FTT) condition, and a *push-to-talk* (PTT) condition where interactivity is restricted. In both conditions, subjects are in different sound-proof rooms and communication is only verbal. In FTT participants communicate by means of headsets with a continuously open audio channel. In PTT subjects use walkie-talkies that only offer a half-duplex channel that precludes simultaneous communication. Speakers have to press a button to get the turn, hold it to keep it, and release it again to yield it.

Twenty German native speakers, 11 females and 9 males between 20 and 40 years old, participated in the experiment. They were grouped in 10 IG-IF pairs and 5 pairs were assigned to each of the two conditions.

2.2 Coding

The 10 dialogues collected make up a total of 194.54 minutes of recorded conversation (in German). The recordings were transcribed and segmented into a total of 2,262 turns, 4,300 utterances and 28,969 words using the software Praat (Boersma, 2001).

Using MMAX (Müller and Strube, 2001), we annotated the dialogues at three different levels:

Dialogue acts (DAs). We distinguish between task acts (including a tag for *description acts* where a piece or a location are described) and grounding acts (including different types of feedback acts and clarification requests). More details on the scheme used can be found in (Fernández et al., 2007).

Moves. The task can be divided into 12 *moves* or cycles, one for each piece. A move covers all speech that deals with a particular piece, from the point when the IG starts to describe it (*"The next piece looks like Oklahoma"*) to the point when the subjects move on to the next item. Moves are sometimes closed with errors, which may lead to later repairs. All speech that deals with the repair of a previously closed move is annotated as a *repair sequence*.

Referential expressions. We annotated the referential expressions used by the subjects distinguishing between those that referred to a piece (*"the Swiss cross"*), those that referred to part of a piece (*"a square sticking up"*), and those that referred to a location on the board (*"between the legs of the elephant"*). Note that referential expressions and description acts are different kinds of units, with the former typically being part of the latter.

3 Global Patterns

All pairs of subjects were able to finish the task and in both conditions they did so in roughly the same time (18.7 min in PTT and 19.8 min in FTT on average; no significant difference). The PTT condition thus did not have any significant impact on task efficiency, although it did have an effect on the shape of the dialogues. PTT pairs were able to finish the task using significantly fewer words than FTT pairs. The structural patterns observed were also highly different across conditions: FTT dialogues contain roughly twice as many turns and utterance as PTT dialogues, with turns and utterances in PTT being much longer than in FTT. Table 1 gives an overview of these results.

average # of	FTT	PTT	t-test, df=8
words/dialogue	3540	2254	$p < 0.05$
turns/dialogue	328	115	$p < 0.005$
utts/dialogue	596	264	$p < 0.005$
words/utt	6	8.6	$p < 0.01$
words/turn	11.3	20.2	$p < 0.05$

Table 1: Summary of structural patterns

We also found that there were significant differences in the distribution of DAs. In particular, the proportion of positive feedback acts, like backchannels and acknowledgements, was consistently higher in FTT (33.8% vs 25.7% on average; χ^2 test, $p < 0.01$), while PTT dialogues contained a higher proportion of task-related acts (45.4% vs 36.7% on average; χ^2 test, $p < 0.01$). The reader will find an extensive discussion of these results in (Fernández et al., 2007).

4 Analysing the Referring Process

In this section, we report some results of our analysis of the referring process and of how this is affected by the global patterns brought about by our two experimental conditions.

4.1 Internal structure of moves

The moves that deal with the different pieces of the puzzle include several sub-tasks: (i) identifying the piece in question, (ii) optionally describing its orientation, and (iii) establishing its location on the board. The latter is the most challenging of the three and the one on which subjects spend most of the effort: in both FTT and PTT, slightly over 60% of the referring expressions used deal with the identification of board locations. For each move, there is minimally one change in sub-tasks, typically with a transition from (i) to (ii). These sub-tasks, however, are not always addressed in the canonical order and often subjects go back and forth between them during a single move.

To measure the orderliness of the referential process, we counted the number of times subjects changed to a different sub-task within a move and found significant differences between conditions. We observed that, on average, subjects in PTT dialogues change to a different referential sub-task 1.2 times per move, while in FTT dialogues the average number of changes per move is 2. These differences are statistically significant (t=3.18, df=8, $p < 0.02$). Thus, participants in PTT dialogues tend to follow a more structured strategy where they first deal with the description of a piece and then with its location, making sure that each of these phases is grounded. This suggests that the stricter control imposed by the turn-taking restriction on the interaction level leads to a stricter and better structured performance at the task level.

4.2 Referential expressions

In the collaborative model put forward by (Clark and Wilkes-Gibbs, 1986), the referential process is divided into three phases: an *initiating* phase, a *refashioning* phase, and a *concluding* phase. In a basic exchange, like e.g. (1), only the first and last phases occur, which correspond to the presentation and acceptance phases of grounding any dialogue contribution.

(1) A: Number 2 is a cross
B: OK, I have that one.

Refashioning may take place because the initial reference is not properly understood or not accepted, or simply because the speaker considers it insufficiently adequate.

According to this model, *"there is a trade-off between initiating the noun phrase and refashioning it. The more effort a speaker puts into the initial noun phrase, in general, the less refashioning it is likely to need."* As the authors point out, however, due to constraints like time pressure and the possible complexity of the referring task, speakers do not always put in enough effort to avoid refashioning, which—in conditions with full interactivity—leads to a more collaborative and interactive process. Indeed, our FTT dialogues are full of installments, provisional references and descriptions presented by proxy, as in the following example (translated from German), all of which are rare in PTT dialogues.

(2) IG: It looks kind of a bit like...
IF: Like an inverted L with an extra bit.
IG: Yeah, could be. Basically like a duck.

Although the referential expressions used in PTT tend to be longer, overall their average length

is not significantly different across conditions (12.6 vs 9.7 words on average; not significant). Averaging over all referential expressions, however, conceals important differences in the way in which the referential process unfolds. The differences are to be found in those expressions used in the initiating phase of the referring process. In particular, we observed that the average length of the initial descriptions used to refer to locations (which, as mentioned above, is the most prominent sub-task) is significantly higher in PTT dialogues (13.6 vs 8.7 words on average; t=2.30, df=8, $p < 0.05$). More generally, the average length of the referential expressions used in initial moves (as opposed to repair sequences; see Section 2.2) is also higher in the restrictive interactivity condition (12.6 vs 9.26; t=2.34, df=8, $p < 0.05$).

This underlines the aforementioned trade-off between the cost of producing detailed initial descriptions and the cost of interactively designing the referential expressions. The turn-taking restriction favours longer initial descriptions, which turns out to be advantageous since interactive refashioning, as in (2) above, is harder in this condition.

5 Conclusions and Further Work

We have reported some first results of our ongoing investigation of the referring process in restricted task-oriented dialogue. We have seen that there is a correspondence between the interaction and the task levels, with restricted interactivity leading to more orderly task performance. We have also observed that our PTT condition favours a strategy whereby participants put more effort in the initiating stages of the referring task, which seems to be advantageous for the task at hand.

Our findings so far support the idea that tasks that require complex, spontaneously generated contributions may not be adversely affected or even be supported by interactivity restrictions. Although understanding such complex descriptions as occur in our corpus is of course way beyond the current state of spoken dialogue systems, our results should be of more immediate significance for designing computer-mediated interaction systems.

We are currently developing a classification scheme for locative referring expressions in the lines of the taxonomy used in (Clark and Wilkes-Gibbs, 1986) for noun phrases. This will allow us to analyse the referring process further and investigate how phenomena like e.g. lexical entrainment and the increasing simplification of referring expressions (which, as shown by (Krauss and Weinheimer, 1966; Clark and Krych, 2004) are severely affected in non-interactive setting) are altered under the restricted interactivity imposed by a PTT policy.

Acknowledgements. This work was supported by the EU Marie Curie Programme (first author) and the DFG Emmy Noether Programme (last author). Thanks to the anonymous reviewers for their helpful comments.

References

P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9–10).

H. Clark and M. Krych. 2004. Speaking while monitoring addressees for understanding. *Journal of Memory and Language*, 50:62–81.

H. Clark and E. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294.

H. Clark and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

R. Fernández, D. Schlangen, and T. Lucht. 2007. Push-to-talk ain't always bad! comparing different interactivity settings in task-oriented dialogue. In *Proceedings of DECALOG*, Trento, Italy.

S. Garrod and A. Anderson. 1987. Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, 27:181–218.

R. Krauss and S. Weinheimer. 1966. Concurrent feedback, confirmation, and the encoding of referents in verbal communication. *Journal of Personality and Social Psychology*, 4:343–346.

C. Müller and M. Strube. 2001. MMAX: A tool for the annotation of multi-modal corpora. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

S. Whittaker. 2003. Theories and methods in mediated communication. In *The Handbook of Discourse Processes*, pages 243–286. Lawrence Erlbaum Associates.

A Multidimensional Approach to Utterance Segmentation and Dialogue Act Classif cation

Jeroen Geertzen and **Volha Petukhova** and **Harry Bunt**

Dept. of Communication & Information Sciences,
Tilburg University, The Netherlands,
{j.geertzen,v.petukhova,h.bunt}@uvt.nl

Abstract

In this paper we present a multidimensional approach to utterance segmentation and automatic dialogue act classif cation. We show that the use of multiple dimensions in distinguishing and annotating units not only supports a more accurate analysis of human communication, but can also help to solve some notorious problems concerning the segmentation of dialogue into functional units. We introduce the use of per-dimension segmentation for dialogue act taxonomies that feature multi-functionality and show that better classif cation results are obtained when using a separate segmentation for each dimension than when using one segmentation that f ts all dimensions. Three machine learning techniques are applied and compared on the task of automatic classif cation of multiple communicative functions of utterances. The results are encouraging and indicate that communicative functions in important dimensions are easy machine-learnable.

1 Introduction

Computer-based interpretation and generation of human dialogue is of growing relevance for today's information society. As natural language based dialogue is increasingly becoming an attractive and technically feasible human-machine interface, so the analysis of human-human interaction (for example in interviews or meetings) is becoming important for archival and retrieval purposes, as well as for knowledge management purposes and for the study of social interaction dynamics.

Since people involved in communication constantly perceive, understand, evaluate, and react to each other's intentions as encoded in statements, questions, requests, offers, and so on, a natural approach to the analysis of human dialogue behaviour is to assign meaning to dialogue units in terms of dialogue acts. The identif cation and automatic recognition of the dialogue acts or *communicative functions*[1] of utterances is therefore an important task for dialogue analysis and the design of applications such as computer dialogue systems.

The assignment of appropriate meanings to 'dialogue units' presupposes a way to segment a dialogue into meaningful units. This turns out to be a complex task in itself. Many previous studies in the area of the automatic dialogue act assignment were typically carried out at the level of 'utterances' or that of 'turns'. A turn can be def ned as a stretch of communicative behaviour produced by one speaker, bounded by periods of inactivity of that speaker or by activity of another speaker (Allwood, 2000). While turn boundaries can be recognised relatively easily, for some analysis segmentation into turns is often unsatisfactory because a turn may contain several smaller meaningful parts. Utterances, on the other hand, are linguistically def ned stretches of communicative behaviour that have one or multiple communicative functions. Utterances may coincide with turns but are usually smaller.

[1] In this paper, we use the terms 'dialogue act' and 'communicative function' synonymously.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 140–149,
Antwerp, September 2007.

The detection of utterance boundaries is a highly nontrivial task. Syntactic features (e.g. part-of-speech, verb frame boundaries of finite verbs) and prosodic features (e.g. boundary tones, phrase final lengthening, silences, etc.) are often used as indicators of utterance endings (Shriberg et al., 1998; Stolcke et al., 2000; Nöth et al., 2002).

One of the problems with dialogue segmentation into utterances is that utterances may be discontinuous. Spontaneous speech in dialogue usually includes filled and unfilled pauses, self-corrections and restarts; for example, the speaker of the utterance in (1) corrects himself two times.

(1) *About half ...* ***about a quar- ... th- ...third*** *of the way down I have some hills*

Dialogue utterances may be interrupted by even more substantial segments than repairs and stallings. For example, the speaker of the utterance in (2) interrupts his Inform with a WH-Question:

(2) *Because twenty five Euros for a remote...* ***how much is that locally in pounds?*** *is too much money to buy an extra remote or a replacement remote*

Examples such as (1) and (2) show that the segmentation of dialogue into utterances that have a communicative function requires these units to be potentially discontinuous. In some cases a dialogue act may be performed by an utterance formed by parts of more than one turn. This often happens in polylogues where participants may interrupt each other or talk simultaneously. For example:

(3) *A: Well we can chat away for ... um... for five minutes or so I think at...* ***B: Mm-hmm*** *... at most*

Another case of a dialogue act that is spread over multiple turns occurs when the speaker is providing complex information and divides it up into parts in order not to overload the addressee, as is shown in (4). The first part of the discontinuous segment that expresses S's answer also has a feedback function (making clear to U what S understood).

(4) U: *Could you tell me what time there are flights to Kuala Lumpur on Monday?*
S: *There are two early KLM flights, at 7.30 and at 8:25,..*
U: *Yes,...*
S: *... and a midday flight by Garoeda at 12.10,...*
U: *Yes,...*
S: *And there's late afternoon flight by Malaysian Airways at 17.55.*

The material in the three turns contributed by S together constitute the 'utterance' expressing S's answer to U's question. Examples such as these show that the units in dialogue that carry communicative functions are often very different from the traditional linguistically defined notion of an utterance. We therefore prefer to give these units a different name, that of *functional segment*, and we define these units as "(possibly discontinuous) stretches of communicative behaviour that have one or more communicative functions" (Bunt and Schiffrin, 2007). In many cases a functional segment corresponds to an 'utterance' as defined by certain linguistic properties, but in other cases it does not; and so the question arises how functional segments can be recognised. This is one of the main issues that this paper addresses.

When we want to segment a dialogue into functional segments, one complication is that of discontinuous segments, either within a turn or spread over several turns as we have already discussed. An even greater challenge is posed by those cases where different functional segments overlap, as in the example shown in 5.

(5) U: *W*hat time is the first train to the airport on Sunday?
S: ***The first train to the airport on Sunday*** is at ...ehm... 6.17.

The first part of S's turn repeats most of the preceding question, displaying what the system has heard, and as such has a feedback function. The turn as a whole minus the part ...ehm... has the communicative function of a WH-Answer, and that part has a stalling function. So the segments corresponding to the WH-Answer and the feedback function share the part "The first train to the airport on Sunday". This means that in this turn we have two functional segments starting at the same position but ending at different ones; in other words, no single segmentation of this turn exists that gives us all the relevant functional segments.

To resolve this problem adequately, we propose not to maintain a single segmentation, but to use multiple segmentations in order to allow multiple functional segments that are associated to a specific utterance to be identified more accurately. This approach is compatible with dialogue act taxonomies that address several aspects ('dimensions') of the

interactive process simultaneously (e.g. DAMSL (Core and Allen, 1997) or DIT (Bunt, 2006)), such as the task or activity that motivates the dialogue, the management of taking turns, or timing and attention. This multidimensional view of dialogue naturally leads to the suggestion of approaching dialogue segmentation in a similarly multidimensional way, and to allow the segmentation of a dialogue *per dimension* rather than in one fixed way. In the case of example (5), this means that S's turn is segmented in the three dimensions addressed by the functional segments in this turn:

- Dimension Task/Activity: segment the turn as consisting of the discontinuous segment "The first train to the airport on Sunday is at / 6.17", which has a communicative function in this dimension, and the contiguous segment *...ehm...*, which does not;
- Dimension Feedback: segment the turn as consisting of the contiguous segment *The first train to the airport on Sunday*, which has a function in this dimension, and the contiguous segment *is at ...ehm... 6.17*, which does not;
- Dimension Time Management: segment the turn as consisting of the contiguous segment *...ehm...*, which has a communicative function in this dimension, and the discontinuous segment: *The first train to the airport on Sunday is at 6.17*, which does not.

In recent work the benefits of multidimensional approaches of dialogue act annotation have been discussed and it has been argued that such approaches allow a more accurate modelling of human dialogue behaviour (Petukhova and Bunt, 2007). In this paper we report the results of two studies: one on segmentation and one on classification of dialogue acts in multiple dimensions using various machine learning techniques. In Section 2 we will outline the two series of experiments describing the data, features, and algorithms that have been used. Section 3 and 4 report on the experimental results on segmentation and classification, respectively. Consequently, conclusions are drawn in Section 5.

2 Studies outline

The first study is motivated by the question of whether a different segmentation for each of the DIT dimensions (per-dimension segmentation) rather than a single segmentation for all dimensions will allow more accurate labelling of the communicative functions. In the second study we present the results of a series of experiments carried out in order to assess the automatic recognition and classification of communicative functions. For this purpose we apply machine-learning techniques. Such techniques have already successfully been used in the area of automatic dialogue processing[2]. Our approach is to train classifiers to learn communicative functions in multiple dimensions, taking functional segments as units.

2.1 Corpus data

In our experiments we used two data sets, namely, human-human dialogues in Dutch (the DIAMOND corpus (Geertzen et al., 2004)) for both the segmentation study, and the classification study and human-human multi-party interactions in English (AMI-meetings)[3] for the classification study.

The *DIAMOND corpus* contains human-machine and human-human Dutch dialogues that have an assistance-seeking nature. The dialogues were video-recorded in a setting where the subject could communicate with a help desk employee using an acoustic channel and ask for explanations on how to configure and operate a fax machine. The dialogues were orthographically transcribed and 952 utterances representing 1,408 functional segments from the human-human subset of the corpus have been selected.

The *AMI corpus* contains manually produced orthographic transcriptions for each individual speaker, including word-level timings that have been derived using a speech recogniser in forced alignment mode. The meetings are video-recorded and each dialogue is also provided with sound files (for our analysis we used recordings made with short range microphones to eliminate noise). Three scenario-based[4] meetings were selected to constitute a training set of 3,676 functional segment instances.

[2] See e.g. (Clark, 2003) for an overview.

[3] Augmented Multi-party Interaction (`http://www.amiproject.org/`).

[4] Meeting participants play different roles in a fictitious design team that takes a new project from kick-off to completion over the course of a day.

Table 1 gives percentages of occurrence of the ten most frequently observed tags in both training sets.

AMI data		DIAMOND data	
Tag	Perc.	Tag	Perc.
Time;STALLING	20.7	Task;INSTRUCT	14.8
Auto-FB;POS.OVERALL	18.7	Task;INFORM	7.7
Turn;Turn Keeping	7.5	Time;stall	6.5
Task;INFORM	6.8	Task;INFORM elaborate	6.3
Task;INFORM Elaborate	3.5	Auto-FB;POS.OVERALL	6.2
Task;INF.Agreement	2.5	Task;WH-Question	4.5
Task;YN-Question	2.3	Auto-FB;POS.INT	3.1
Task;SUGGEST	2.0	Task;YN-Question	2.9
Task;INFORM Justify	2.0	Task;CHECK	2.6
Task;CHECK	1.6	Task:INFORM Clarify	2.1

Table 1: Percentage of instances for most frequent tags in the AMI and DIAMOND training sets.

For the DIAMOND training set, the order for the most frequently addressed dimensions is similar with Task dimension (45.6%), followed by Auto-Feedback (19.2%), and Turn Management (16.8%). For the AMI training set, the majority of the dialogue units address the Task dimension (33%), followed by Auto-Feedback (21.7%), Time Management (20.3%) and Turn Management (12.5%).

2.2 Tagset

Both data sets were annotated with the DIT^{++} tagset[5]. The DIT taxonomy distinguishes 11 dimensions, addressing information about: the domain or task (*Task*), feedback on communicative behaviour of the speaker (*Auto-feedback*) or other interlocutors (*Allo-feedback*), managing difficulties in the speaker's contributions (*Own-Communication Management*) or those of other interlocutors (*Partner Communication Management*), the speaker's need for time to continue the dialogue (*Time Management*), establishing and maintaining contact (*Contact Management*), about who should have the next turn (*Turn Management*), the way the speaker is planning to structure the dialogue (*Dialogue Structuring*), introducing, changing or closing the topic (*Topic Management*), and the information motivated by social conventions (*Social Obligations Management*).

For each dimension, at most one communicative function can be assigned, which can either occur in this dimension alone (the function is *dimension specific*) or occur in all dimensions (the function is *general purpose*). For example, the utterance in 1 has a dimension-specific function SELF CORRECTION assigned to it that can only be assigned in the *Own Communication Management* dimension. Utterance A in example 3 has the communicative function of INFORM in the *Dialogue Structuring* dimension. Being a *general purpose* function, INFORM could possibly also be assigned to any other dimension (such as e.g. *Task*).

The tagset used in the studies contains 38 domain-specific functions and 44 general purpose functions. As a result of difference in function type, a tag consists either of a pair of the addressed dimension (D) and general purpose function (GP) or the addressed dimension and dimension specific function (DS). Some functional segments can address several dimensions simultaneously. For example, utterances like *uhm.., ehm..* have the communicative function of STALLING in the dimension *Time Management*, but also have the TURN KEEPING function in the *Turn Management* dimension. These utterances typically have two $\langle D, DS\rangle$ tags assigned: $\langle TimeM, STALLING\rangle$ and $\langle TurnM, KEEPING\rangle$.

For both data sets the annotation is first carried out on a single segmentation and then additionally on dialogue segmented in each of the dimensions separately.

2.3 Features

Every communicative function is required to have some reflection in observable features of communicative behaviour, i.e. for every communicative function there are devices which a speaker can use in order to allow its successful recognition by the addressee such as linguistic cues, intonation properties, dialogue history, etc. State-of-the-art automatic dialogue understanding uses all available sources to interpret a spoken utterance. Features and their selection play a very important role in supporting accurate recognition and classification of functional segments and their computational modelling may be expected to contribute to improved automatic dia-

[5]For more information about the tagset and the dimensions that are identified, please visit:
`http://dit.uvt.nl/`

logue processing. The features included in the data sets are those relating to *dialogue history*, *prosody*, and *word occurrence*.

For the AMI meetings and the DIAMOND dialogues, history consists of the tags of the 10 and 4 previous turns, respectively[6]. Additionally, the tags of utterances to which the utterance in focus was a direct response to, as well as timing, are included as features. For the data which is segmented per dimension, some segments are located inside other segments. This occurs for instance with backchannels and interruptions that do not cause turn shifting; the occurrence of these events is encoded as a feature.

Prosodic features that are included are minimum, maximum, mean, and standard deviation of *pitch* (F0 in Hz), *energy* (RMS), *voicing* (fraction of locally unvoiced frames and number of voice breaks), and *duration*. Word occurrence is represented by a bag-of-words vector[7] indicating the presence or absence of words in the segment. In total, 1,668 features are used for AMI data and 947 for DIAMOND data. For AMI data we additionally indicated the speaker (A, B, C, D) and the addressee (other participants individually or the group as a whole).

2.4 Classifiers

A wide variety of machine-learning techniques has been used for NLP tasks with various instantiations of feature-sets and target class encodings, and for dialogue processing, it is still an open issue which techniques are the most suitable for which task. We used three different types of classifiers to test their performance on our dialogue data: a probabilistic one, a rule inducer and memory-based learner.

For a probabilistic classifier we used *Naive Bayes*. This classifier assumes class-conditional independence, which does not always respect the characteristics of the features used. However, Naive Bayes classifiers often work quite well for complex real-world situations and are particularly suitable for situations in which the dimensionality of the input is high. Moreover, this classifier requires relatively little computation and can be efficiently trained.

For rule induction algorithm, we chose *Ripper* (Cohen, 1995). The advantage of such an algorithm is that the regularities discovered in the data are represented as human-readable rules.

The third classifier is *IB1*, which is a memory-based learner that is a successor of the k-nearest neighbour (k-NN) classifier. The algorithm first stores a representation of all training examples in memory. When classifying new instances, it searches for the k most similar examples (nearest neighbours) in memory according to a similarity metric, and extrapolates the target class from this set to the new instances. The algorithm may yield more precise results given sufficient training data, because it does not abstract away low-frequent phenomena during the learning (Daelemans et al., 1999).

The results of all experiments were obtained using 10-fold cross-validation[8]. When setting a baseline it is common practice to predict the majority class tag, but for our data sets such a baseline is not very useful because of the relative low frequencies of the tags in most dimensions. Instead, we use a baseline that is based on a single feature, namely, the tag of the previous dialogue utterance (see (Lendvai et al., 2003)).

3 Multidimensional dialogue act segmentation

Any segmentation of dialogue (or multi-party interaction) into meaningful units, such as functional segments, is motivated by the meaning that is conveyed. As a result, the segmentation strongly depends on the definition of the dialogue acts in the taxonomy that is used. The multidimensional tagset used in this paper allows several aspects of communicative behaviour for a single functional segment to be addressed. However, the functions of a segment do not necessarily address the same span in the communicative channels. Hence it could be argued that separate segmentation for each dimension should al-

[6] We use more preceding tags for the AMI data than for the DIAMOND data since there is often more distance between related utterances in multi-party interaction than in dialogue.

[7] With a size of 1,640 entries for AMI data and 923 for DIAMOND data.

[8] In order to reduce the effect of imbalances in the data, it is partitioned ten times. Each time a different 10% of the data is used as test set and the remaining 90% as training set. The procedure is repeated ten times so that in the end, every instance has been used exactly once for testing (Witten and Frank, 2000) and the scores are averaged. The cross-validation was stratified, i.e. the 10 folds contained approximately the same proportions of instances with relevant tags as in the entire dataset.

low for a more accurate identif cation of spans associated to specif c communicative functions. When we assume that this is the case, it would follow that classif cation of communicative functions based on per-dimension segments should be more successful than classif cation based on a single segmentation for all dimensions.

For testing the above-mentioned hypothesis, *Ripper* —the classif er that provides the highest accuracy scores in our experiments— was used on the DIAMOND dialogues annotated with the DIT^{++} tagset. Two classif cation tasks on exactly the same dialogues with exactly the same kind of features and annotated communicative functions were performed. The only difference being that in one task *one segmentation that f ts all dimensions (OSFAD)* was used, whereas in the other task *per-dimension segmentation (PDS)* was used. Because DIT allows the assignment of at most one function in a specif c dimension, a segment in the PDS task has one tag whereas a segment in the OSFAD setting might have a combination of tags[9]. Running *Ripper* (with default parameters) for both tasks resulted in the scores presented in Table 2:

Dimension	OSFAD	PDS	
Task	66.1	72.8	$*$
Auto Feedback	80.4	86.3	$*$
Allo Feedback	98.4	99.6	
Turn M.	88.3	90.0	
Time M.	72.6	82.1	$*$
Contact M.	97.3	97.3	
Topic M.	55.2	55.2	
Own Communication M.	85.9	87.1	
Partner Communication M.	64.5	64.5	
Dialogue Structuring	74.3	74.3	
Social Obligations M.	93.2	93.3	

Table 2: Accuracy scores for communicative functions with one segmentation that f ts all dimensions (OSFAD) and per-dimension segmentation (PDS).
$*$ signif cant at $p < .05$, one-tailed z-test.

From the results in Table 2 we can observe that for most important dimensions, PDS results in better classif cation performance: the functions related to the dimensions *Task*, *Auto Feedback*, and *Time Management* show signif cant improvement. For some dimensions, classif cation does not take advantage of PDS, mainly because of two reasons: in the dataset some dimensions are rarely addressed (e.g. *Partner Communication Management*) and some dimensions are addressed without any other dimension being addressed around the same time (e.g. *Contact Management*). These observations are motivated by the kinds and characteristics of interaction and in some extend by the limited size of the dataset.

Although not all dimensions benef t signif cantly, it is clear that multidimensional segmentation helps to classify communicative functions more accurately. However, it should be noted that the gain of more accurately identif ed functions comes at the cost of a slightly more complex segmentation procedure.

4 Dialogue Act Classif cation in Multiple Dimensions

Since a segment is often multi-functional, it is not only interesting to identify the dimension, the communicative function, and the tag separately, but also to test whether or not and to what extent it is possible to learn the combination of tags (e.g. $\langle TimeM, STALLING\rangle$, $\langle TurnM, KEEP\rangle$).

We carried out a set of experiments studying the performance of the three classif ers described in Section 2 on the following classif cation tasks:

- each addressed dimension separately or multiple addressed dimensions in combination, e.g. a single dimension like *Task*, *Auto-Feedback*, *Turn Management*, or a combination like *Turn Management* and *Time Management*;
- communicative function per dimension in isolation, e.g. INFORM, CORRECTION, WH-QUESTION, etc. in the *Auto-Feedback* dimension;
- tag or combination of tags, e.g. either $\langle D, GP\rangle$ or $\langle D, DS\rangle$, or $\langle D, GP\rangle$,$\langle D, DS\rangle$ or $\langle D, DS\rangle$,$\langle D, DS\rangle$.

4.1 Experimental results

Table 3 gives an overview of classif cation scores expressed as the percentage of correctly predicted classes in all training experiments.

For the prediction of a dimension addressed by a functional segment (upper data row in the table)

[9] In our data, at most four functions occurred simultaneously.

Classif cation task	BL	NBayes	Ripper	IB1
Dimension tag	38.0	69.5	**72.8**	50.4
Task management	66.8	71.2	**72.3**	53.6
Auto-Feedback	77.9	86.0	**89.7**	85.9
Turn initial	93.2	92.9	93.2	88.0
Turn closing	58.9	85.1	**91.1**	69.6
Time management	69.7	99.2	**99.4**	99.5
OCM	89.6	90.0	**94.1**	85.6
Functional tag	25.7	48.0	**50.2**	38.9

Table 3: Overview of accuracy on the baseline (BL) and the classif ers on all classif cation tasks

all algorithms outperform the baseline by a broad margin. Ripper clearly outperforms the other two learners. The middle part of the table gives an overview of the performance of the tested classif ers on communicative functions per dimension. *Ripper* again outperforms *Naive Bayes* and *IB1*. The scores are the same (e.g. with turn initial functions) or higher then those of the baseline. Some of the dimensions distinguished in DIT are not included in Table 3 since the segments which were tagged as having communicative functions in the dimensions *Allo-feedback*, *Contact management*, *Topic management*, *Dialogue Structuring*, *Partner Communication management*, and *Social Obligation Management* are rare in the AMI training data. The instances from these dimensions were almost perfectly classif ed by all classif ers, reaching an accuracy higher than 99%, but not better than those of the baseline.

In Appendix A of this paper we present a selection of the RIPPER induced rules illustrated with examples from the corpus. As was to be expected, for the prediction of the *Task* dimension, the bag-of-words feature representing word occurrence in the segment was important. For example, the presence of 'because' in a segment was a good indicator for identifying INFORM JUSTIFY; the occurrence of 'like', or 'for example', or 'maybe' and 'might' for SUGGESTION. Also the duration of the segment was usually longer than for example segments which addressed the *Time* or *Turn Management* dimensions. For the prediction of questions, word occurrence (e.g. occurrence of wh-words in WH-Questions, and 'or' for Alternative Questions) and prosodic features like standard deviation in pitch were essential. For the segments which are identif ed as having Information-Providing functions, important features were detected in the dialogue history, e.g. CONFIRM about the task was a response to a previous CHECK question about the task. The segments addressing the *Auto-Feedback* dimension were classif ed successfully on the basis of their word occurrence and dialogue history. The occurrence of words like 'alright', 'right', 'okay', 'uh-huh' are important clues for their recognition.

As for the dimensions *Turn* and *Time Management*, the duration of the segment was a key feature, because the duration of these segments tends to be shorter than that of others. Moreover, these utterances were pronounced more softly (e.g. $<$49dB) and are less voiced (e.g. about 47% of unvoiced frames). They usually occur inside 'larger' segments, mostly in the beginning or in the middle. If they appear in clause-initial position, they usually have turn initial functions (TAKE, ACCEPT, GRAB) and the function STALLING in the *Time Management* dimension; if they occur in the middle of the 'main' segment they are used to signal that the speaker has some diff culties in completing his/her utterance, needs some time and wants to keep the turn (see examples 3 and 5). Of course, usage of words like 'um', 'well', but also lengthening the words indicates the speaker's hesitation and/or diff culties in utterance completion.

Segments having communicative functions in the dimension *Dialogue Structuring* often have linguistic cues like 'meeting', 'f nish', 'wrap up', etc. Important cues for RETRACTs (in the dimension *Own Communication Management*) are their relation to what is actually retracted ('reply_to' feature), and the energy with which they are spoken (i.e. they are pronounced louder than the retracted 'reparandum', i.e. $>$55dB).

Looking further at the results we can observe that tag labels were diff cult to classify (see bottom data row of the table). They eventually reach an accuracy of 50.2% (baseline: 25.7%). These scores should be evaluated in the light of the relatively high degree of granularity of these tags (97 unique tags and 132 unique combinations of tags) and relatively lower frequency of each of those in the training sets. We have however reason to expect that by increasing the size of the training set higher accuracy could be reached.

5 Conclusions and future work

In this paper a multidimensional approach to utterance segmentation and automatic dialogue act classif cation has been presented in which some problematic issues with the segmentation of dialogue into functional units are addressed.

Whereas it is common practice to assign dialogue acts to a single segmentation, we conclude that for dialogue act taxonomies that allow assignment of multiple functions to dialogue units we can describe human communication more accurately by using per-dimension segmentation instead.

We have shown that machine learning techniques can be prof tably used on a complex task such as the automatic recognition of multiple communicative functions of dialogue segments. All three classif ers that have been tested performed well on all classif cation tasks. For the majority of tasks, the scores we obtained are signif cantly higher than those of the baseline. However, the datasets that we used were not very rich with respect to all the communicative functions distinguished in the various dimensions: some classes were underrepresented.

For future work, we intend to extended the studies into two directions. First, we plan to increase the size of our dataset to obtain a suff cient number of instances for each class by manually segmenting and annotating more dialogue data with both segmentations. This would allow us to get a fair indication of the classif cation performance of general purpose functions in dimensions other than *Task* and *Feedback*. Furthermore, we plan to consider multi-party interactions (the AMI sessions for instance) and use other modalities besides speech audio in comparing both segmentations. We expect that for such data, dialogue act classif cation may benef t more from using per-dimension segmentation.

References

Jens Allwood. 2000. An activity-based approach to pragmatics. In Harry Bunt and William Black, editors, *Abduction, Belief and Context in Dialogue; Studies in Computational Pragmatics*, pages 47–80. John Benjamins, Amsterdam, The Netherlands.

Harry Bunt and Amanda Schiffrin. 2007. Def ning interoperable concepts for dialogue act annotation. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS)*, pages 16–27.

Harry Bunt. 2006. Dimensions in dialogue annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.

Alexander Clark. 2003. Machine learning approaches to shallow discourse parsing: A literature review. IM2.MDM Project Deliverable, March.

William W. Cohen. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, pages 115–123.

Mark G. Core and James F. Allen. 1997. Coding dialogues with the DAMSL annotation scheme. In *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1/3):11–43.

Jeroen Geertzen, Yann Girard, and Roser Morante. 2004. The diamond project. Poster at the 8th Workshop on the Semantics and Pragmatics of Dialogue (CATALOG 2004)

Piroska Lendvai, Antal van den Bosch, and Emiel Krahmer. 2003. Machine learning for shallow interpretation of user utterances in spoken dialogue systems. In *Proceedings of EACL-03 Workshop on Dialogue Systems: interaction, adaptation and styles of management*, pages 69–78.

Elmar Nöth, Anton Batliner, Volker Warnke, Johannes-Peter Haas, Manuela Boros, Jan Buckow, Richard Huber, Florian Gallwitz, Matthias Nutt, and Heinrich Niemann. 2002. On the use of prosody in automatic dialogue understanding. *Speech Communication*, 36(1-2):45–62.

Volha V. Petukhova and Harry Bunt. 2007. A multidimensional approach to multimodal dialogue act annotation. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS)*, pages 142–153.

Elizabeth Shriberg, Rebecca Bates, Andreas Stolcke, Paul Taylor, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 1998. Can prosody aid the automatic classif cation of dialog acts in conversational speech? *Language and Speech (Special Issue on Prosody and Conversation)*, 41(3-4):439–487.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Ian H. Witten and Eibe Frank. 2000. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers, San Francisco:CA, USA.

Appendix A: Selected RIPPER rules illustrated with corpus examples

The structure of a rule is: if (feature = x) and (feature= x, etc.) $\Longrightarrow$ class (*n/m*), where x is a nominal feature value, an element of a set feature, or a range of a numeric feature; *n* indicates the number of instances a rule covers and *m* the number of false predictions. We illustrate the induced rules with some interesting examples from the training set.

Task Management:
(it = p) and (wouldnt = p) $\Longrightarrow$ da=task:check (5.0/1.0)
(right = p) and (max.pitch <= 203.87) $\Longrightarrow$ da=task:check (8.0/2.0)
Example:
(1052:88-1057:12) D: We were given sort of an example of a coffee machine or something, right? (dimension: Task, GP:CHECK; FT: *task:check*)

(reply_to = task;ynq) $\Longrightarrow$ da=task:yna (60.0/22.0)
(reply_to = task;ynq;t_give) $\Longrightarrow$ da=task:yna (2.0/0.0)
(reply_to = task;ynq;t_grab) $\Longrightarrow$ da=task:yna (2.0/0.0)
(reply_to = task;ynq;t_release) $\Longrightarrow$ da=task:yna (3.0/1.0)
Example:
(1407:56-1413:72) B: Do you think maybe we need like further advances in that kind of area until it's worthwhile incorporating it though (dimension:Task; GP: YN-QUESTION; FT: *task:ynq*)
(1412:96-1415:6) C: I , think , it'd , probably , quite , expensive , to , put , in (dimension:Task; GP: YN-ANSWER; FT: *task:yna*)

(yeah = p) and (dss_reply <= -3.920044) and (duration >= 0.56) and (min.pitch >= 95.007) $\Longrightarrow$ da=task:inf.agree (27.0/8.0)
(yeah = p) and (fraction:voiced/unvoiced >= 0.36634) and (dss_reply ¡= -0.52002) and (fraction:voiced/unvoiced <= 0.46875) $\Longrightarrow$ da=task:inf.agree (8.0/1.0)
(yeah = p) and (energy >= 56.862651) and (mean.pitch <= 144.971) $\Longrightarrow$ da=task:inf.agree (9.0/2.0)
(dss_reply <= -0.359985) and (sure = p) and (max.pitch <= 187.065) $\Longrightarrow$ da=task:inf.agree (8.0/0.0)
(yeah = p) and (U3 = turn:t_keep;time:stal) $\Longrightarrow$ da=task:inf.agree (14.0/6.0)
Example:
(1277:88-1286:28) D: but people who are about forty-ish and above now would not be so dependent and reliant on a computer or mobile phone (dimension:Task; GP:INFORM; FT:*task;inf*)
(1284:32-1286:16) D: Yeah, sure (dimension: Task; GP:INFORM AGREEMENT; FT: *task:inf.agree*)

(problem = p) $\Longrightarrow$ da=task:inf.warn (7.0/3.0)
(because = p) $\Longrightarrow$ da=task:inf.just (33.0/7.0)
(cause = p) $\Longrightarrow$ da=task:inf.just (26.0/9.0)
(dss_reply <= -1.52002) and (voice_breaks >= 4) and (energy >= 54.435098) and (mean.pitch <= 173.572) $\Longrightarrow$ da=task:inf.ela (51.0/21.0)
Example:
(1396:84-1403:76) C: One problem with speech recognition is the technology that was in that one wasn't particularly amazing (dimension: Task; GP: INFORM WARNING; FT: *task:inf.warn*)

(maybe = p) and (dss_reply >= 0) $\Longrightarrow$ da=task:suggest (38.0/11.0)
(duration >= 2.12) and (reply_to = _) and (might = p) $\Longrightarrow$ da=task:suggest (12.0/4.0)
Example:
(1694:6-1703:48) B: It might be a good idea just to restrict our creative inf uence on this and not worry so much about how we transmit it (dimension:Task; GP: SUGGESTION; FT:*task;suggest*)
(1704:4-1708:44) B: because I mean it tried and tested intra-red (dimension:Task; GP: INFORM JUSTIFY; FT:*task:inf.just*)

Auto-Feedback:
(dss_reply <= -0.039978) and (break <= 1) $\Longrightarrow$ da=au_f:au_f_p_ex (168.0/24.0)
(dss_reply <= -0.039917) and (duration <= 1.08) and (okay = p) $\Longrightarrow$ da=au_f:au_f_p_ex (84.0/8.0)
(dss_reply <= -0.039978) and (break <= 1) and (mmhmm = p) $\Longrightarrow$ da=au_f:au_f_p_ex (34.0/1.0)
(dss_reply <= -0.039978) and (break <= 3) and (voclaugh = p) $\Longrightarrow$ da=au_f:au_f_p_ex (25.0/2.0)
(okay = p) and (energy <= 56.617891) and (duration >= 1.16) $\Longrightarrow$ da=au_f:au_f_p_ex (21.0/4.0)
Example:
(1728:36-1729:88) A: Then you need to send the signal out (dimension: Task; GP:INFORM; FT:*task:inf*)
(1729:8-1730:2) B: Mmhmm (dimension: Auto-Feedback; DS: POS.EXECUTION; FT: *au_f:au_f_p_ex*)

(within = turn:t_keep;time:stal) and (duration <= 0.44) $\Longrightarrow$ da=au_f:au_f_p_ex;turn:t_give (83.0/11.0)
(within = turn:t_keep;time:stal) and (energy <= 50.235299) $\Longrightarrow$ da=au_f:au_f_p_ex;turn:t_give (9.0/2.0)
Example:
(1285:32-1292:36) B: you're gonna have audio which is gonna be like you know
B: um and (dimension:Time/Turn; DS: STALLING/T_KEEPING; FT: *turn:t_keep;time:stal*)
(1289:44-1290:08)A: mmhm (dimension: Auto-Feedback/Turn; DS: POS.EXECUTION/T_GIVING; FT: *au_f:au_f_p_ex;turn:t_give*)
B: your bass settings and actual volume hi

Turn Management:
(um = p) and (dss_reply <= -1.199997) ⟹ da=turn:t_acc;t_keep;time:stal (13.0/6.0)
(well = p) and (dss_within <= -0.159912) and (duration <= 0.72) ⟹ da=turn:t_grab;t_keep (9.0/3.0)
(um = p) and (dse_within >= 0.040039) and (dse_within <= 1.040039) and (min.pitch >= 107.875) ⟹ da=turn:t_grab;t_keep;time:stal (18.0/4.0)
(well = p) and (dss_within <= -1.119995) ⟹ da=turn:t_grab;t_keep;time:stal (6.0/2.0)
(um = p) and (dse_within <= 0) and (energy <= 49.86226) and (mean.pitch >= 114.669) ⟹ da=turn:t_take;t_keep;time:stal (21.0/10.0)
Examples:
(819:08-821:88) D: Well like um (dimension: Turn/Time; DS:T_GRABBING/STALLING; FT: *turn:t_grab;t_keep;time:stal*)
D: maybe what we could use is a sort of like a example of a successful other piece technology is palm pilots

Topic Management:
(back = p) and (go = p) ⟹ da=topic:suggest (5.0/2.0)
Example:
(1587:16-1591:72) A: I guess we should maybe go back to what the functions are (dimension: Topic Management; GP: SUGGESTION; FT:*topic:suggest*)

Dialogue Structuring:
(end = p) and (min.pitch >= 175.915) ⟹ da=ds:inf (2.0/0.0) (wrap = p) and (U3 = au_f:au_f_p_ex) ⟹ da=ds:inf (2.0/0.0)
Examples:

(978:6- 981:68) D: so just to wrap up the next meeting's gonna be in thirty minutes (dimension: Dialogue Structuring; GP:INFORM; FT: *ds:inf*)
(1036:44-1037:68) B: And that's the end of the meeting (dimension: Dialogue Structuring; GP:INFORM; FT: *ds:inf*)

Contact Management:
ready = p) ⟹ da=contact:check (2.0/0.0)
Example:
(34:06-35:56) B: All ready to go? (dimension: Contact Management; GP: Check; FT: *contact:check*)

Own Communication Management:
(oh = p) ⟹ da=ocm:error (7.0/3.0)
(reply_to = time;t_keep;stal) and (duration >= 0.36) and (U5 = turn:t_keep;time:stal) ⟹ da=turn:t_keep;ocm:retract (12.0/5.0)
(reply_to = time;t_keep;stal) and (energy >= 55.581619)⟹ da=turn:t_keep;ocm:retract (185.0/17.0)
(dse_within >= 0.679993) and (duration <= 0.24) and (min.pitch >= 107.013) and (max.pitch <= 155.745) and (mean.pitch >= 122.459) ⟹ da=turn:t_keep;ocm:retract (17.0/4.0)
Example:
(96:32-96:68) B: Oh (dimension: Own Communication Management; DS: Error; FT: *ocm:error*
B: I have to record who's here actually

Social Obligation Management:
(thanks = p) ⟹ da=som:thanking (2.0/0.0)
(reply_to = som;ini_self ntro) ⟹ da=som:react_self ntro (4.0/1.0)
Examples:
(72:8-74:44) B: I'm Laura and I'm the project manager (dimension: Social Obligation Management; DS: INITIATE SELF-INTRODUCTION; FT:*som;ini_self ntro*)
(77:44-77:76) A: I'm David and I'm supposed to be an industrial designer(dimension: Social Obligation Management; DS: REACT SELF-INTRODUCTION; FT:*som;react_self ntro*)

Accented Pronouns and Unusual Antecedents: A Corpus Study

Anubha Kothari
Department of Linguistics
Stanford University
Stanford, CA 94305-2150
anubha@stanford.edu

Abstract

Accent on a pronoun has often been assumed to signal an "unusual" antecedent, i.e. something other than the most salient compatible antecedent. However, this assumption has not received adequate empirical investigation to date, and in particular, spontaneous conversational dialogues have never been studied to verify the salience-based proposals. I analyze a richly annotated corpus of naturalistic speech, manually labeled for coreference relations, accents, and contrast, in order to understand what factors govern the presence of accent on a pronoun and thereby gain insight into what pronominal accent may be communicating. The results suggest that not only are differences among speakers and pronouns key components in explaining the variation in pronominal accentuation, but also that pronominal accent may often be signaling contrast rather than something about the attentional status or salience of the pronoun's referent.

1 Introduction

One phenomenon in which prosody is often assumed to play a disambiguating role is anaphora resolution. In particular, many have proposed that the presence of accent on a pronoun is a signal that the pronoun has an "unusual" antecedent, not the maximally salient compatible discourse referent, which is what an unaccented pronoun would normally refer to (Ariel, 1990; Cahn, 1995; Gundel et al., 1993; Kameyama, 1999; Nakatani, 1997). The following pair illustrates this reference-switching effect:[1]

(1) a. John hit Bill. Then he hit Mary. (*John hit Mary.*)
b. John hit Bill. Then HE hit Mary. (*Bill hit Mary.*)

In (1a), *John* is the topic or maximally salient entity from the first sentence by virtue of being subject, and serves as the continuing topic and referent of *he* in the second sentence. In (1b), however, the accent signals a topic shift, indicating that the pronoun's referent is lower in a saliency-ranked list of entities, which in this case forces it to be *Bill*. Thus, under such theories, the choice of an accented pronoun as a referring expression is linked directly to the attentional or cognitive status of the referent.

However, aside from analysis of short constructed discourses such as these, the attentional theories have seen very little empirical evaluation using longer, naturally-produced discourses. For applications that produce or comprehend naturalistic speech, it is crucial to understand what is being communicated by accent on a coreferential pronoun and what factors govern its presence, thereby also testing whether accent truly is a robust cue to "unusual" resolution. For instance, accent may instead be conveying contrast between the actual referent and the expected referent, a potential confound that has not been investigated in any study.

In this paper, I address these questions using a richly annotated corpus of spontaneous conversational speech, manually labeled for coreference relations, pitch accent, and contrast. Using logistic regression, I explore the usefulness of various factors

[1] Capitals indicate a pitch accent, in all examples here.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 150–157,
Antwerp, September 2007.

reflecting properties of the antecedent or of the pronoun itself in predicting the presence of accent on coreferential pronouns. Previous studies addressing these same questions have not had access to as large and ideal a corpus as the one I make use of here, and relatedly, have ignored various factors or could not arrive at statistically significant conclusions.

The rest of the paper is structured as follows: In Section 2, I summarize the relevant theoretical and experimental work to date on how prosodic prominence may affect anaphoric relationships. Section 3 describes the corpus and the features extracted from it. Section 4 presents the statistical models using these features and their analysis. The fifth section discusses the results more generally, and the final section lists the conclusions of this study.

2 Background

Accents tend to accompany information that is new in the discourse (Brown, 1983), so their presence on pronouns, words that stand for given and highly accessible information, is surprising. This has led to the hypothesis that pronominal accent is somehow special, and (based on examples like the one above) that it has an attentional function. For instance, in Gundel et al.'s (1993) Givenness Hierarchy, stressed pronouns are said to align with a referent that is 'activated' but not 'in focus'. Centering Theory (Grosz et al., 1995), however, has been the primary framework in which accented pronouns have been examined. Within this approach, Kameyama (1999) proposes that at least two linguistic hierarchies are relevant in ranking entities for salience: more salient entities are realized by a higher-ranked grammatical function (Subject $>$ Object $>$ Object2 $>$ Others) or a higher-ranked expression type (Zero Pronominal $>$ Pronoun $>$ Definite NP $>$ Indefinite NP). If so, subjecthood and pronominality should be two important properties in determining whether an antecedent is "unusual".

Watson et al. (2006) tested the first of these hierarchies in a controlled experiment, and found that speakers do produce NPs like *the bed* with acoustic prominence in mini-discourses like the following where the first mention has a lower-ranked grammatical function: *Put the house above the bed. Put the BED above the pineapple.* They concluded that a shift in attentional salience plays a role. However, in a production experiment, Wolters and Beaver (2001) found that although speakers generally accented subject pronouns having object antecedents, the effect was very weak. The main problem appeared to be that the speaking styles varied considerably, from monotonous intonation contours to very natural ones. They also analyzed news stories read by 3 speakers who contributed 122, 22, and 8 pronouns respectively. They found a significant relationship between antecedent pronominality and accentuation of a pronoun, but most of the accented pronouns could also be analyzed as cuing some sort of contrast. Further work is necessary to understand the role of contrast, especially as the meaning effects of accent on pronouns might be no different than to evoke contrast within a contextually salient set of alternatives (De Hoop, 2003). Moreover, sparse and unequally distributed data meant that speaker effects could not be investigated rigorously in either study.

Wolters and Byron (2000) studied a larger corpus of task-oriented spontaneous dialogues from a total of 16 speakers (although the data are highly unbalanced as two of the speakers contribute 48% of the data). They found no correlations in their logistic regression experiments between acoustic prosodic properties of the pronoun and various properties of the antecedent, but SPEAKER was a significant factor in most of their models. They too concluded that inter-speaker variation gets in the way of safe generalizations and that different speaker types need to be understood. However, they included SPEAKER as a fixed-effect rather than a random-effect in their models which means they assumed that their speakers represent 16 repeatable and fixed levels of a factor. That is, they incorrectly assume that the speakers are mutually exclusive and exhaustive in representing speaker-types in the population.

3 The Corpus and Features

I use 19 dialogues of the Switchboard corpus of spontaneous phone conversations (Godfrey et al., 1992) that have manual annotations for the presence or absence of pitch accent on each word (Ostendorf et al., 2001; Calhoun, 2006), "kontrast" relations (Calhoun et al., 2005), and coreference links (Nissim et al., 2004). All non-demonstrative

he	*her*	*him*	*his*	*it*
56	19	17	13	303
its	*she*	*their*	*them*	*they*
2	56	37	101	230

Table 1: Number of instances of each pronoun-type

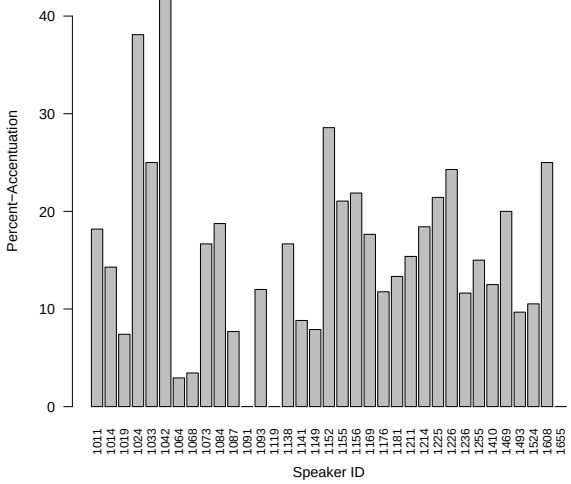


Figure 1: Variation between speakers

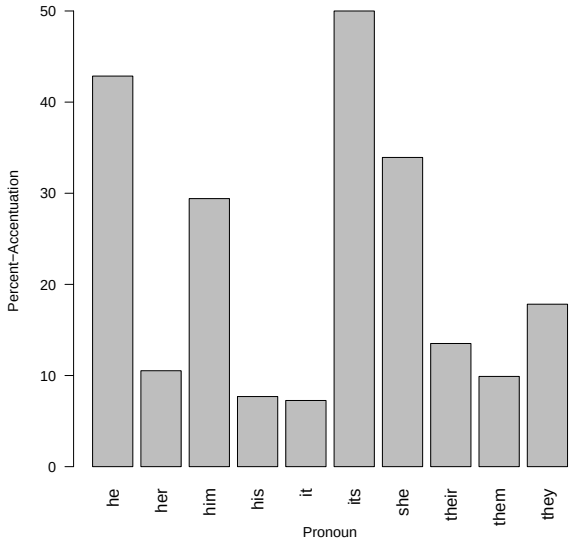


Figure 2: Variation between pronouns

third-person coreferential pronouns were isolated for analysis, a total of 834 pronouns of which 15.6% bear a pitch accent.[2] The pronoun-types and their frequencies are given in Table 1; all pronouns were made case-insensitive and stripped of bound reduced verbs. A total of 35 speakers of American English contributed the pronouns, 22 females and 13 males, with a fairly balanced division of the pronouns amongst them.

One striking aspect of these data is that both speakers and pronouns exhibit great variation in accentuation. Figures 1 and 2 illustrate this, motivating some of the factors included in the models below.

A number of attributes of each pronoun and its antecedent were calculated or extracted from the annotations. These features are described in Table 2, and assigned to three groups.

The first set of features reflect antecedent properties that could be useful in detecting a relationship between "unusual" antecedents and `accent` (the presence or absence of accent on a pronoun). The two referential distance features are discourse measurements of topic continuity inspired by Givón (1983).[3] All features in this group bear a non-significant relationship to `accent` on the basis of chi-square and correlational tests, except for `antecDistCat` ($p < .05$).

The second group has features that capture properties of the pronoun itself. The first two are primarily intended as control factors, in case disfluencies or reductions (as in *he's*) behave in a non-standard manner. Chi-square tests revealed that these factors do not have a significant relationship with `accent` (both $p > 0.1$). On the other hand, the last three features are all significantly related to `accent` (all $p < .001$). Subject pronouns are

[2]English reflexives may bear an emphatic function and are disyllabic, differentiating them from other pronouns, so they were excluded. There were only 14 tokens in all.

[3]A log-transformed distance metric was also explored but abandoned. Log-distance had a bimodal distribution while "regular" distance was skewed but unimodal. What the exploration did highlight is that most antecedents are in the same or adjacent clause, thus motivating the categorical distance metric I use.

said to continue topics, i.e. to refer to attentionally salient discourse referents, so pronoun-subjecthood could interact with other factors in an important way. The two `kontrast` features need much elaboration. A "kontrast" introduces a presupposition of alternatives to the kontrasted word in the discourse context, thereby making it informationally salient. The feature `kontrast` reflects the reason or trigger for this salience (Calhoun et al., 2005). Several types of triggers were marked, but the two values which are of most concern here are: *contrastive*, for when the word is directly contrasted with a previous topical, semantically-related word, and *background* when the word is not intended to be salient. The second kontrast feature is a binned version that lumps together all values of `kontrast` except for `background` and `contrastive`; I created this to test particular kontrast-related conclusions below.

Finally, in the third group is `spkrAccentRate`, a rough approximation of speaker styles based on the percentage of pronouns a speaker accented. It is a continuous measure that represents "styles" ranging from "monotonous" speakers with low rates of accentuation to animated or expressive speakers who accent plenty of their pronouns. Naturally, this feature is significantly related to `accent` ($p \approx 0$).

4 Analysis and Results

The general strategy adopted here is to test the usefulness of the features above by testing them simultaneously in logistic regression models predicting `accent`. In addition to these fixed-effects factors, I depart from previous studies in also including two random-effects factors, namely `Speaker` and `Pronoun`. Given the enormous inter-speaker and inter-pronoun variation demonstrated in Figures 1 and 2, it is essential to check for these dependencies; treating them as random-effects in mixed-effects logistic regression models is the most appropriate modeling technique here as then we do not tailor our models to the specific speakers and pronouns in the study but instead assume they are randomly sampled levels from a much larger population of interest. Within this setup I carry out two sets of studies, the first on all the pronouns isolated for analysis, and the second on only those pronouns with antecedents in adjacent clauses since these constitute the type of 2-utterance discourses discussed most in the literature. I use the `lme4` and `Design` packages in R (Ihaka and Gentleman, 1996).

4.1 All Coreferential Pronouns

Variable Selection: In order to inspect the variables and select which ones to include in the final models, I use a regular logistic regression model to predict `accent` using all the fixed-effects factors in Table 2 (except for `kontrastBinned`). Fast backward elimination, a routine that deletes irrelevant factors by comparing the AIC model fit value of the full model against that of a reduced model lacking the factor being tested, retains only `pronIsSubj`, `kontrast`, and `spkrAccentRate`. So all the other fixed-effects factors, including antecedent properties and control factors `disfluency` and `cliticized` do not improve the quality of a model predicting `accent`. In the models that follow I do not include the control factors, though I retain the factors having to do with antecedents since these are of primary interest in this study.

I construct three kinds of models – (i) a fixed-effects-only model with only the selected fixed-effects, (ii) one using only the two random-effects factors, and (iii) a generalized linear mixed model that uses the two random-effects and the significant fixed-effects predictors from the first model (except `kontrastBinned` is substituted for `kontrast`).

Fixed-effects-only Model: I use a regular logistic regression model to predict `accent` using only `kontrast`, `spkrAccentRate`, and the factors related to antecedents. The VIF values of these factors range between 1 and 2.19 (all much lower than 10), so there is no danger of collinearities among the predictors. The coefficients, standard errors, and p-values for the different levels of these factors are reported in Table 3.[4] The quality of the model is modest (concordance C, a measure of model discriminability, is 0.741), but this is not surprising given that there are probably many other factors needed to predict accent placement, including speaker and pronoun variation. What is more interesting is that none of the factors related to unusual antecedents are significant, while `pronIsSubj`, `spkrAccentRate` and contrastive `kontrast` are significant (all $p <$

[4] p-value sig. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'

Feature	Description	Possible Values
antecIsSubj	Antecedent is in subject position of its clause	yes, no
antecIsPro	Antecedent is pronominal	yes, no
antecDistCont	Distance *n* to antecedent in number of clauses	$0 \leq n \leq 84$
antecDistCat	Location of antecedent clause relative to pronoun	same, adjacent, remote
disfluency	Disfluency characteristic of pronoun	none, repair, reparandum
cliticized	Pronoun followed by reduced verb	yes, no
pronIsSubj	Pronoun is in subject position of its clause	yes, no
kontrast	Reason for informational salience	adverbial, answer, background, contrastive, nonapplic, other, subset
kontrastBinned	Binned version of kontrast	background, contrastive, other
spkrAccentRate	Speaker's propensity to accent pronouns	0% - 38%

Table 2: Feature descriptions and values

	Coef.	S.E.	P-value
Intercept	-3.7401	0.3668	0.0000
antecIsSubj=yes	-0.1781	0.2207	0.4197
antecIsPro=yes	0.3105	0.2375	0.1912
antecDistCont	-0.0552	0.0469	0.2390
antecDistCat=adjacent	0.4608	0.2629	0.0797
antecDistCat=remote	0.0544	0.3277	0.8682
pronIsSubj=yes	0.6958	0.2353	0.0031 *
kontrast=adverbial	10.7832	79.0508	0.8915
kontrast=contrastive	2.0066	0.4683	0.0000 ***
kontrast=nonapplic	1.1516	0.7894	0.1446
kontrast=other	10.3226	45.3334	0.8199
kontrast=subset	-0.0635	1.0864	0.9534
spkrAccentRate	6.9339	1.1478	0.0000 ***

Table 3: Fixed-effects-only; all pronouns

.01). Their coefficients indicate that, as expected, the higher a speaker's propensity to accent pronouns the higher the log-odds of the pronoun being accented, and likewise if the pronoun is contrastive rather than backgrounded and if it is in subject-position rather than not. To check for overfitting I run bootstrap validation, and almost all runs remove all predictors other than pronIsSubj, kontrast and spkrAccentRate. The number of runs in which all seven predictors are retained is extremely small (4), so the model cannot be overfitting the data. Using penalized maximum likelihood estimation to discourage large values for the coefficients due to potentially extreme data points, I find that although all coefficients are slightly shrunk towards zero, the same factors remain significant. Finally, since only *contrastive* kontrast is a significant predictor within kontrast, it would make sense to bin kontrast into background, contrastive and other, as I do below.

Random-effects-only Model: Next, I build a model with only the random-effects factors and find that this has a concordance of 0.723, slightly lower than the previous model but still decent, suggesting that inter-speaker and inter-pronoun variation could be critical components in determining accent placement on coreferential pronouns.

Mixed-effects Model: Finally, I build a mixed-effects model that has the two random-effects as well as pronIsSubj, kontrastBinned and spkrAccentRate as fixed-effects. This model has the highest concordance so far (0.761), so this combination of fixed- and random-effects factors leads to a model of better quality. Comparisons of fuller models to smaller sub-models using the difference of their log likelihoods reveals that only kontrastBinned, spkrAccentRate, and the Pronoun random-effect were significant, justified factors. Table 4 lists for each factor, the difference in log likelihood between the full model and a reduced model lacking that factor, as well as the *p*-value for the factor. From these results, it appears that kontrast and speaker style are beneficial in predicting accent and inter-pronoun variation is an important dependency as well. It appears that it could be useful to understand different speaker strategies or styles, beyond the crude metric used here; inter-speaker differences at an individual level are not as useful to study as seen by the insignificant Speaker random-effect. Since none of the antecedent properties were significant, these models could not verify that accent on a coreferential pro-

Factor	ΔlogLik	P-value
pronIsSubj	0.92	0.1747
kontrastBinned	9.37	8.528e-05 ***
spkrAccentRate	16.89	6.184e-09 ***
Speaker	0	0.9936
Pronoun	8.53	3.616e-05 ***

Table 4: Mixed-effects; all pronouns

	Coef.	S.E.	P-value
Intercept	-4.3109	0.6491	0.0000
antecIsSubj=yes	-0.5193	0.3858	0.1784
antecIsPro=yes	0.6126	0.4203	0.1449
pronIsSubj=yes	0.9163	0.4392	0.0369 .
kontrast=contrastive	3.3118	1.2220	0.0067 *
kontrast=nonapplic	3.2669	1.3964	0.0193 .
kontrast=other	9.5187	45.0338	0.8326
kontrast=subset	8.3827	45.0328	0.8523
spkrAccentRate	10.4269	2.1914	0.0000 ***

Table 5: Fixed-effects-only; adjacent antecedents

noun signals anything about antecedents, at least not in the presence of these other significant factors.

4.2 Only Adjacent Antecedents

Here I limit the dataset to only those pronouns with antecedents in the previous clause, thus reproducing the 2-utterance-scenario often discussed by the attentional theories. This dataset has only 257 pronouns. Still, most of the results of the larger dataset are again found to be valid here.

First, logistic regression without any random-effects (and without antecedent-distance metrics since distance is constant here) produces a model with a fairly good concordance of 0.805, and significant pronIsSubj, contrastive kontrast and spkrAccentRate again. Table 5 lists the coefficients, *p*-values etc. for the different levels of the various factors. Again, fast backward elimination only retains the last three factors, and none of the saliency-based antecedent factors.

However, the influence of speaker and pronoun variation is less clear here, perhaps due to the smaller size of this dataset. A model with only the two random-effects has a concordance of 0.745, which is close to but slightly lower than that of the fixed-effects only model. A full mixed-effects model with the two random-effects and pronIsSubj, kontrastBinned and spkrAccentRate fixed-effects has about the same concordance as the fixed-effects-only model for this dataset, namely 0.795, but model comparisons show only kontrastBinned and spkrAccentRate to be significant (see Table 6).

Factor	ΔlogLik	P-value
pronIsSubj	1.33	0.1026
kontrast	8.73	0.0002 ***
spkrAccentRate	12.28	0.0000 ***
Speaker	0	0.9766
Pronoun	0.16	0.5703

Table 6: Mixed-effects; adjacent antecedents

Unaccented	antecIsPro = 0	antecIsPro = 1
antecSubj = 0	**24**	24
antecSubj = 1	17	54
Accented	antecIsPro = 0	antecIsPro = 1
antecSubj = 0	8	9
antecSubj = 1	3	**22**

Table 7: Subject pronouns with adjacent antecedents

5 Discussion

On the whole, the results suggest that pronominal accent may often be signaling contrast rather than something about the attentional status of the pronoun's referent. At the very least, we have no evidence that topic shift is being signaled via accent in spoken conversational speech. Instead, the recurring theme is that speaker-propensities, pronoun identities, and contrast-status, will go a long way in predicting whether a speaker will produce a particular pronoun with an accent.

The two-utterance discourses studied theoretically or via controlled experiments do seem to intuitively support an attentional/saliency-ranking account, but actual naturalistic productions often do not accord with such an account. Consider the accent distribution among just subject pronouns in this corpus which have antecedents in the previous clause, given in Table 7. In spite of topic-discontinuity, 24 pronouns do not bear accent; and in spite of topic continuity, 22 pronouns bear an accent, counter to the predictions of the attentional story.

Here, for instance, is an example of a subject pronoun *she* which was accented by a *high* accenting speaker, even though the antecedent in the adjacent clause is pronominal and in subject position:[5]

(2) Well, UM Y- you MENTIONED your DAUGHTER had graduated from COLLEGE. WELL, when SHE was in high SCHOOL did SHE always HAVE to have all the new FASHIONS?

And here is a long stretch of discourse in which the same speaker accents *he* nearly every time, even though the referent, her brother, is clearly the continuing topic throughout:

(3) i MEAN UH MY BROTHER works for TI and HE'S a computer PROGRAMMER or computer ENGINEER. AND YOU know whenever HE was going to school HE was EXPECTING to HAVING to wear uh a TIE or a DRESS shirt EVERYDAY. BUT UH he GOES to WORK in HIS blue JEANS T-SHIRT and TENNIS shoes. And HE just LOVES it.

However, it isn't always the case that subject pronouns get accented. Here is a more "balanced" production by the same speaker with unaccented *she*:

(4) But she NEVER would BUY me like the NEW designer JEANS that had come OUT that were THIRTY dollars or UM or she wouldn't BUY me the FIFTY dollar TENNIS shoes and stuff like THAT.

On the other hand, the following is an accented subject pronoun *he* with an adjacent antecedent, labeled as 'contrastive' because the referent, the speaker's dad, is being compared to people who do not care about the environment:

(5) Well, my DAD'S in the in the SOLAR energy BUSINESS. SO uh you know WE'RE ACCUTELY AWARE of a lot of this. BUT you KNOW on the OTHER hand he VOTED for George BUSH. So UM you KNOW i i WONDER SOMETIMES if **HE** knows what he's DOING.

The presence of accent on a coreferential pronoun could be the result of many interacting constraints, semantic and prosodic, including those imposed by both the larger discourse context and the words immediately surrounding the pronoun. For example, the dialogue act, the speaker's and pronoun's tendencies, the overall prosody of the utterance, and the presence of other referring expressions such as *my dad* or even *I* or *you* must all interfere with the presence of accent on a coreferential pronoun.

[5] Audio clips: www.stanford.edu/~anubha/accentedPro.html.

While the models and examples presented here cannot shed light on the precise mechanisms and constraints, they do show that a simple attentional story suffers from being limited to the analysis of local 2-utterance-windows and (primarily) pronouns like *he* or *she*. Also, they demonstrate that the role of contrast may have been seriously underestimated by previous theoretical work. It is especially vital to understand whether contrast might in fact subsume the attentional explanation for pronominal accent because switching to a less salient referent for an accented pronoun might very well be viewed as contrast between the expected situation (where the topic is expected to be continued) and the unexpected situation in which a lower ranked referent takes front stage. So in example (1), where the accented *HE* is taken to refer to the lower-ranked object *Bill* from the previous sentence, the accent could be signaling contrast between the expected referent *John* and the unexpected but true referent *Bill*.

Topic-continuity could be just one type of (linguistic) expectation language-users are sensitive to, such that they might choose to signal a violation of that expectation through accentuation. Other expectation-violations do lead to similar accentuation patterns, as when a situation is judged to be unexpected by common knowledge or context; this is evident in an utterance like "*SHE married HIM?*", expressing surprise at an unlikely couple. A contrast-based explanation is bolstered by the eye-tracking experiments of Venditti et al. (Venditti et al., 2002) which show that both potential antecedents are evoked upon hearing an accented pronoun rather than just the antecedent predicted by a saliency ranking; in fact, the referent is not fixed until more propositional information is encountered and the discourse coherence relation determined. Moreover, if not for contrast, there is no explanation of why accent appears appropriate on the pronouns in the following discourse, even though their referents are not ambiguous at all:

(6) John called Mary a Republican. Then SHE insulted HIM.

Future work would need to look at genres of speech other than spontaneous conversational dialogue. Even more data from more speakers would be beneficial, in order to cluster them into meaningful speaker types or styles.

6 Conclusions

The analysis presented here makes use of a large quantity of spontaneous speech, with more features and more sophisticated statistical models than have been available or employed to date. The ensuing results lead to the following conclusions:

- Pronominal accent is not a robust cue of an "unusual" antecedent, at least not when "unusual" is defined in terms of the attentional salience of the pronoun's referent.
- Pronominal accent does serve as a cue to contrast beyond the effects of antecedent properties and speakers' accentuation preferences, though the exact constraints and interactions need to be understood. A contrast-based explanation may subsume the salience-based examples.
- Understanding speaker and pronoun dependencies is highly important. It may be quite fruitful to discover speaker types or styles. Also, inter-pronoun variation in accentuation is also a significant predictor of how likely a given pronoun is to bear accent, although it is possible that variation between pronouns is an indirect reflection of yet-to-be-discovered constraints.

Acknowledgements This research was partly funded by the Edinburgh-Stanford Link. Many thanks to Joan Bresnan and Ani Nenkova for helpful discussion, and to Jason Brenier and Sasha Calhoun for help in collecting the prosodic data.

References

Mira Ariel. 1990. *Accessing Noun-Phrase Antecedents*. Routledge, London, UK.

Gillian Brown. 1983. Prosodic Structure and the Given/New Distinction. D. R. Ladd and A. Cutler, eds., *Prosody: Models and Measurements*, 67–78. Springer Verlag, Berlin, Germany.

Janet Cahn. 1995. The Effect of Pitch Accenting on Pronoun Referent Resolution. *Proc. of the Association for Computational Linguistics*, 290–293.

Sasha Calhoun. 2006. *Information Structure and the Prosodic Structure of English: a Probabilistic Relationship*, Ph.D. thesis, University of Edinburgh.

Sasha Calhoun, Malvina Nissim, Mark Steedman, and Jason Brenier. 2005. A Framework for Annotating Information Structure in Discourse. *Proc. of the Association for Computational Linguistics Conference Workshop on Frontiers in Corpus Annotation II: Pie in the Sky*.

Talmy Givón. 1983. *Topic Continuity in Discourse.* John Benjamins, Philadelphia, PA.

John J. Godfrey, Edward C. Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone Speech Corpus for Research and Development. *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, 1:517–520.

Barbara Grosz, Aravind Joshi, and Scott Weinstein. 1995. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21(2):203–225.

Jeanette K. Gundel, Nancy Hedberg, and Ron Zacharski. 1993. Cognitive Status and the Form of Referring Expressions in Discourse. *Language*, 69(2):274–307.

Helen de Hoop. 2003. On the Interpretation of Stressed Pronouns. *Proc. of the Conference "sub7 - Sinn und Bedeutung"*, 159–172.

Ross Ihaka and Robert Gentleman. 1996. R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5(3):299–314.

Megumi Kameyama. 1999. Stressed and Unstressed Pronouns: Complementary Preferences. Peter Bosch and Rob van der Sandt, eds., *Focus: Linguistic, Cognitive, and Computational Perspectives*, 306–321. Cambridge University Press, Cambridge, UK.

Christine Nakatani. 1997. *The Computational Processing of Intonational Prominence: A Functional Prosody Perspective*, Ph.D. thesis, Harvard University.

Malvina Nissim, Shipra Dingare, Jean Carletta, and Mark Steedman. 2004. An Annotation Scheme for Information Status in Dialogue. *Proc. of the 4th Language Resources and Evaluation Conference*.

Mari Ostendorf, Izhak Shafran, Stefanie Shattuck-Hufnagel, Leslie Carmichael, and William Byrne. 2001. A Prosodically Labeled Database of Spontaneous Speech. *Proc. of the ISCA Workshop on Prosody in Speech Recognition and Understanding*, 119–121.

Jennifer Venditti, Matthew Stone, Preetham Nanda, and Paul Tepper. 2002. Discourse Constraints on the Interpretation of Nuclear-accented Pronouns. *Proc. of the 2002 International Conference on Speech Prosody*.

Duane Watson, Jennifer E. Arnold, and Michael K. Tanenhaus. 2006. Acoustic Prominence and Reference Accessibility in Language Production. *Proc. of the 2006 International Conference on Speech Prosody*.

Maria Wolters and David Beaver. 2001. What does *he* mean? *Proc. of the Annual Meeting of the Cognitive Science Society*.

Maria Wolters and Donna K. Byron. 2000. Prosody and the Resolution of Pronominal Anaphora. *Proc. of COLING2000*.

Evaluating combinations of dialogue acts for generation

Simon Keizer & Harry Bunt
Department of Communication and Information Sciences
Faculty of Humanities
Tilburg University, The Netherlands
{s.keizer,h.bunt}@uvt.nl

Abstract

We will discuss an approach to dialogue act generation that reflects the multidimensionality of communication. Dialogue acts from different dimensions in the taxonomy used are generated in parallel, resulting in a buffer of candidates. The main focus of the paper will be on an additional process of evaluating these candidates, resulting in definitive combinations of dialogue acts to be generated. This evaluation process is required to deal with the interdependencies there still are between dimensions, and involves logical, strategic and pragmatic considerations.

1 Introduction

In natural language dialogue, participants have to take into account several aspects of the communicative process in interpreting and generating utterances. Besides asking questions, giving instructions, and putting requests, related to some underlying task, the dialogue partners should also keep track of the status of processing each other's utterances, deal with interaction management issues such as turn-taking and topic management, and with social aspects of communication like greeting and apologising.

These different aspects of communication are reflected in the dialogue act taxonomy[1] as developed within Dynamic Interpretation Theory (DIT) (Bunt, 2000). This taxonomy consists of currently 10 *dimensions*, each containing communicative functions addressing one of the aspects. The taxonomy allows for multifunctional utterances, in the sense that every utterance in a dialogue gets assigned at most one function from each dimension.

This multidimensionality suggests that in generating dialogue behaviour, participants select dialogue acts from different dimensions simultaneously and independently, and then combine them into multifunctional utterances. However, this combination process is not straightforward. There are dependencies between dialogue acts from different dimensions that have to be taken into account. For example, dialogue acts may be in conflict with each other, so only one of them can be generated, or a dialogue act may already be implied by another and a decision has to be made whether or not to explicitly generate it. Moreover, not every combination of dialogue acts can be realised in a multifunctional natural language utterance, but only in subsequent utterances.

In this paper, we will discuss an approach to dialogue act generation that both acknowledges the multidimensionality of communication and deals with the problem of interdependencies between dimensions. In this approach, we distinguish a separate process of evaluating candidate dialogue acts that have been generated on the basis of dimensions in isolation. In general, the evaluation involves 1) resolving logical conflicts between dialogue acts, 2) strategic and pragmatic considerations for prioritising dialogue acts, and 3) language generation and non-verbal aspects of realising dialogue acts. Here, we will particularly focus on the first and second phases of the evaluation.

We have developed a dialogue manager using

[1] http://let.uvt.nl/general/people/bunt/docs/dit-schema2.html

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 158–165,
Antwerp, September 2007.

the abovementioned multidimensional taxonomy in generating dialogue behaviour (Keizer and Bunt, 2006). Dialogue acts from different dimensions are generated in parallel through several *Dialogue Act Agents* operating on the system's information state. Each agent is associated with one specific dimension, and generates contributions related to that dimension only. An additional *Evaluation Agent* takes care of constructing combinations of dialogue acts for actual generation in system utterances. This multi-agent design allows to experiment with different dialogue strategies and styles of communication, having their specification concentrated in the Evaluation agent. A similar argument is used in (Stent, 2002), discussing a dialogue manager consisting of three independent agents operating in parallel. The 'organisation of conversation acts into coherent and natural dialogue contributions' is taken care of by one of these agents, called the Generation Manager. The distinction between the processes of 'contribution planning' and 'contribution structuring' has some similarity with our distinction between the dialogue act agents (over-)generating dialogue acts and the Evaluation agent selecting and combining the resulting candidates. However, contribution structuring deals with interrelationships between the *levels* of conversation acts, whereas our Evaluation agent operates on the basis of interdependencies between *dimensions* of dialogue acts.

Another multi-agent approach to dialogue management is taken in JASPIS (Turunen et al., 2005), a speech application architecture for adaptive and flexible human-computer interaction. The system uses so-called 'Evaluators' that determine which agents should be selected for different interaction tasks, based on evaluation scores. Part of an Evaluator's task may be to decide on a particular dialogue strategy by selecting a corresponding dialogue agent. Our approach of evaluation also involves issues of dialogue strategy, but this is not carried out by selecting between contributions from alternative agents for the same task.

2 Dynamic Interpretation Theory

In Dynamic Interpretation Theory (DIT) (Bunt, 2000), utterances in a dialogue are modelled in terms of combinations of dialogue acts that operate on the information state of the dialogue participants. A dialogue act has a *semantic content*, expressing what the act is about, and a *communicative function* specifying how the semantic content is to be taken to change the information state of the addressee. Communicative functions are organised in a 10-dimensional taxonomy, in which the dimensions reflect different aspects of communication that speakers may address simultaneously in their dialogue behaviour. In each utterance, several dialogue acts can be performed, each dialogue act from a different dimension. The overview below shows a layered structure in which the dimensions are given in boldface italic. So, besides the *Task* dimension, the taxonomy provides for several *Dialogue Control* dimensions, organised into the layers of *Feedback*, *Interaction Management (IM)* and *Social Obligations Management (SOM)*.

- ***Task/domain***: acts that concern the specific underlying task and/or domain;
- **Dialogue Control**
  - **Feedback**
    - ***Auto-Feedback***: acts dealing with the speaker's processing of the addressee's utterances; contains positive and negative feedback acts on different levels of understanding (see below);
    - ***Allo-Feedback***: acts dealing with the addressee's processing of the speaker's previous utterances (as viewed by the speaker); contains positive and negative feedback-giving acts and feedback elicitation acts on different levels of understanding (see below);
  - **Interaction management**
    - ***Turn Management***: turn accepting, giving, grabbing, keeping;
    - ***Time Management***: stalling, pausing;
    - ***Partner Processing Management***: completion, correct-misspeaking;
    - ***Own Processing Management***: error signalling, retraction, self-correction;
    - ***Contact Management***: contact check, indication;
    - ***Topic Management***: topic introduction, closing, shift, shift announcement;
  - ***Social Obligations Management***: initiative and response acts for salutation, self-introduction, gratitude, apology, and valediction.

A participant's information state in DIT is called his *context model*, and contains all information considered relevant for his interpretation and generation of dialogue acts. A context model is structured into several components:

1. *Linguistic Context*: linguistic information about the utterances produced so far (an

extended dialogue history); information about planned system dialogue acts (dialogue future);

2. *Semantic Context*: contains current information about the task/domain, including assumptions about the dialogue partner's information;
3. *Cognitive Context*: the current processing states of both participants, expressed in terms of a level of understanding reached (see below)
4. *Physical and Perceptual Context*: the perceptible aspects of the communication process and the task/domain;
5. *Social Context*: current communicative pressures.

In keeping track of the participants' processing states in the cognitive context, four levels of understanding are distinguished: 1) *perception*: the system was able to hear the utterance (successful speech recognition), 2) *interpretation*: the system understood what was meant by the utterance (successful dialogue act recognition), 3) *evaluation*: the information presented in the utterance did not conflict with the system's context (successful consistency checking), and 4) *execution*: the system could act upon, do something with, the utterance (for example, answering a question, adopting the information given, carrying out a request, etcetera).

These levels of understanding are also used in distinguishing different types of auto- and allo-feedback dialogue acts, each for signalling processing problems on a specific level.

3 Dialogue act generation

The architecture of the dialogue manager is given in Figure 1. Central is the context model, currently containing four of the five components defined in DIT, the Physical & Perceptual Context currently considered to be irrelevant four our purposes. The Context Manager takes care of updating the context model during the dialogue with every new utterance being produced, be it a user or a system utterance. Both for interpretation of user utterances and generation of system utterances, the dialogue manager makes use of the multidimensional taxonomy. User utterances are analysed and eventually interpreted in terms of sets of dialogue acts by the Dialogue Act Recogniser (DAR), the results of which are then written in the dialogue history. The Context Manager then takes care of updating the entire context model and checking it for inconsistencies.

For dialogue act generation, separate *Dialogue Act Agents* are used, that each take care of generating acts from a particular dimension of the taxonomy. These generated acts are recorded as candidates in the dialogue future of the Linguistic Context. Currently, five dialogue act agents have been implemented, covering the five most relevant dimensions for our purposes.

The *Task Agent* is associated with the task/domain dimension: it is responsible for the underlying task itself. In the case of question answering (QA), it basically generates answers to domain questions, where it can turn to either a structured database with domain information, or to a QA module taking self-contained domain questions, to retrieve the information to be contained in the answers it generates. The Task Agent operates primarily on the information in the Semantic Context.

The *Auto-FB Agent* monitors the own processing state as stored in the Cognitive Context, making sure that the system correctly understood the user's utterances. The agent generates negative auto-feedback acts in case of processing problems and occasional positive feedback in case of successful processing.

Similarly, the *Allo-FB Agent* monitors the partner's processing state, also in the Cognitive Context. It generates positive and negative feedback concerning the extent to which the user understood the system correctly.

The *SOM Agent* takes care of the social aspects of the communication. It generates reactive SOM acts to release reactive pressures in the social context (created by initiative SOM acts by the user). It can occasionally generate initiative SOM acts such as apologies (for example, after repeated processing problems).

Finally, the *TimeM Agent* generates pausing acts in case the system wants to gain time in order to perform some task, like retrieving information for answering a domain question.

Although the dimensions of the taxonomy are supposed to be orthogonal, i.e., dialogue acts in a dimension are selected independently, there are still

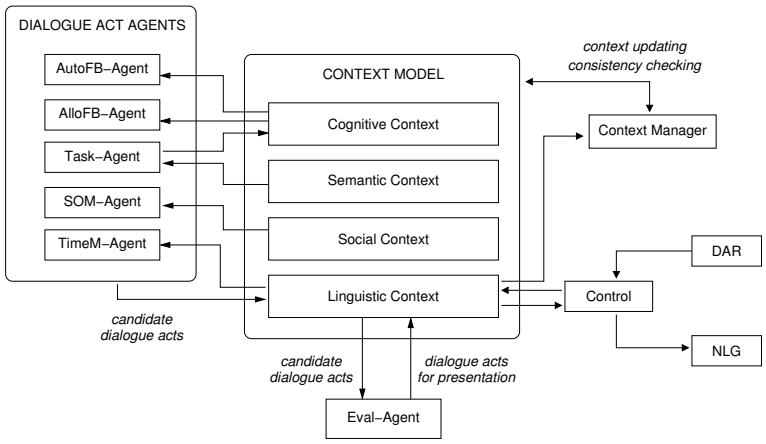


Figure 1: Architecture of the dialogue manager.

dependencies between the dialogue acts that have to be taken into account when combining them for actual generation in multifunctional system utterances. Therefore, an addition *Evaluation agent* is introduced that takes care of evaluating these candidate dialogue acts and decide on definitive combinations of dialogue acts for generation.

3.1 Design of the Evaluation Agent

The procedure for the Evaluation Agent to schedule combinations of dialogue acts from the list of candidates is subdivided into three phases. In the first phase, the dialogue act candidates are evaluated for any inherent dependencies among them. Dialogue acts from different dimensions may be in conflict with each other, so only one of them can be selected for generation, and the other has to be cancelled. The choice of which one to select and which one to cancel is based on some priority ordering among dialogue acts.

The occurrence of dialogue acts having a logical conflict implies that there is some inconsistency in the context model. Although this is undesirable and should be avoided in the design of the dialogue manager, it is nevertheless preferable to design the Evaluation Agent in such a way that it can deal with *any* combination of dialogue acts, irrespective of how the candidates were generated on the basis of the context model.

Moreover, the context model does allow for some type of inconsistency, and therefore, the generation of conflicting candidate dialogue acts. During the updating of the context model with new utterances, new information that has not been successfully integrated in the context model yet, gets stored in what is called the *pending context*, and therefore might be conflicting with the definitive context. Once the context manager has detected such an inconsistency, it records an evaluation level processing problem in the cognitive context, which could trigger the generation of a corresponding auto-feedback act. Inconsistencies might also occur within the pending context itself.

In the second phase, the remaining list of non-conflicting candidates is evaluated from a pragmatic and dialogue strategic point of view. In some circumstances, depending on the nature of the underlying task and the communicative setting, it makes more sense to postpone certain dialogue acts and give priority to others.

One type of strategic consideration is related to the planning of task acts and does not involve the combination of dialogue acts from different dimensions. For example, if the system has several questions to ask to the user, it has to decide whether to combine these questions in a single turn, or to ask them one at a time, in a particular order, distributed over several turns. The latter strategy is the more conservative one, and is used in situations where the risk of misunderstandings is higher, like in noisy environments or where the quality of speech recognition is limited.

Another strategic issue involves the choice whether or not to explicitly produce a dialogue act

that is already implied by the other candidates. For example, a positive auto-feedback act does not need to be generated explicitly, if an answer gets generated that already implies this feedback. However, in some circumstances, there might be good reasons for explicitly performing the implied dialogue act anyway.

A third evaluation issue related to strategic considerations is that of dialogue acts being triggered by reactive pressures, e.g., a thanking down-player ("you're welcome"). Such dialogue acts have to be either generated 'immediately', or not at all: they cannot be postponed. If the system is to behave very socially and there is less need for efficiency, it should generate these response social acts more frequently. This is also true for the generation of initiative social acts, like apologies and thanking.

Finally, in the third phase, combinations of dialogue acts are selected that can actually be realised in multifunctional system utterances. Some combinations of dialogue acts may not carry any logical conflicts, but the particular natural language may not provide a multifunctional utterance for the dialogue acts to be realised. For example, a question in one dimension cannot be combined with an inform in another dimension using one single utterance, because the question requires an interrogative and the inform a declarative sentence.

Besides the construction of multifunctional utterances, some of the dialogue acts can also be realised in a non-verbal manner, for example by means of animations on the graphical user interface of the system, or by means of gestures made by the system if it is an embodied virtual agent.

4 Logical conflicts

Suppose the list of candidates contains both a dialogue act with an answer function (WH-Answer, YN-Answer, etc.) and a negative auto-feedback act on the level of perception or interpretation. Clearly, it would be absurd to generate both dialogue acts, as the answer also implies (overall) positive feedback. One exception would be that the negative feedback act would be about a different utterance in the dialogue history than the question the answer referred to. In that case, either the feedback act or the answer has to be specific about which utterance in the history it responds to. The dialogue act combination should be realised in an utterance following the pattern "<wh-answer>, but <neg-feedback>".

Combining a negative auto-feedback act on execution level and an answer dialogue act also leads to a logical conflict, since an answer implies successful execution(-level processing) of the corresponding question. Giving an answer and at the same time signalling it could not find an answer is inconsistent. Note that it might be the case that the Task Agent found one or more answers to a question, but decided they were not reliable enough to present them to the user. Such answers might be stored somewhere by the Task agent, and possibly generated after all later in the dialogue, but initially they are not candidate dialogue acts.

As in the case of answers, all dialogue acts that have an aspect of referring back to some previous utterance (or, in terms of the DAMSL dialogue act annotation scheme (Allen and Core, 1997), that are 'backward-looking'), imply overall positive feedback and hence conflict with negative auto-feedback in the ways indicated above. In the DIT taxonomy, this is the case for any type of allo-feedback, for reactive SOM acts (react-greet, apology-downplay, etc.) and for dialogue acts with a communicative function such as Agreement, Correction, and Address Request.

5 Strategic issues

Given a list of dialogue act candidates that have no logical conflicts among them, it is still not just a matter of simply mapping them onto a multifunctional utterance. Depending on the situation, it might be strategically or pragmatically preferable to give priority to some dialogue acts and postpone or even cancel others.

Whereas the relative priorities of dialogue acts for dealing with conflicts are strict in the sense that they should ensure rational behaviour (which means not producing conflicting dialogue acts, nor giving priority to a dialogue act implying positive feedback over a negative auto-feedback act), those of non-conflicting dialogue acts can be adjusted for implementing different dialogue strategies and styles of communication.

5.1 Negative auto-feedback

If the system encounters processing problems during the dialogue, it should try to solve these problems, before attending to any other aspects. So in general, negative auto-feedback acts should be given priority over all other dialogue acts.

As we have seen in the previous section, combinations of answers and negative auto-feedback on the level of either perception or interpretation give a logical conflict. However, combinations of answers and negative feedback on the level of evaluation do not. The Task Agent can be triggered by a new user goal, even if this is part of the pending context only. This is the case if the dialogue act recogniser was able to detect a question about the domain in the user utterance, but the context manager did not check this new information for consistency with the context model yet, or already detected an inconsistency (i.e., an evaluation problem was encountered). Now, the candidates list could contain both an answer to the question and a negative auto-feedback act on the level of evaluation.

The example dialogue fragment below illustrates such a situation in which only the feedback act is selected for the eventual system utterance. The dialogue acts indicated reflect the system's interpretation of the user's utterances. The system encountered a conflict in his context model, because it believes that the user can see the 'send button' (after U0), and therefore knows where it is, but it should also believe that the user wants to know where the send button is (after U2). This conflict makes the Auto-feedback agent generate a negative auto-feedback act on evaluation level, whereas the recognised user goal in U2 triggers the Task agent to construct an answer. In the example, the Evaluation agent selects only the feedback act for generation.

- U0: *I see the send button.* INFORM(see_sbutton)
- S1: *okay.* POS-AUTO-FEEDBACK-EXE
- U2: *where is the send button?* WHQ(loc,sbutton)
- S3: *but you just told me you saw the send button!* NEG-AUTO-FEEDBACK-EVAL

The answer to U2 is kept in the candidates list until it is clear whether the system had misinterpreted U0 or U2. In the following dialogue continuation, the user in U4 corrects the system in his interpretation of U0, and hence, the answer can be generated:

- U0: *I see the send button.* INFORM(see_sbutton) [user intended INF(need_sbutton)]
- S1: *okay.* POS-AUTO-FEEDBACK-EXE
- U2: *where is the send button?* WHQ(loc,sbutton) [user intended WHQ(loc,sbutton)]
- S3: *but you just told me that you saw the send button!* NEG-AUTO-FEEDBACK-EVAL
- U4: *no, I told you that I* needed *it.* NEG-ALLO-FB-INT; INF
- S5: *oh, hold on ... the send button is on the bottom right.* POS-AUTO-FB-EXE; PAUSE; WHA(loc,sbutton,bottomr)

Alternatively, the system misinterpreted U2, in which case the answer can be cancelled. In the dialogue continuation below, the user in U4 corrects the system in his interpretation of U2, and hence, the answer has to be replaced:

- U0: *I see the send button.* INFORM(see_button) [user intended INF(see_sbutton)]
- S1: *okay.* POS-AUTO-FEEDBACK-EXE
- U2: *where is the send button?* WHQ(loc,button) [user intended WHQ(loc,pbutton)]
- S3: *but you just told me that you saw the send button!* NEG-AUTO-FEEDBACK-EVAL
- U4: *no, I wanted to know where the* print *button is.* NEG-ALLO-FB-INT; IND-WHQ(loc,pbutton)
- S5: *oh, hold on ... the print button is on the bottom left.* POS-AUTO-FB-EXE; PAUSE; WHA(loc,pbutton,bottoml)

In the above examples, only the negative feedback act was selected for generating S3 and the answer was cancelled. However, the system could also follow an alternative strategy of generating both the negative feedback evaluation and the answer, which would result in something like "the 'send button' is on the bottom right, but didn't you just tell me you saw it?".

5.2 Negative allo-feedback

If the system, after processing a user utterance, has reason to believe that the user did not correctly understand the system's previous utterance, then this last user utterance may not be so relevant anymore. Any candidate dialogue acts triggered by changes in the context model due to this user utterance will not be so relevant either. Therefore, dealing with the user's processing problems should get priority

over any other aspects. In general, negative allo-feedback acts should be given priority over other dialogue acts, except for negative auto-feedback acts.

In the example dialogue fragment below, user and system are discussing a music concert by the Borodin Quartet. The system asks a question and the user responds with a return question which, to the system, seems unrelated. After processing U1, the system could conclude that he misinterpreted the user, because it expects some answer in the form of numerical information only. In that case, no answer would be generated as a candidate dialogue act. Only a negative auto-feedback act on interpretation level, possibly in combination with a request in the task dimension would be generated, resulting in system utterance (S2).

- S0: *how many tickets do you want?* WHQ
- U1: *how much is the Kronos Quartet concert?* WHQ
- S2: *Sorry, I do not understand what you mean.* APO;NEG-AUTO-FB-INT
  Please indicate the number of tickets you want REQ

  S2a: *No, I would like to know the number of tickets you want* NEG-ALLO-FB-INT

  S2b: *The Kronos Quartet concert is 30 euro,* POS-AUTO-FB-INT; WHA
  but I asked about the Borodin Quartet. allo-fb:INF

  S2c: *The Kronos Quartet concert is 30 euro.* WHA

Another scenario would be that the system successfully interpreted U1 as a domain question, but concludes that the user must have misinterpreted S0. This causes the generation of two candidate dialogue acts: a negative allo-feedback act on interpretation level, and an answer to U1. The particular strategy of the system will determine whether only the feedback act will be generated (S2a), both the feedback act and the answer (S2b), or even only the answer (S2c).

5.3 Scheduling task acts

After dealing with any processing problems, the underlying task should be the most important thing to attend to, so dialogue acts in the task dimension should get the highest priority, after negative auto- and allo-feedback acts.

On the basis of the user's input, the generation of several task-oriented dialogue acts can be triggered at once. Some user question or request could trigger several questions the system needs the user to answer before he can answer the question or carry out the request. In the case of several task-oriented dialogue acts, the relative priorities of these candidates are based on task-specific considerations. This could be based on some preferred, logical order in which subtasks should be carried out; in route-planning for example, it might be preferable to ask for the destination location before asking for the date on which the user wants to travel.

5.4 Positive auto-feedback

Every time the system reaches some level of successful processing a user utterance, a positive auto-feedback act signalling this to the user can be triggered. However, actually generating this dialogue act in all of these cases leads to a kind of communicative behaviour that can be experienced by the user as rather annoying. Instead, positive feedback should be generated only occasionally, with a frequency depending on the specific communicative setting.

In the case of dialogues involving the transfer of important information such as credit card numbers, it is desirable to give more positive feedback, but in the case of more informal dialogues, too much positive feedback should be avoided. Although the extent to which positive auto-feedback is given can be taken care of by the Auto-feedback Agent in generating candidates, it is also a matter of evaluating such acts against the other dialogue act candidates. In particular, positive feedback is often already implied by other acts, and therefore does not necessarily have to be generated explicitly.

Also in the case of train table information, like in the dialogue fragment below, giving positive feedback can be a good strategy. After U2, the system has gathered enough information from the user in order to answer his original (Indirect WH-)Question U0. In S3, the system generates this answer, thereby implying positive feedback about U2. However, successful processing of U2 also results in an auto-feedback candidate act that might be generated explicitly as well, as is the case in S3' or S3". In these cases, generating the feedback act reflects a strategy of implicit verification.

- U0: *I'd like to know when the next train to Amsterdam is*

leaving. IWHQ

- S1: *From where are you travelling?* WHQ
- U2: *From Tilburg.* WHA
- S3: *The next train leaves at 10:30h from platform 1.* WHA

S3': *So you want to go from Tilburg to Amsterdam. The next train leaves at 10:30h from platform 1.* POS-AUTO-FB-EXE; WHA

S3": *The next train from Tilburg to Amsterdam leaves at 10:30h from platform 1.* POS-AUTO-FB-EXE WHA

Another typical example of generating positive auto-feedback in combination with other acts is in the dialogue fragment below. The system asks the user a question (S0), but he is not happy about the answer given by the user (U1):

- S0: *When do you want to go?* WHQ
- U1: *I want to go to Amsterdam.* INF
- S2: *Okay, but* when*?* POS-AUTO-FB-EXE NEG-ALLO-FB-INT; WHQ

In S2, the system gives negative allo-feedback about the user's interpretation of S0 and positive auto-feedback about U1. Only after successful interpretation of the previous utterance (U1) as an answer to his question, the system may conclude that the user did not correctly understand the original question (S0).

5.5 Styles of communicative behaviour

The extent to which positive feedback acts are generated, is also a matter of communication style, besides the strategic considerations behind it. Communication style is also reflected through the generation of both initiative and reactive SOM acts. In more formal, task-oriented dialogues, the generation of apologies for example should be kept to a minimum, whereas in more informal dialogues, apologies can make the system's behaviour more natural and therefore, pleasant to the user.

Again, the dialogue act agent responsible for the generation of these acts could take care of the frequency in which these are actually generated, but this also depends on the other available candidates, making it an issue for the Evaluation Agent as well. For example, apologies can be used in combination with negative feedback acts, but their impact in utterances like "sorry?" is not as high as in utterances like "I'm sorry, I did not hear what you were saying".

6 Conclusion and future work

We have discussed an approach to dialogue act generation reflecting the multidimensionality of communication. Particular focus was on the problem of dealing with interdependencies between dialogue acts from different dimensions that have been constructed independently. Giving priority to some dialogue acts and postponing or cancelling others involved logical, strategic and pragmatic considerations, besides specific language generation issues that we did not discuss. A separate process of evaluating candidate dialogue acts allows for implementing different dialogue strategies and communication styles in the dialogue manager.

An interesting topic for future research would be to look at the possibility to assess the (relative) priorities among candidate dialogue acts from data. An advantage of this would be that one could easily adjust the dialogue manager for different types of dialogue (both in terms of the underlying task and style of communication) by reassessing the priorities with appropriate data.

Acknowledgement

This research is carried out in the IMIX-PARADIME project, funded by the Dutch national research foundation NWO.

References

J. Allen and M. Core. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers. Dagstuhl Workshop. `http://www.cs.rochester.edu/research/cisd/resources/damsl/`.

H. Bunt. 2000. Dialogue pragmatics and context specification. In H. Bunt and W. Black, editors, *Abduction, Belief and Context in Dialogue*, Studies in Computational Pragmatics, pages 81–150. John Benjamins.

S. Keizer and H. Bunt. 2006. Multidimensional dialogue management. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, pages 37–45.

A.J. Stent. 2002. A conversation acts model for generating spoken dialogue contributions. *Computer Speech and Language, Special Issue on Spoken Language Generation*, 16(3–4):313–352.

M. Turunen, J. Hakulinen, K.-J. Räihä, E.-P. Salonen, A. Kainulainen, and P. Prusi. 2005. An architecture and applications for speech-based accessibility systems. *IBM Systems Journal*, 44(3):485–504.

Measuring Adaptation Between Dialogs

Svetlana Stenchikova and **Amanda Stent**
Computer Science Department
Stony Brook University
Stony Brook, NY 11794-4400
sveta, stent@cs.sunysb.edu

Abstract

The paper proposes two new approaches for measuring adaptation between dialogs. These approaches permit measurement of adaptation both to conversational partner (*partner adaptation*) and to the local dialog context (*recency adaptation*), and can be used with different types of feature. We used these measures to study adaptation in the Maptask corpus of spoken dialogs. We show that for syntactic features, recency adaptation is stronger than partner adaptation; however, we find no significant differences for lexical adaptation using these measures.

1 Introduction

Numerous psycholinguistic studies have demonstrated that people adapt their language use in conversation to that of their conversational partners. For example, conversational partners adapt to each other's choice of words, particularly referring expressions (Brennan and Clark, 1996), converge on certain syntactic choices (Pickering et al., 2000; C. Lockridge, 2002), adapt their prosody to help their partners disambiguate syntactic ambiguities (Kraljic and Brennan, 2005), and also adapt using audiovisual information (Kraut et al., 2003).

Some of these results have been duplicated using corpus studies; for example, researchers have found evidence of within-speaker and between-speaker convergence to certain syntactic constructions (Dubey et al., 2006; Reitter et al., 2006). Corpus studies can be a good addition to more tightly controlled empirical studies in cases where there is a corpus already available. Corpus studies can confirm the results of psycholinguistic research, and can identify issues that may 'muddy' empirical results.

Finally, there is some evidence that people adapt their language use in conversation with computer partners. For example, researchers have shown that users of dialog systems adapt the system's choice of referring expressions (Brennan, 1996), the system's choice of modality for referring (Bell et al., 2000; Skantze, 2002), or the system's choice of words (Gustafson et al., 1997).

Currently, there is a debate in the psycholinguistics community about whether this adaptation is:

- *partner adaptation* – adaptation based on a model of the partner. This type of adaptation is sometimes called entrainment or audience design (Brennan and Clark, 1996; Horton and Gerrig, 2002).
- *recency adaptation* – adaptation due to the representations of words, concepts etc. being *activated*, or brought to the forefront during language production, by previous perception or comprehension. This type of adaptation is sometimes called convergence, priming or alignment (Brown and Dell, 1987; Pickering and Garrod, 2004; Chartrand and Bargh, 1999).

In this paper, we consider measures used in corpus-based studies of adaptation such as (Dubey et al., 2006; Reitter et al., 2006; Church, 2000). These measures do not permit examination of whether adaptation is due to the partner or to recency, and do

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 166–173,
Antwerp, September 2007.

not measure the strength of adaptation. We propose two new measures, one that measures the presence of adaptation and another that measures its strength. Together, these measures can identify adaptation within a single document or between documents; can identify the strength of adaptation as well as its presence; and can be used to identify the source of the adaptation. We use these measures to study adaptation in the Maptask spoken dialog corpus. We show that for syntactic features, recency adaptation is stronger than partner adaptation; however, we find no significant differences for lexical adaptation using these measures. We close with some ideas about how to apply these measures to dialog system development, and some ideas for future work.

2 Other Measures

Church (Church, 2000) introduced a method for measuring lexical "adaptation" in text. This method determines whether appearance of a lexical feature in the 'priming portion' of a document affects the likelihood of its appearance in the 'target' (later) portion. This method requires the construction of a contingency table for each feature in a corpus of texts, showing how many of the texts contained the feature: (a) in the 'priming portion' only, (b) in the 'target' only, (c) in both portions, and (d) in neither portion. The probability of positive adaptation is computed as $c/(a+c)$. This must be compared with a prior probability, which is $(a+c)/(a+b+c+d)$. Church applied this method to the study of a corpus of text documents, treating the first half of each document as the 'priming portion' and the second half as the 'target'. He showed that positive lexical adaptation does occur, more strongly for content words than for function words.

Dubey et al. used Church's method to evaluate adaptation for selected syntactic constructions in the Brown and Switchboard corpora (Dubey et al., 2006). They reported positive adaptation for each of the syntactic constructions they considered.

Church's measure was developed to identify the most useful features for information retrieval, rather than for study of adaptation *per se*. Consequently, it has several disadvantages for studying adaptation directly:

- For each feature, this method provides an answer to the question "Did the feature occur in the prime/target?"; however, it does not take into account the frequency of occurrence of a feature, so cannot be used to measure the strength of adaptation
- This method cannot be used to identify adaptation in a single document or between a pair of documents
- This method under-reports adaptation in frequently occurring features

In recent work, Reitter *et al.* (Reitter et al., 2006) investigated syntactic adaptation in Switchboard and Maptask. Instead of using Church's method, they used logistic regression to examine short-term priming effects within a small window of time in single dialogs. This method permits study of the time course of adaptation, but because it applies within a single document only it does not permit examination of the source of adaptation (recency/partner model).

3 Our Measures

We propose two measures. The first one measures the prevalence of adaptation between two documents, while the second one measures the strength of adaptation.

Throughout this discussion, we will use the term 'document' to refer to a dialog or part of a dialog, and the term 'feature' to refer to any phenomenon (lexical, syntactic, referring expression, dialog act, etc.) that occurs in or is labeled in documents.

To measure the degree to which a feature f exhibits adaptation, we divide the corpus into a collection of 'prime' documents and 'target' documents. For each feature f, we compute the frequency of occurrence of the feature in the 'prime' document (*p*), the 'target' document (*t*), and the corpus as a whole (baseline, or *b*). One may use relative frequencies rather than absolute frequencies, or smooth low-frequency features; we do not do this in the experiments reported in this paper because earlier experiments showed that these did not change our results. Both of our measures compare *p* and *t* to *b*. We use the notation $f \in D$ as a shortcut to indicate that the frequency of occurrence of f in document D is greater than the baseline frequency for f.

3.1 Measure 1: Adaptation Ratio

This measure is a modification of Church's measure in two ways. First, it uses the frequency of occurrence of each feature in each document rather than merely its presence or absence. Second, instead of using Church's prior we use an estimate of the probability of feature co-occurence in prime and target by *chance*.

Chance The probability of a feature co-occuring in prime and target by chance is the product of probabilities of its occurence in prime and target independently, assuming independence of the two.

$$P(f \in prime \cap f \in target) =$$

$$P(f \in prime) * P(f \in target) \quad (1)$$

For N $(prime, target)$ dialog pairs where feature f occurs more that *b* times in P $primes$ and more than *b* times in T $targets$, the probability of chance co-occurrence of f in $prime$ and $target$ can be approximated by:

$$chance = (P/N) * (T/N) \quad (2)$$

+Adapt Church defines positive adaptation for a feature f as follows:

$$+adapt = Pr(f \in target \mid f \in Prime) \quad (3)$$

which we approximate as:

$$+adapt = T \cap P/P \quad (4)$$

For this method, we compute for each feature both $chance$ and $+adapt$. We define the *adaptation ratio* as $+adapt/chance$. We sort the features in decreasing order by adaptation ratio. Those at the top of the list exhibit more positive adaptation. We also compute χ^2 to identify features for which the adaptation ratio is significant.

3.2 Measure 2: Adaptation Strength

For this measure, instead of using binary values for each feature indicating presence or absence of that feature in a document, we use the actual frequency of occurrence of the feature in the document.

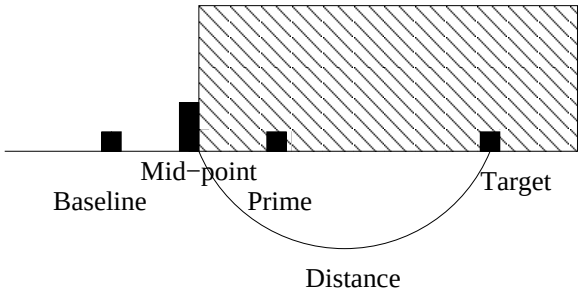


Figure 1: Graphical depiction of Distance

To measure the strength of adaptation on a per-feature basis, we use a $distance$ measure. For a feature f with frequency in $prime$ of p, frequency in $target$ of t and baseline frequency b,

$$distance = t - (p - b)/2 \quad (5)$$

Distance is computed for each feature for each dialog pair. Its value suggests the strength of adaptation for this feature in this dialog pair. Imagine adaptation as a force pulling t towards p and away from b. If there is positive adaptation, then t will be closer to p than to b, as illustrated in Figure 1 (we conservatively chose the midpoint between b and p; a point closer to b could be chosen for a more liberal interpretation of adaptation). We consider a feature to be *adapted* in a pair of dialogs if the target point lies to the right of mid-point in figure 1. We define the *adaptation strength* for a dialog as the average $distance$ over all *adapted* features.

4 Data

The Maptask corpus (Anderson and et. al., 1991) contains 32 sequences of dialogs involving four speakers who discuss routes displayed on maps and trade dialog partners as shown in Table 1. In each dialog, one partner is a *giver* of the route description and the other is a *receiver*. From each dialog sequence, we extract the dialog triples (1,4,6) and (2,3,5). The *follower,* **A**, in the first dialog in each triple (1 or 2) is the *giver* in the second and third dialogs; in the second dialog, **A** speaks with a new conversational partner and in the third dialog **A** speaks with the giver from the first dialog. We hypothesize that persistent recency adaptation will display between the first (*prime*) and second

(*recency*) dialogs in each triple (which are consecutive dialogs for **A**), and partner adaptation between the first (*prime*) and third (*partner*) dialogs in each triple.

Table 2 shows examples of two stem/POS features. *you/DET* occurs in 13 prime dialogs and 11 target dialogs. For 8 (prime, target) dialog pairs it occurs in both dialogs in the pair. For this feature, $chance$ is .14 and $+adapt$ is .62, so we say that this feature exhibits positive adaptation. For the feature *finish/VB* $+adapt$ is less than $chance$, so this feature does not exhibit adaptation.

dlg #	giver	follower	pair1	pair2
1	a1	b1	*prime*	
2	b2	a2		*prime*
3	a2	a1		*recency*
4	b1	b2	*recency*	
5	a2	b2		*partner*
6	b1	a1	*partner*	
7	a1	a2		
8	b2	b1		

Table 1: Maptask dialog order

5 Experiments

In these experiments we ask the following questions:

1. Can we identify the features that affect partner adaptation and recency adaptation?

2. Is partner adaptation or recency adaptation more prevalent?

3. Does the feature frequency in prime affect adaptation of the feature?

We consider two feature types: lexical (word stems, part-of-speech tagged to help distinguish between word senses; and bigrams); and syntactic (productions from the Maptask parse tree annotations).

5.1 Identifying features that exhibit adaptation

In this experiment we identify features with high *adaptation ratios*, looking at both partner and recency adaptation dialog pairs. To minimize noise from infrequently occurring features, in this experiment we only consider features occurring in more than 30% of prime dialogs with frequency higher than the baseline.

	partner	recency
ADJ	right-hand	bottom, right-hand
ADV	when, diagonal	right, well, about
AUX		have
CONJ	if	till, that, so
DET	you, across, on, what, that	my, i, just, that
INTJ	sorri, er,	uh
NOUN	bottom	map
PREP	across, through, along, from	from, by, to
VERB	know, got, take, pass	say

Table 3: Stem/POS features where *adaptation ratio*>1

partner	recency
your left, right-hand side, come to, you come, about the, when you, go round, and round, you got, if you, up toward, a wee, you just, round the, right you, just abov, abov the	no no, my map, okay and, you just, on my, down about, yeah i, you got, down to, have a, i mean, 'til you, just below, just to, now you, no you

Table 4: Bigrams of Stem/POS features where *adaptation ratio*>1

Tables 3, 4, and 5 show the stem/POS, bigram, and syntactic features with *adaptation ratio*>1 and significant χ^2. We observe two interesting categories of features that adapt: perspective and directionality.

In Maptask, speakers can take up a "map-based" perspective (and use words like *north, south, east, west*) or a "paper-based" perspective (and use words like *right, left, top, bottom*). Lexical features indicating perspective are adapted in both partner and recency dialog pairs; the same is true for bigram features. (Other features in this category (e.g. *left, top*) also show adaptation, but occur too infrequently for the adaptation to be significant.)

feature	prime	target	prime ∩target	+adapt	chance
you/DET	13	11	8	0.62	0.14
finish/VB	11	9	1	.09	.10

Table 2: Example lexical features

	partner	recency
advp->		advp
np->	at at ap nn	ap nn; np ap nn; at nn nn; np; np np; pn; ppg nn
pp->	in; rp	pp not pp; ql rp pp; rp aff
s ->	s aff aff s; hv np vp; np; np bez; s s	aff s; np; np s
vp->	vp be np; bez pp; to vp; vb np pp; vb vb pp; vbg pp	advp vp; ber vp; md vp; vb np; vbg; vbg pp vbn pp; vp vp

Table 5: Syntactic features where *adaptation ratio*>1

Directionality in Maptask is indicated by prepositions such as *across, through, along, around* and verbs such as *go* (vs. *take, send*). These prepositions are adapted in both partner and recency dialog pairs, for both lexical and bigram features; the verbs exhibit partner adaptation.

More syntactic features exhibit recency adaptation than partner adaptation.

Table 7 shows adaptation ratio and adaptation strength for some of the syntactic features that were examined in (Dubey et al., 2006). All but the first and last features show comparable partner and recency adaptation ratios. The adaptation strength for the feature $NP->NPPP$ shows stronger partner adaptation than recency adaptation. By contrast, the feature $NP->NN$ shows stronger recency adaptation.

5.2 Comparing partner and recency adaptation

In this experiment, we use *adaptation ratio* and *adaptation strength* to compare partner and recency adaptation. Table 8 shows *adaptation ratio* and *adaptation strength* averaged over all features for each feature type (Stem/POS, Stem/bigram, Syntactic). Positive adaptation for recency dialog pairs in this corpus appears significantly stronger for each feature type, however the probability of chance co-occurrence is also significantly stronger for recency. This explains why there is no significant difference in *adaptation ratio* for lexical features between partner and recency adaptation dialog pairs.

According to the *adaptation ratio* measure, lexical features do not exhibit significant differences between partner adaptation and recency adaptation. However, according to the *adaptation strength* measure, lexical features have stronger adaptation in the partner adaptation dialog pairs. Syntactic features, taken as a whole, do exhibit significantly greater *adaptation ratios* for partner adaptation than for recency adaptation.

Table 9 reports the same measures as Table 8 over the subset of features from Tables 3, 4, 5. The results on the subset of features that exhibit significant positive adaptation are similar to the results for all features.

5.3 Measuring effect of priming frequency on adaptation

This section describes how *adaptation ratio* and *adaptation strength* depend on the frequency of a feature in the prime dialog. Table 10 shows the average *adaptation ratio* and *adaptation strength* values for varying thresholds on the prime: $prime > baseline, prime > baseline + 1, prime > baseline + 2$. The *adaptation ratio* does not depend on variations in the prime dialog frequencies; however, *adaptation strenth* increases as the thresh-

feature	Adapt. ratio		Adapt. strength	
	partner	recency	partner	recency
across	7.314	4.655	0.285	3.452
sorri	4.180	1.741	0.410	0.161
through	5.642	3.385	0.785	1.285
i	1.714	3.0	7.240	8.573
uh	3.413	5.973	1.054	0.471
sai	1.693	5.642	2.430	4.680
about the	4.478	1.492	0.640	2.016
right-hand side	5.924	3.022	2.099	1.640
when you	5.642	2.987	0.660	0.493
my map	2.418	7.052	1.816	0.416
on my	3.173	6.770	1.328	0.328
to be	0.846	3.847	0.265	1.065

Table 6: Comparison of *adaptation ratio* and *adaptation strength* between partner and recency adaptation dialog pairs for the features that have highest differences between the ratios

feature	Adapt. ratio		Adapt. strength	
	partner	recency	partner	recency
NP->NP PP	1.896	2.6	31.699	17.249
NP->NN	2.963	2.963	0.781	2.656
NP->DT NN	3.048	3.048	0.445	0.695
NP->DT AP NN	2.308	3.077	0.254	0.503

Table 7: Adaptation to chance ratio and adaptation strength for the syntactic features examined by Dubey.

old for the prime dialog increases for both recency and user-primed dialog pairs. This trend illustrates that higher occurrence of a feature in the prime dialog causes stronger adaptation (higher frequency of a feature in target), but has no effect on the probability of adaptation.

6 Conclusion

In this paper, we presented two methods for measuring adaptation in dialog. Our *adaptation ratio* measure, a variation on Church's measure of adaptation, evaluates how likely a feature is to appear in a target document with frequency $>$ average if it appears in the prime document with frequency $>$ average. Our *adaptation strength* measure evaluates the strength of adaptation. These measures have several advantages over those used in previous work. Comparing the frequency to average instead of using a binary 'occurred'/'did not occur' allows us to measure effect on both frequent and infrequent features. We think that our measure of *prior* is more sound for measuring adaptation in a relatively small corpus of dialog pairs. Evaluation of adaptation strength allows us to measure adaptation of a feature in single dialog pair.

We used these measures to compare adaptation in partner- and recency-primed dialog pairs. We showed through a series of experiments using the Maptask corpus that these measures can identify features that exhibit variation and can be used across dialogs to evaluate the presence and strength of partner and recency adaptation.

We are still not satisfied with these measures. Some drawbacks to our measures include:

- The *adaptation strength* measure does not take into account the probability of a feature repeating in the same document; some features may be likely to repeat independent of priming.
- In the *adaptation ratio* measure we cut off features that occurred less than 30% in the prime.

feature	Adaptation ratio		Adaptation strength	
	partner	recency	partner	recency
Stem/POS	2.64	2.71	3.46	3.67*
Stem/bigram	2.99	3.03	1.71	1.91*
Syntactic	2.71	2.92*	4.70*	4.11

Table 8: Adaptation ratio and adaptation strength averaged over all features. * indicates significant difference between partner and recency adaptation ($p<.05$)

feature	Pr(+adapt)/Pr(Chance)		Adapt. Strength	
	partner	recency	partner	recency
Stem/POS	3.36	3.15	3.71	3.82
Stem/bigram	3.86	3.68	1.30	1.62*
Syntactic	3.09	3.36*	5.49*	4.99

Table 9: Adaptation ratio and adaptation strength averaged over significant features listed in Tables 3, 4 and 5. * indicates a significant difference between partner and recency adaptation ($p<.05$)

Taking a different cut-off may influence the result.

We hope to address these issues in future work.

In current work, we are incorporating models of adaptation to syntactic and lexical choice into our RavenCalendar dialog system (Stenchikova et al., 2007). We are creating a tight integration between parsing, dialog management and response generation so that words and syntactic constructions used by the user can be highly salient for the system, and ones used by the system are available for interpretation of user utterances (cf. (Isard et al., 2006)). In experiments with this system, we plan to use our adaptation measures to evaluate user adaptation to system behavior for different system adaptation rates.

References

A. Anderson and et. al. 1991. The HCRC map task corpus. *Language and Speech*, 34:351–366.

L. Bell, J. Boye, J. Gustafson, and M. Wiren. 2000. Modality convergence in a multimodal dialogue system. In *Proceedings of GOTALOG*.

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.

P. Brown and G. Dell. 1987. Adapting production to comprehension: The explicit mention of instruments. *Cognitive Psychology*, 19:441–472.

S. Brennan C. Lockridge. 2002. Addressees' needs influence speakers' early syntactic choices. *Psychonomics Bulletin and Review*. Psych.

T. Chartrand and J. Bargh. 1999. The chameleon effect: The perception-behavior link and social interaction. *Journal of Personality and Social Psychology*, 76:893–910.

K. Church. 2000. Empirical estimates of adaptation: the chance of two noriegas is closer to p/2 than p2. In *Proceedings of the ACL*.

A. Dubey, P. Sturt, and F. Keller. 2006. Parallelism in coordination as an instance of syntactic priming: evidence from corpus-based modeling. In *Proceedings of EMNLP*.

J. Gustafson, A. Larsson, R. Carlson, and K. Hellman. 1997. How do system questions influence lexical choices in user answers? In *Proceedings of Eurospeech*.

W. Horton and R. Gerrig. 2002. Speakers' experiences and audience design: knowing when and knowing how to adjust utterances to addressees. *Journal of Memory and Language*, 47:589–606.

A. Isard, C. Brockmann, and J. Oberlander. 2006. Individuality and alignment in generated dialogues. In *Proceedings of INLG*.

T. Kraljic and S. Brennan. 2005. Prosodic disambiguation of syntactic structure: For the speaker or for the addressee? *Cognitive Psychology*, 50:194–231.

	num	%adapted		avg. adapt. strength	
		partner	recency	partner	recency
p>b	151.7	.14	.17	2.42	2.55
p>b+1	78	.12	.14	3.47	3.59
p>b+2	51.8	.12	.15	3.94	3.82

Table 10: Average distance measures for *adapted* features (Stem/POS only) [1]

R. Kraut, S. Fussell, and J. Siegel. 2003. Visual information as a conversational resource in collaborative physical tasks. *Human-Computer Interaction*, 18(1–2).

M. Pickering and S. Garrod. 2004. Toward a mechanistic psychology of dialog. *Behavioral and Brain Sciences*, 27:169–190. Psych.

M. Pickering, H. Branigan, A. Cleland, and A. Stewart. 2000. Activation of syntactic priming during language production. *Journal of Psycholinguistic Research*, 29(2):205–216.

D. Reitter, F. Keller, and J. Moore. 2006. Computational modelling of structural priming in dialogue. In *Proceedings of HLT/NAACL*.

G. Skantze. 2002. Coordination of referring expressions in multimodal human-computer dialogue. In *Proceedings of ICSLP*.

S. Stenchikova, B. Mucha, S. Hoffman, and A. Stent. 2007. RavenCalendar: A multimodal dialog system for managing a personal calendar. In *Proceedings of HLT/NAACL*.

Token-based Chunking of Turn-internal Dialogue Act Sequences

Piroska Lendvai and **Jeroen Geertzen**

Dept. of Communication & Information Sciences,
Tilburg University, The Netherlands,
{p.lendvai,j.geertzen}@uvt.nl

Abstract

In this study we compare two sequence learning approaches to chunk dialogue acts within a speaker's turn. We assign a dialogue act label to each token in the transcribed speech stream of a dialogue participant, additionally classifying if the token is at the *beginning* of, *inside*, or *outside* that specific dialogue act. Experimental findings show that both our approaches – conditional random fields and memory-based tagging – largely improve over local classification methods, obtaining comparable scores on distinct datasets. We discuss the interplay between transcription granularity of turns and dialogue act chunking.

1 Introduction

Previous supervised learning approaches to dialogue act tagging are typically applied to dialogue units that are pre-segmented on e.g. turn level, utterance level, or functional unit level, the exceptions being (Warnke et al., 1997) and (Zimmermann et al., 2005). However, automatic segmentation into dialogue units is a significant challenge in itself. Even a short speaker turn can contain more than one dialogue act units, for example an agreement in reply to a proposal and an immediate question ('Fine. Which airport?'); on the other hand, multiple turns of the same speaker may feature one single dialogue act, for instance a sequence of statements.

An important aspect of corpus-based approaches is that training data are mostly derived from transcribed speech, where it is common practice to structure the dialogue participants' token stream (typically containing words, but also disfluent elements, non-speech events, symbols for overlapping speech, etc.) into syntactically or semantically complete units, which are then further segmented into turns along speaker change and time line.

In some circumstances of interaction however, like in situations in which interlocutors are under time pressure to communicate, are under stress, or are engaged in a heated discussion, spoken dialogue does not fully proceed in sequence, but often contains simultaneously occurring events, since speakers may cross-react on each other's (incomplete) utterances in a dynamic way. Transcriptions inevitably commit to one or another granularity criterion, and as such superimpose knowledge-based considerations on how to structure dialogue to some extent. In (Traum and Heeman, 1996) the issue of defining utterance units in spoken dialogue is treated extensively.

Arguably, it is easier to automatically assign a dialogue act (DA) to (semantically) complete units than to incomplete ones, and thus the question arises to what extent DA classification generalises across material created by annotation schemes of different DA unit granularity. In the current study we attempt to make the first explorations of this issue by pursuing a boundary-knowledge-lean approach to two differently transcribed dialogue corpora, focusing on turn-internal DA transitions.

The method we advocate is the application of state-of-the art sequence learning approaches to token-based classification of DAs. Our approach

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 174–181,
Antwerp, September 2007.

is to perform sequential tagging based on transcribed words and disfluent elements (henceforth: *tokens*) in streams of utterances up to the point of a speaker change (aka turns). Two supervised classifiers, a memory-based tagger and conditional random fields, are trained to identify each element of the word stream as one of a set of DA types, and also whether the token is an initial or an internal element of a larger DA chunk. This approach can be likened to syntactic phrase chunking, and has been shown to work well for identifying disfluent chunks in spontaneous spoken Dutch discourse (Lendvai et al., 2003).

In the following section we describe the corpora employed in this study, and how token sequences and their contextual attributes were derived from transcribed material. Next, the classification procedure is discussed, where we elaborate on sequence learning as a tagging approach, as well as on the measures of evaluating a chunking task. In Section 4 we present our experimental results, followed by a summary of our findings and pointers to future work.

2 Data

The experiments reported in this paper are carried out on two English language datasets drawn from two corpora, each coding dialogue units in a different way: the Monroe corpus and the MRDA corpus.

The *Monroe corpus* (Stent, 2000) consists of human-human, mixed-initiative, task-oriented dialogues about disaster handling tasks. In each dialogue, the interlocutors are engaged in a collaborative problem-solving, mixed initiative interaction, which involved a scenario at an emergency control centre: an instructor (U) receiving incoming information about a disaster, and a remote subject (S) initially knowing nothing about the task. A typical fragment of these interactions is given in Table 1.

For eight dialogues speech has been manually transcribed, segmented into utterances and turns, and annotated with the DAMSL tag set[1], resulting in a data set of 2,897 speaker utterances that are segmented into 1,701 turns (on average 189 turns per dialogue). Each utterance can have multiple communicative functions in four layers (Allen and Core, 1997); there is almost always at least one function assigned to an utterance. The Monroe corpus is annotated with 13 main DA types that can further contain arguments. We worked with the nine labels contained in the forward-looking and the backward-looking dimension of the annotation. These are: `statement, influence-on-listener, influence-on-speaker, info-request, conventional, other, agreement, understanding, answer`.

$S1$	there are [SIL] three people on a stretcher at the airport
$U1$	mm hm
$S2$	then there's one stretcher [SIL] patient [SIL] at [SIL] the mall
$U2$	+ uh huh [SIL] +
$S3$	+ [SIL] and +
$U3$	here was the heart attack right
$S4$	yeah yeah yeah
$S5$	we should get them to the nearest hospital asap

Table 1: An excerpt from the Monroe corpus.

Because of the nature of the DAMSL scheme, the transcribed utterances in this dataset tend to be long, as DA units are segmented in a rather coarse-grained fashion. It can be guessed however, that interaction between the participants is of a more segmented nature, since overlapping speech is marked by numerous turn-internal $+$ symbols in the transcriptions.

The MRDA corpus (Shriberg et al., 2004) is a companion set of segmentations and annotations on the ICSI Meeting Corpus, which consists of 75 non-scenario based meetings that each are roughly an hour in length. On average, there are about six English speakers, native and non-native, per meeting. Most of the meetings were group discussions about the ICSI meeting recorder project itself or on topics on natural language processing. The sample in Table 2 illustrates an interaction with three dialogue participants.

The utterances in the MRDA corpus have been annotated with a modified version of the SWBD-DAMSL

[1] Annotations are publicly available at `http://www.cs.rochester.edu/research/cisd/resources/monroe/`.

$c1$	um ... so far I have thought of it as sort of adding it onto the modeler knowledge module
$c0$	that is the d-
$c3$	hmm
$c0$	ok
$c0$	yeah

Table 2: An excerpt from the MRDA corpus.

tagset (Jurafsky et al., 1997), in which a dialogue act is a combination of at least one general tag, with a variable number of possible specific tags attached. There are 11 general tags. The MRDA corpus has been used in various segmentation and dialogue act classification studies, e.g. (Zimmermann et al., 2005), and as in most of these studies we worked with dialogue act labels grouped into five types: `backchannels (B)`, `disruptions (D)`, `floorgrabbers (F)`, `questions (Q)`, and `statements (S)`, as well as two miscellaneous labels, `(X)` and `(Z)`.

In this corpus tokens from a speaker are segmented into minimal units that are semantically complete, so that a unit always has only one general DA tag assigned to it. Tags in this dataset are thus mutually exclusive, which is a major difference from the Monroe material. The MRDA data contains 51,452 turns (on average 826 turns per dialogue).

It is important to see that due to these fine-grained DA chunks, the speech stream of one speaker tends be transcribed in a much more scattered way along the course of the interaction than in the Monroe corpus. All three utterances from the speaker on channel 0 in Table 2 would have been transcribed as one utterance in the Monroe corpus, because the DAMSL annotation scheme applied there allows for assigning DA labels on different dimensions, so that a statement and a backchannel could be segmented into one unit. But in the MRDA transcriptions, these token streams are considered as separate units, even with a DA unit of a different speaker inserted between them.

There is an abundance of self-interruptions an other type of disfluencies, overlapping speech, and turn-internal silence in both corpora. The latter two elements are also encoded in markedly different ways in the two datasets: the Monroe transcriptions contain these directly as symbols ($+$ and `[SIL]`, respectively) in the token stream, whereas the MRDA material breaks up the token stream along overlapping speech into separate segments, and encodes silence between tokens by time stamps.

There are a number of other differences between our datasets. First, the DA sets in the two corpora overlap to only a small extent, both in their amount and their aspects: `statement` is a DA in both of them, and there is a `Question` DA type in the MRDA and `Information request` in the Monroe corpus, but the mapping between `Backchannel` in MRDA and `Agreement` as well as `Understanding` in the Monroe material is only partial, whereas the other DA types are difficult to relate across corpora. Additionally, the amount of data in the two datasets differs as well: the Monroe dataset is rather sparse, whereas the MRDA corpus provides thousands of examples to the learners. Finally, the Monroe corpus is a two-party interaction with 'giver and follower' type of roles, whereas the MRDA discussions involve many speakers and a more intertwined flow of interaction.

3 A chunking approach to segmenting dialogue acts

3.1 Classifiers

For the joint learning of the segmentation and labeling, we used two different sequence-based machine learning techniques: *conditional random fields (CRFs)* and *memory-based tagging (MBT)*. Both of these have been shown to be particularly suitable for sequential natural language processing tasks such as part-of-speech (POS) tagging.

CRFs (Lafferty et al., 2001) are probabilistic learners for labeling and segmenting structured data. The algorithm defines a conditional probability distribution over label sequences given a particular observation sequence (in our case a sequence of tokens), rather than a joint distribution over both label and observation sequences. The main advantage of CRFs over e.g. hidden Markov models (HMMs) is their conditional nature, resulting in the relaxation of the independence assumptions that is required by HMMs in order to remain computationally feasible.

We used the CRF++ package with default settings[2].

MBT is a memory-based tagger-generator that generates a sequence tagger on the basis of a training set of labelled sequences, and consecutively can tag new sequences (Daelemans et al., 2003). It has been used to generate POS taggers and various chunkers. MBT can make use of full algorithmic parameters of TiMBL 5.2, a memory-based software package[3].

In our setup, a learner classifies a token from a dialogue (the token under consideration, which we call the *focus token*) in its context of other tokens (the *context tokens*). It depends on internal design how much of a context a sequence learner will consider during classification, we worked with a default token context of 1. For all classifiers we mostly used default settings. It is possible to provide the learners additional information, by means of a vector of features. We discuss our selection of features below.

3.2 Features

Our method for both corpora was to merge all tokens into one single sequence up to a transcribed speaker change. In this way, we preserved a minimum boundary information uniformly for both corpora. In the Monroe dataset a sequence-to-be-chunked on average contains 1.5 DA boundaries, and consists of rather long utterances (e.g., $S4$ and $S5$ in Table 1 would constitute a sequence). In the MRDA dataset, the last two DA units on channel 0 would be merged, as they are transcribed consecutively, but the unit transcribed between $c1$ and $c3$ is regarded as a single-token sequence. By merging the 'utterances' into longer segments of 'turns', we created about 5% less segment boundaries in the MRDA data than in the transcriptions. On average there are 1.8 DA type boundaries in the segments.

The features that we use are straightforward and automatically extractable from the dialogue transcriptions. The majority of these would be internally available from a linearised token stream in a dialogue application as well. Some attributes were derived using some knowledge of transcribed boundaries; this has to do with the fact that although sequence learners can handle a sequence of hundreds of tokens, it is not feasible to feed them an entire dialogue.

Tokens All words were tokenised, dealing with capitalisation, separating and expanding clitics, etc., and subsequently stemmed with a Porter stemmer (Porter, 1980). Apart from taking the word token as a focus feature, we also use the token's part-of-speech tag, automatically obtained by using MBT trained on the Wall Street Journal treebank. We included in the feature vector a context window of 12 left context and six right context elements, both tokens and POS tags. The size of the left context is taken to be the average turn length in tokens, which is estimated 12 for both the Monroe and the MRDA corpus. The context window does not include information contained across the above-explained speaker boundary.

Bag-of-words It has been shown in previous work that redundant encoding of dialogue context may improve the automatic detection of DAs (Lendvai and van den Bosch, 2005). We thus additionally represent lexical context as a bag-of-words (BOW): BOW$left$ contains the lastly uttered 12 words of the current speaker, BOW$leftOth$ contains the most recently uttered 12 words of the speaker that spoke immediately before the focus speaker, and BOW$right$ covers six tokens of right-context for the current speaker only, since it would be incorrect to assume the current speaker to have certainty about what the next speaker will contribute. A threshold on the lexicon size of the BOW has been set to only consider the 200 most frequent word tokens. Note that the BOWs exclude information contained across their own boundaries, and that speaker identity is not encoded.

Silence and overlapping speech For the Monroe data the [SIL] and + markings in the transcriptions were used to derive features. These indicate whether or not an utterance starts or stops with a silence. For the MRDA data, we represented the time elapsed between the previously uttered token in the interaction and the focus token.

3.3 Experimental setup

Our task is to identify in one process for each token in a sequence its DA label, and whether it is a label boundary or not. We represent the DA labels by so-called IOB tags (Tjong Kim Sang and Veenstra,

[2] CRF++ is publicly available at `http://crfpp.sourceforge.net/`

[3] MBT and TiMBL are publicly available at `http://ilk.uvt.nl/`

1999), which is one of the many encoding possibilities. For each DA label a prefix marks whether a token is starting a new DA chunk (B_<DAtype>), is inside a DA chunk (I_<DAtype>), or outside (O_<DAtype>), cf. Table 3. This extended DA label is the class to be guessed by the learners.

	token	Q	S	…	comb.
U	can	I	O	…	I-q
	you	I	O	…	I-q
	see	I	O	…	I-q
	the	I	O	…	I-q
	map	I	O	…	I-q
	have	B	O	…	B-q
	you	I	O	…	I-q
	found	I	O	…	I-q
	it	I	O	…	I-q
S	i	O	I	…	I-s
	can	O	I	…	I-s
	not	O	I	…	I-s
	see	O	I	…	I-s
	it	O	I	…	I-s

Table 3: IOB encoding for questions (Q) and statements (S) in binary classification on the Monroe data.

3.4 Evaluation aspects and metrics

In many previous work on segmentation and classification of dialogue acts, accuracy-based measures such as segmentation and dialogue act error rates have been proposed to asses segmentation and classification performance. Even though these metrics give reasonable insight about performance on the task, higher accuracy or lower error rates do not necessarily imply better performance on DA chunking. Hence we will pay most attention to the traditional measure of information retrieval and chunking: F_1 *score*, a harmonic mean of precision and recall. For comparison with similar work, we additionally report on dialogue act error rate (DER), as described in (Zimmermann et al., 2005): the percentage of misrecognised DAs (i.e., the lower the DER is, the better), where a DA is successfully recognised if both the predicted DA type is correct and the chunk boundaries are successfully predicted. Note that in terms of information retrieval, the DER is none other than the *inverted* DA chunk recall (recall is the proportion of correctly found chunks over the gold-standard amount of chunks). On the token level, we report on the accuracy of predicting the correct IOB tag.

All experiments are carried out separately on the Monroe and on the MRDA datasets. The MRDA dataset allows for multi-class learning, but the Monroe corpus is not annotated with mutually exclusive DAs, yielding over 200, often low-frequent multi-dimensional tags, whose boundaries do not always overlap. Multi-class DA chunking on these data is not straightforward, thus we trained a separate binary classifier for each of the nine occurring DA classes. If we average the results over the classes, we calculate macro averages (in the case of F_1 scores denoted by $F_{1,ma}$). These are in general significantly lower than micro averages that are traditionally reported for chunking tasks. We therefore also report on the F_1 micro score (denoted by $F_{1,mi}$), which is available for the MRDA data results. Accuracy is not affected by this difference of classification method.

4 Experiments and results

On each dataset we run both sequence learners twice: first they have access to the token sequence only, and in a different experiment they can draw on the full feature vector. Additionally, to put the results of CRF and MBL into perspective, we test a baseline method on the DA chunking task, as well as two local classification methods: a Naive Bayes and a k-nearest neighbour approach. The results for Monroe are presented in Table 4 and those for MRDA in Table 5.

4.1 Baselines

A simple majority class baseline is to always guess the majority chunk, which is in both datasets `statement`. This approach labels the beginning of each sequence as `B_statement`, and the rest of the sequence as `I_statement`. We get markedly different scores on the two corpora, since in MRDA the majority of turns include a number of chunks (recall that this material is segmented according to minimal units), whereas in Monroe the segments are typically larger (because the DAMSL annotation scheme allows for assigning multi-level tags to one and the same unit).

When we look to Table 5, we see that for the MRDA dataset this baseline (denoted with *MajChu*) is already rather accurate, (81%), but recalls only a small fraction of the chunks correctly, yielding the relatively low F_1 score (27 points). On the Monroe dataset with separate binary classifiers this labeling clearly is a very bad strategy (8% accuracy, see Ta-

ble 4), since only one out of the nine binary classifiers has a chance to score at all.

Next, we test powerful local classifiers on the DA chunking task. The naive Bayes classifier is probabilistic and assumes feature independence. It often requires only a small amount of training data to be rather effective. We indeed see that on the Monroe dataset, which contains longer and in a sense more complete utterances, this baseline acquires high accuracy (89%), from only knowing the focus token. When it is provided a relatively large and unorganised additional feature set (recall that the feature vector encodes among others three times 200 bits of contextual bags-of-words), its performance is however dramatically undermined. The same trend can be observed on the MRDA set for the naive Bayes classifier.

Our third baseline is computed by running the IB1 algorithm implemented in the TiMBL package. IB1 is a memory-based learning technique, a direct descendant of the classical *k*-nearest neighbour approach to classification. The number of nearest neighbours used in the experiments was set to nine, and the modified value difference metric was employed in the internal weighting of features. The *k*-nearest neighbours voted on the class using the inverse distance weighting parameter. Note that our sequence learner MBT is also set to employ the IB1 algorithm and the above parameters, thus the differences between a local and a sequential application of the same algorithm are directly comparable. Contrary to the performance of the naive Bayes classifier, IB1's F_1 score improves (on MRDA) or at least remains constant (on Monroe) when it can draw on additional features.

4.2 Sequence learners

A direct comparison between the scores from the two datasets in Tables 4 and 5 may not be informative, due to the differences between these, as explained in Section 2. Nonetheless, we can observe trends within each dataset. The F_1 scores of both sequence learners improve largely over all baselines, indicating that sequential approaches are superior to global classification in the DA chunking task.

CRF's performance is affected in the allFeatures setup to its disadvantage on the Monroe corpus (30 vs 23 F_1,ma), whereas on this material MBT scores identically regardless of the features involved (22 and 23 F_1, ma). The best score is 30 points of micro F score, obtained by the CRF algorithm.

On the MRDA data we see a slight improvement over the token-only experiment for CRF (44 vs 41 F_1, mi). In contrast, MBT's scores seem to weaken on the large feature vector (40 vs 47 F_1, mi).

The two sequence learners work in a rather different way inherently, which explains this divergence. On the smaller dataset (Monroe) CRF performs better than MBT, especially in the TokenOnly experiment (30 vs 22 F_1, ma), but it is not the case on the large dataset (MRDA): at least on the single focus token, MBT beats CRF (47 vs 41 F_1), but not in the allFeatures experiment.

In general, we see that the magnitude of performance is in the same range for both datasets, despite that it may be more difficult to find a large number of boundaries of short chunks than to identify longer spans of fewer DA type spans. Note that we have much more data from the MRDA corpus, that probably allows the learners to be better trained.

Arguably, we set a rather hard task for the learners by limiting the token sequence to material from one speaker only, regardless of own and others' previously uttered tokens, and thereby missing all context that an utterance can have. We deliberately formulated this task, and conjecture that the scores we obtained are in fact out-of-context baseline scores to turn-internal DA chunking, and as such are rather high already. Comparison of our results with previous work cannot be straightforwardly done, due to the differences in creating the token sequences that need to be chunked. The obtained DER scores verify the general trend of the sequence learners improving over local classification methods.

We have additionally run experiments to give an impression of the effect of adding more context to the focus token, in the form of the BOW from the immediately previous other speaker (BOW*leftOth*). When splitting down the scores according to DA types, the results indicate that on some DA types there is indeed an improvement over the AllFeatures approach (although not over the TokenOnly experiment), from this additional information. The figures for the two datasets are reported in Table 6 and Table 7.

	tokenOnly				allFeatures			
	Acc	$F_{1,ma}$	$F_{1,mi}$	*DER*	Acc	$F_{1,ma}$	$F_{1,mi}$	*DER*
MajChu	8	3	-	97	8	3	-	97
NBay	89	18	-	84	77	6	-	91
IB1	87	13	-	87	85	13	-	83
CRF	88	30	-	74	84	25	-	77
MBT	86	22	-	80	86	23	-	82

Table 4: Classification performance of nine binary classifiers on the Monroe corpus.

	tokenOnly				allFeatures			
	Acc	$F_{1,ma}$	$F_{1,mi}$	*DER*	Acc	$F_{1,ma}$	$F_{1,mi}$	*DER*
MajChu	81	5	27	78	81	5	27	78
NBay	82	15	16	79	8	7	2	98
IB1	79	1	23	80	83	23	37	61
CRF	83	27	41	65	84	27	44	60
MBT	84	30	47	57	84	25	40	61

Table 5: Classification performance on the MRDA corpus, computed in multi-class learning of seven DA types.

5 Conclusions and future work

In this study we aimed to explore if it is feasible to take a boundary-knowledge-lean approach to jointly segment and label dialogue acts in two corpora. Dialogue processing is dependent on transcribed material, but the representation and segmentation of DA units in dialogue transcriptions is not standardised. Supervised learning of DAs is however dependent on labelled material, where variations of encoding the flow of dialogue supposedly bias the mapping of a dialogue unit to a DA type.

We proposed to refrain from encoding knowledge-based unit boundaries as much as possible, and based DA processing on tokens as basic units. Sequence learning procedures were applied to each token uttered by a speaker, including disfluencies, and a token was classified either as chunk-initial or chunk-internal with respect to a limited set of DA types in the SWBD-DAMSL, respectively the MRDA annotation scheme.

Two sequence learners, a memory-based tagger and conditional random fields, were trained and tested on the task of segmenting tokens into turn-internal DA chunks. They could draw on a set of straightforward features, or on the token sequence only. We showed that sequence learning methods are suitable for DA chunking, improving over the results of a chunk majority baseline and local classifiers. The best chunk F_1 score we obtained is 47 on the transcribed tokens of MRDA spoken discussions, using the MBT sequence tagger in multi-class learning of seven DA types. (Note that two out of the seven employed DA labels are highly sparse meta-labels, on which the classifiers typically yield near-zero scores, which severely affects the F_1 scores.)

Our sequence learning methods that performed token-based DA chunking were able to produce comparable results on rather distinctly transcribed dialogue datasets, both on the MRDA meeting transcriptions and the more traditionally transcribed Monroe scenario dialogues that feature longer turns and a giver-follower dialogue style. Comparing the utility of the lexical token only versus a large bag of straightforward contextual features, we conclude that in our setup lexical items carry the best information for assigning chunk-initial and chunk-internal DA types.

We regard our method as a baseline technique to objectively investigate the role of context in DA chunking. Our plans include explorations on how larger context, including prosodic phenomena, affects performance of sequence learning approaches on DA chunking.

		Agr	Und	Answ	Stat	IList	ISpk	IReq	Conv	Oth
CRF	TokenOnly	54	60	23	33	26	21	23	6	26
	AllFeatures	45	52	16	29	19	15	16	32	3
	Token + BOW*leftOth*	47	52	17	30	15	12	12	0	2
MBT	TokenOnly	53	58	6	39	13	9	18	0	6
	AllFeatures	46	51	11	35	18	15	11	0	17
	Token + BOW*leftOth*	38	43	10	32	11	8	7	16	14

Table 6: F scores per DA type on the Monroe corpus using different feature sets and sequence learners.

		Backch	Disr	Floorgr	Quest	Statem	X	Z
CRF	TokenOnly	69	14	40	23	38	0	5
	AllFeatures	68	1	38	20	44	0	21
	Token + BOW*leftOth*	66	1	28	10	38	0	14
MBT	TokenOnly	70	16	39	34	46	0	4
	AllFeatures	59	16	31	26	39	0	4
	Token + BOW*leftOth*	64	18	38	31	42	0	5

Table 7: F scores per DA type on the MRDA corpus using different feature sets and sequence learners.

Acknowledgements

The authors thank Mary Swift, Joel Tetreault, and Amanda Stent for providing the Monroe data, and Elizabeth Shriberg for sharing the ICSI-MRDA dataset. We thank Antal van den Bosch, Sander Canisius, and Erik Tjong Kim Sang for insightful discussions and software help.

References

James Allen and Mark Core. 1997. Draft of DAMSL: Dialog act markup in several layers. Unpublished manuscript.

Walter Daelemans, Jakub Zavrel, Antal van den Bosch, and Ko van der Sloot. 2003. MBT: Memory based tagger, version 2.0, Reference guide. ILK research group technical report series 03-13, Tilburg.

Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 2004. TiMBL: Tilburg Memory Based Learner, version 5.1, Reference guide. Technical Report 04-02, ILK, Tilburg University, Tilburg, The Netherlands.

Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, Institute of Cognitive Science, University of Colorado, USA.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Piroska Lendvai, Antal van den Bosch, and Emiel Krahmer. 2003. Memory-based Disfluency Chunking. In *Proceedings of DISS-03, Disfluency in Spontaneous Speech Workshop*, pages 63—66.

Piroska Lendvai and Antal van den Bosch. 2005. Robust ASR lattice representation types in pragma-semantic processing of spoken input. In *Proceedings of the AAAI Spoken Language Understanding Workshop, SLU-2005*, pages 15–22.

Martin F. Porter. 1980. An algorithm for suffix stripping. In *Program*, *3*(14), pages 130–137.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The ICSI meeting recorder dialog act (MRDA) corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100.

Amanda J. Stent. 2000. The Monroe corpus. Technical Report TR728/TN99-2, University of Rochester, Rochester, UK.

Erik F. Tjong Kim Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the Ninth Conference on European chapter of the Association for Computational Linguistics (EACL)*, pages 173–179, Morristown, NJ, USA. Association for Computational Linguistics.

David R. Traum and Peter A. Heeman. 1996. Utterance Units in Spoken Dialogue. In *ECAI Workshop on Dialogue Processing in Spoken Language Systems*, pages 125–140.

Volker Warnke, Ralf Kompe, Heinrich Niemann, and Elmar Nöth. 1997. Integrated dialog act segmentation and classification using prosodic features and language models. In *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH-97)*, pages 207–210.

Matthias Zimmermann, Yang Liu, Elizabeth Shriberg, and Andreas Stolcke. 2005. Toward joint segmentation and classification of dialog acts in multiparty meetings. In Steve Renals and Samy Bengio, editors, *MLMI*, volume 3869 of *Lecture Notes in Computer Science*, pages 187–193. Springer.

A Comprehensive Disfluency Model for Multi-Party Interaction*

Jana Besser and **Jan Alexandersson**
DFKI GmbH
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
GERMANY
{jbesser,janal}@dfki.de

Abstract

We present a disfluency model derived from analysing transcriptions of the AMI meeting corpus. Our model goes beyond previous work in that it discriminates several classes that are elsewhere regarded the same. Furthermore, we provide a formal account for naturally occurring phenomena that are rarely modeled in other schemes. Our annotations show significant occurrences of these classes. An evaluation of the annotations from four different annotators reveals a high agreement, $\kappa = 0.92 - 0.93$, $AC1 = 0.93$.

1 Introduction

Speech differs highly from written language. Spoken language contains a lot of linguistic irregularities, so called *disfluencies* (Henceforth DF), e.g., (Shriberg, 1994). In general, disfluencies can be classified on different levels, but in this work, we will solely treat syntactic and grammatical errors according to standard syntax and grammar. Hence, we present a classification scheme for speech DFs that defines DF classes according to their surface structure.

Previous approaches have failed to cover the existent phenomena to a satisfying degree. To our knowledge, the presented scheme is more fine-grained than previous schemes and covers a larger set of DF types. In fact, this scheme models almost 99% of the phenomena found in our corpus.

In a data-driven approach, we identified the existing phenomena via examinations of meeting transcriptions from the AMI[1] meeting corpus (McCowan et al., 2005). The corpus contains unrestricted and uncontrolled human-human discussions, recorded in business meetings. The meetings were held in English, but not all participants were native speakers.

We consider only phenomena that actually lead to the interruption of the syntactic or grammatical fluency of an utterance. This excludes meta comments and certain stylistic devices from the classification. Our approach is only concerned with the structural correctness of an utterance and thus no analysis of the semantic or pragmatic impacts of DFs were considered. The underlying psychological processes were neither examined.

The disfluency classification scheme was developed as part of the AMI project. The project's goal was to develop technology to support and enrich communications between individuals and groups of people. Some research topics of the project are 1) *Definition and analysis of meeting scenarios*, 2) *Infrastructure design, data collection and annotation*, 3) *Processing and analysis of raw multi-modal data*, 4) *Processing and analysis of derived data*, and 5) *Multimedia presentation*, see also (McCowan et al., 2005). The project was, e.g., concerned with automated meeting summarizations. Disfluency detection and correction is a nearly mandatory matter for

* This research is funded by the EU 6th Framework Program under grants FP6-506811 (AMI) FP6-033502 (i2home) The responsibility lies with the authors.

[1] AMI = "Augmented Multi-party Interaction", see http://www.amiproject.org and its successor AMIDA = "Augmented Multi-party Interaction with Distance Access", see http://www.amidaproject.org.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 182–189,
Antwerp, September 2007.

reaching this goal.

The paper is organised as follows: In the next section (2) the classification scheme is thoroughly described. Section 3 presents a scheme for DF annotations in XML format. In section 4 an evaluation of DF annotations according to some metrics is conducted. In 5 we present and discuss previous work. Finally, we conclude the paper with section 6.

2 A Classification Scheme

This section will give definitions for all DF classes that we have identified for the classification scheme. For some classes, XML-annotated examples will be presented. The annotations follow an annotation scheme for DFs that we have developed based on the DF classifications (see 3).

```
lorem ipsum
<DF>
    <RM>erroneous material</RM>
    <interregnum>editing material</interregnum>
    <RS>correction</RS>
</DF>
consectetuer adipiscing elit.
```

Figure 1: The general schema of a disfluency consists of the disfluency material—*reparandum* (RM)—followed by the *interregnum* (IM). The third part called *reparans* (RS) constitutes the actual repair.

Beforehand, we illustrate the general surface structure of a DF, see figure 1: DFs usually consist of three parts. The first part contains the "erroneous", disfluent material, that will be corrected later on, the *reparandum* (RM). The RM is followed by the *interregnum* (IM), a term which is adapted from (Shriberg, 1994). The third part of a DF is the repairing section, the *reparans* (RS). The RM denotes the whole stretch of material from the beginning of the DF's first part to the beginning of the IM, not only the words that are replaced or corrected in the reparans. This is due to the fact that replacing the RM with the RS has to result in a meaningful, grammatically correct sentence, which would not always be the case if only the modified parts were denoted as RM.

The DFs are grouped into three sets based on their surface similarity: *uncorrected* DFs, *deletable* phenomena, and *revision*s, see figure 2. Only *revision*s can optionally contain an IM whereas RS is omitted in all *uncorrected* phenomena. We divided the *deletable* DFs into two subgroups: *delay* and *parenthesis*. DFs of type *delay* are sounds, not words, that hold up the speech flow, e.g. for gaining time to plan the utterance. *Parenthesis* DFs are real words that do not contribute to the utterance's meaning.

In what follows, we provide definitions for all DFs and examples for some:

2.1 Uncorrected

The following two conditions have to be fulfilled by a DF to be classified as uncorrected:

1. The speaker's original utterance may only contain a RM. The RS (and thus the IM) is missing.
2. The content of the RM is relevant for the sentence and may not just be deleted. Therefore, the correction of the DF implies creating a suitable RS.

There are three types of uncorrected utterances:

Mistake: A *mistake* is an uncorrected speech error, which leads to a grammatically incorrect sentence. Examples are agreement errors and other grammatical errors.
Omission: The speaker omitted a word, which would be necessary for the segment in order to be grammatically correct.
Order: The segment's word order has to be changed in order to make the utterance grammatically correct.

2.2 Deletable

The following two conditions have to be fulfilled by a DF to be classified as uncorrected:

1. The DF's content can be discarded from the utterance without impact on the utterance's propositional content.
2. The DF does only contain a RM and no correction, which is quite naturally following from 1, since non-contentional expressions can hardly be corrected.

There are six types of *deletables*. The types *Hesitation* and *stuttering* are grouped into *Delay*, and *EET* and *DM* are grouped into the class *Parenthesis*.

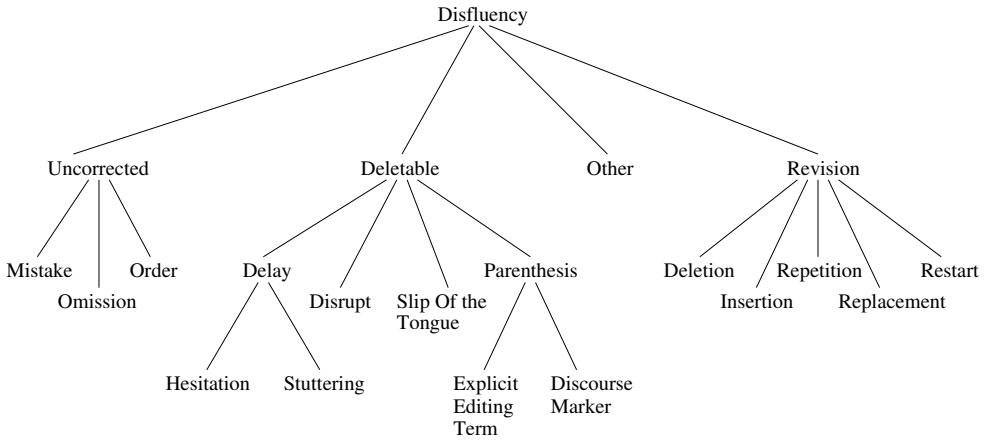


Figure 2: The hierarchy of our disfluencies where the classes are grouped into three main branches, *uncorrected*, *deletable* and *revision*s. The classes *stuttering* and *hesitation* are specializations of *delay* and *EET* and *DM* are specializations of *Parenthesis*.

Hesitation: Hesitations are rather sounds than words. They are usually used in order to gain time and are thus expressions of the speaker's cogitation. Typical *hesitation*s are: uh, uhm, eh, em, mm etc.

Stuttering: Stutterings are non-lexical word fragments, which are similar to the beginning of the next fully articulated word.

Example:

(1) <stutter>N n</stutter> no, I don't think so.

As the example shows, sequences of stuttering sounds are seen as one single *stuttering* and are not treated separately.

Disruption: Denotes whole or partial segments that do not form a meaningful statement and are so fragmentary that no meaning can be established by adding information. The fragmentary material may not occur at the beginning of a segment.

Slip Of the Tongue (SOT): *SOT*s are speech sounds, syllables or syllable fragments which do not form a correct (existing) word and cannot be classified as *stuttering*.

Example:

(2) looking at the <sot>tex</sot> technical functions...

Discourse Marker (DM): *DM*s do not contribute to the content of an utterance, but have a rather discourse related function. Their usage gives the speaker time to think of what to say next and to hold the turn. Examples are: I mean, so, well, you know, like etc.

Explicit Editing Term (EET): *EET*s are roughly the same expressions as *DM*s but they always stand in the IM of a *revision*.

Example:

(3) <replace>
<RM>The design of</RM>
<eet>or</eet>
<RS>the point of</RS>
</replace>
putting two sensors on each side

2.3 Revisions

*Revision*s are phenomena, where both RM and RS are given by the speaker. They could also be named "self-corrections" or "self-repairs".

Deletion: The RS repeats some parts of its RM, while omitting some other material. The deleted material has to be from the central region of the RM.

Example:

(4) But
<delete>
<RM>it's really not</RM>
<RS>it's not</RS>
</delete> functional.

Insertion: The RS repeats the RM with supplementary information added at some point. The added information may not be the last material in the RS.
Example:

(5) <insert>
<RM>What else it</RM>
<RS>what else do we want it<RS>
</insert>
to do?

Repetition: Those are expressions that occur several times consecutively. This does not include word fragments. RM and RS have to contain exactly the same material.
Replacement: The RS repeats some material of the RM. The remaining information is substituted with new material.
Restart: The RS replaces all the information given in the RM. It restarts the region of the sentence, which was started by the RM. The *restart* does not have to occur at the beginning of the sentence.
Example:

(6) How would we go about
<restart>
<RM>making</RM>
<RS>getting</RS>
</restart>
rid of our weak points?

Other: Those are DF structures that do not match any of the specified classes.

2.4 Complex Disfluencies

DFs are called *complex* if some of the contained material belongs to more than one DF. An example is shown in (7) where the first "she" is both RS to the first DF and RM to the second.

(7) he she she went

When a DF is completely contained in the RM or RS of another DF, it is called a *nested* DF. The annotation is simply carried out starting from the inmost DF and then proceeding stepwise outwards:

(8) But then to go back
<replace>
<RM>to the</RM>
<RS>to
<sot>th</sot>
<stutter>s</stutter>
something
<RS>
</replace>
along those things.

Troublesome events are *complex partially chained DFs* (Shriberg, 1994), where not all of one DF's output is the input to another DF, see (9)[2].

(9) show me the flight the delta flight delta fare

Here "the delta flight" substitutes "the flight" by an insertion and "delta fare" replaces "delta flight". The complication is that the first DF's output (and second DF's input) is not "delta flight" but "the delta flight". This means, that "delta fare" actually replaces "the delta flight". Thus "the" is omitted resulting in the corrected sentence "show me delta fare".

This arises due to the fact that our annotations are made from left to right. Our annotation scheme does not yet provide a solution for this. Thus, in the case of a partially chained DF some loss of information must be accepted, see (Shriberg, 1994) for a discussion on this issue.

3 Annotation

In order to evaluate the reliability and clearness of the DF class definitions, we have annotated a subset of four meetings from the AMI meeting corpus (McCowan et al., 2005) based on an annotation manual we developed. The meetings contained a total of 2876 segments as identified during dialogue act (DA) annotation. These 2876 segments were parsed with the LKB parser (Copestake, 2002). The 792 segments (27.5%) that did not receive a parse were extracted and considered for manual annotation by four annotators. On average, 74% of these 792 segments received a DF annotation. In what follows, we call these segments "corpus A".

At the time of writing, the four meetings used in creating corpus A have been completely reannotated. Additionally, three more meetings have been annotated. For these annotations, the complete meetings were considered for annotation by the annotators. In total these meetings contain 4718 segments. 2095 segments, corresponding to 44% (ranging from 28% to 52%), were annotated with at least one disfluency.

3.1 Statistics and Metrics

We have applied two different statistics in order to rate the inter-annotator agreement: the κ-statistic

[2] taken from (Shriberg, 1994)

and the AC1-formula (Gwet, 2002). The reason for using AC1 is that it is insensitive to disproportionate distribution of class frequencies. Otherwise they share the same co-domain.

The formulae have been adapted to the comparison of multi-category annotations by two annotators. There, N stands for the total number of compared annotations, M is the number of categories, i is an integer (1,..., M), AGR_i is the number of agreements on category i, A_i is the number of annotations into category i by annotator A, and B_i is the number of annotations into category i by annotator B.

κ-statistic

$$\kappa = \frac{p - e(\kappa)}{1 - e(\kappa)}$$

p is the total agreement of the annotators, whereas e(κ) computes their agreement by chance (value between 0 and 1). p and $e(\kappa)$ are calculated in the following way:

$$p = \frac{\sum_{i=1}^{M}(AGR_i)}{N}, \quad e(\kappa) = \sum_{i=1}^{M}\left(\frac{A_i}{N}\right)\left(\frac{B_i}{N}\right)$$

AC1-statistic

$$AC1 = \frac{p - e(\gamma)}{1 - e(\gamma)}$$

Again, p is the total agreement of the annotators. It is calculated in the same way as in the κ-statistic. The chance agreement e(γ) computes a value between 0 and 0.5:

$$e(\gamma) = \frac{\sum_{i=1}^{M} P_i(1 - P_i)}{M - 1}, P_i = \frac{(A_i + B_i)/2}{N}$$

The number of agreements and disagreements as well as the number of compared annotations (N) were gained by applying the following four metrics to the gathered data:

Strict comparison: Two DF annotations are equal if both annotators have marked the same stretch of material with the same disfluency type. If the DF contains RM and RS (and IM), also those have to be absolutely equal.

Strict comparison without DF type: The conditions are the same as for the first metrics, but the annotated DF type may be different. If, e.g., annotator A classified the phenomenon as a *replacement* whereas B classified it as a *restart*, the annotations would count as equal anyway. This is motivated by the existence of some relatively similar DF classes, which can be hard to distinguish.

Result oriented comparison: In this metrics the regions, which were marked for deletion by the annotators, are compared. This includes RMs, *hesitation*s, *stutterings*, *DM*s, *EET*s, *SOT*s and *disruption*s. If the same regions are marked with one of these tags, they are counted as equal.

In this way the metrics accounts for the fact that if the same regions of a segment are erased, then the final outcome of the correction is the same, no matter, which class assignments were made.

Liberal concerning IM: This metrics compares annotations in the same way as the first metrics (strict comparison) but *EET*s are treated in a special way: Two annotations containing an *EET* are also counted as equal, if the boundaries of the *EET*s are the same but the *EET* is annotated as part the RM in both or in one of the annotations . The annotations are also considered equal if the a region was labelled as *EET* in one annotation but as *DM* in the other.

It should be noted that *uncorrected* DFs were excluded from the result-oriented evaluation, since the comparison of their corrections can be quite hard to assess and would often some semantic analysis. For example, if annotator A adds "an" as missing determine (RS), and annotator B "the", their annotations are different from a shallow perspective, but they could be seen as equal regarding functional perspective.

4 Evaluation

The results from the comparisons according to the different metrics were gathered in confusion matrices. We then calculated the κ- and the AC1-value for each matrix with the statistics described above. The total agreement was derived by calculating the average of all computed κ- vs. AC1-values of all meetings. This gave the results presented in table 1. Column 4 shows the percentage of the DF instances that had equal boundaries and were also assigned the same DF type. It becomes clear that once the annotators identified the same boundaries for a

Table 1: Inter-annotator agreement according to both statistics for strict and liberal comparison, the total agreement, and the percentage of DFs that were assigned to the same class.

	κ-value	AC1-value	Total agreement	Same DF type
Strict comparison	0.924	0.934	0.958	93.8 %
Liberal concerning IM	0.930	0.936	0.967	94 %

DF, the agreement on the class assignment was very high. The demanding task was rather to agree on the boundaries of a phenomenon. It can, e.g., be quite hard to decide where the reparans of a DF ends. Also the decision on the class assignment of a phenomenon can influence the definition of its boundaries.

Additionally, we have computed the AC1- and κ-value for the three main classes in the DF hierarchy: *uncorrected*, *deletable* and *revision*. We received a κ-value of 0.998 and an AC1-value of 0.999 for the strict comparison. Thus, the annotators agreed invariably on the DF assignment to the main classes.

The evaluation of the result-oriented metrics yielded that the annotators agreed to 77.5 % on the material that would have to be removed for correction purposes.

Altogether, the annotators identified an average of 1206 DFs in the 792 segments. This means that the mean number of DFs per DA was 1.5.

Table 2 shows the number of occurrences of each DF type, along with the total and proportional annotator agreement for each class. The DF classes are not equally distributed and there is a high discrepancy between the most common phenomenon (*hesitation*s) and the scarcest one (*deletion*). The six most prevalent DF classes constitute 67 % of the encountered phenomena, whereas the five least common types correspond to only 5 % of the DF instances.

Classes rarely mentioned in previous schemes, e.g., *mistake* and *omission* are prevalent in our corpus. However, *order* only occurs in about 1% of the annotated segments. (Finkler, 1997) considers these three phenomena as one: "uncorrected". However, our findings support the division. Finally, *disruption*s are very common but seem to be hard to annotate reliably. A similar low reliability is found for *order*. This is probably due to their inhomogeneous structure. However, it is our hope that an annotator will improve the performance over time.

Table 2: The average number of annotations of a certain DF type in corpus A and corpus B. "**%**" depicts the proportion of a certain DF-type in the corpus and "**% Agr**" depicts the percentage of cases in which all four annotators agreed on the DF annotation.

Corpus	**A**			**B**	
DF	$\Sigma/4$	**%**	**% Agr**	Σ	**%**
Delete	2	0.0	0.0	2	0.0
Disrupt	143	11.9	11.2	509	11.9
DM	165	13.7	52.7	642	15.0
EET	16	1.3	43.8	43	1.0
Hesit	202	16.8	84.7	842	19.7
Insert	15	1.2	33.3	38	0.8
Mistake	79	6.6	34.2	259	6.0
Omiss	68	5.6	35.3	276	6.4
Order	12	1.0	16.7	32	0.7
Other	14	1.2	7.1	44	1.0
Repeat	177	14.7	72.3	641	15.0
Replace	69	5.7	39.1	165	3.8
Restart	41	3.4	24.4	190	4.4
SOT	124	10.3	78.2	366	8.5
Stutter	79	6.6	82.3	223	5.2
Σ	1206	100	—	4272	100

4.1 Discussion

The annotator agreement on the classes *hesitation*, *stuttering*, SOT and *repetition* is especially high. The structure of these phenomena is easy to identify, independent of their context. Even if they occur within complex multi-nested DF structures. The lowest agreement lies on the classes *disruption*, *other* and *order*. The assignment to these categories is to a high degree based on the annotator's estimation of the phenomenon. Moreover, the structure of these phenomena is inhomogeneous and cannot clearly be defined. Furthermore, we counted only phenomena as equal that were annotated with ex-

actly the same boundaries. For the regarded classes it is particularly hard to say for sure where they end and start. Annotation differences though do not necessarily have an impact on the meaning of the sentence after the correction has been applied, since different annotations can still result in the same correction.

Such facts could be accounted for via a less strict comparison of the annotations. Phenomena that overlap widely but do not have exactly the same boundaries could be counted as equal. The presented work does not include such an approach, since we could not implement a corresponding metrics due to time limitation. Such tolerant metrics is complicated by the existence of complex disfluencies. They imply that overlapping DFs do not always need to correspond to each other. They can even be assigned to different layers of a complex DF. The inmost DF of one complex DF does not have to be the inmost DF of another annotator's (complex) DF.

5 Related Work

Several researchers have investigated speech disfluencies before with different underlying motivations. There are four basic types of disfluencies that have been identified by most previous classification schemes, e.g. (Liu et al., 2003), (Shriberg, 1999), (Heeman and Allen, 1999), and (de Mareüil et al., 2005). Those are *filler*s (e.g. *filled pause*s, *discourse marker*s, and *editing term*s), *repetition*s, *fresh start*s and *modification*s. *Fresh start*s denote cases in which an utterance is abandoned and a new one is started. *Modification*s are self-corrections, in which the RS modifies the RM and has a strong correspondence to the RM.

Only some schemes go beyond this classification. One of them was developed in (Shriberg, 1994). Her thesis is an absolute foundation in this research field. She elaborated regularities in the production of DFs and created a detailed classification scheme of DF phenomena. The scheme has been adapted by several other approaches, for example by (Zechner, 2001) and (Strassel, 2004). Zechner has summarization in mind whereas the main motivation in (Strassel, 2004) is rich metadata annotation for the production of maximally readable transcripts. Another valuable and elaborate classification—also based on the findings in (Shriberg, 1994)—is presented in (Finkler, 1997). His main motivation is the incremental generation of natural language utterances.

Although some of these schemes are quite elaborated, they do not give a formal account for all disfluency phenomena occurring in our corpus. For example, in (Shriberg, 1994), no DFs were considered where material has to be added or changed in order to gain the sequence the speaker (presumably) intended. Thus phenomena, which are classified as *Omission* or *Order* in our scheme are not covered by her classification. These phenomena have been mentioned in (Carbonell and Hayes, 1983), but are only informally described.

We also applied changes to some prevalent definitions of certain DF phenomena. An example for this is *repetition*. In Shriberg's approach these include also cases, where the first element of the repetition (the RM) is a word fragment or a mispronunciation. Our work is more rigid: a DF is only classified as *repetition* in case the RM consists of full words and RM and RS contain exactly the same material. Fragments are instead modelled in *stuttering*, *SOT* and *replacement*.

Moreover, our schema is more fine grained than the related work mentioned here. This concerns e.g. the *uncorrected* classes and the class *disruption*. Some schemata, do not differenciate between our *stuttering* and *slip-of-the-tongue* either.

6 Conclusions

Our aim has been to develop a classification scheme for disfluencies occurring in spontaneous speech. With the goal of serving as a theoretical basis for all applications that have to deal with such phenomena, our scheme extends previous work on this topic, e.g., (Shriberg, 1994; Finkler, 1997; Strassel, 2004; Heeman and Allen, 1999).

We identified the existent phenomena by examining transcriptions of business meetings from the AMI meeting corpus (McCowan et al., 2005). Our investigations led to an identification of 15 DF classes that we defined according to the disfluencies' surface structure. We developed a hierarchy of disfluencies and divided them into three subgroups. The subgroups are *uncorrected* DFs, *deletable* DFs,

and *revision*s. *Uncorrected* DFs are phenomena that were not corrected by the speaker. For these DFs, a correction has to be created to eliminate the irregularity. *Deletable* DFs are removed in order to correct the utterance. *Revision*s are DFs where the speaker made a self-correction.

We also developed an annotation manual for disfluencies. Four annotators annotated 792 segments from the AMI meeting corpus that could not be parsed by the LKB parser. It turned out that the number of DFs identified by the annotators was quite high (1206 DFs in a total). This supports the fact that disfluencies are very common in spontaneous speech. On the other hand, this might be due to the high number of non-native speakers in our corpus.

We defined four metrics for comparing the annotations. The metrics counted only phenomena as equal that were annotated with exactly the same boundaries. Annotations with the same boundaries showed a high agreement (0.93) with respect to the DF type. We also computed the agreement for the three main classes in the DF hierarchy. There we yielded a score of 0.999. The inter-annotator agreement was measured by the κ-statistic and the AC1-formula (Gwet, 2002). In this experiment, they both yielded approximately the same value. The result-oriented metrics, comparing the output of the annotations, gained 77.5% agreement.

Our evaluation showed that the DFs are not equally distributed ranging from 16.8% (*hesitation*) to approximately 0% (*deletion*). There is also a discrepancy in the accuracy of identifying the different DFs. The proportion of identically annotated DFs varied strongly. We attribute this to the DF structures rather than to the clearness of the annotation manual. This is motivated by the fact that the agreement was much higher for phenomena that have an easily recognised structure.

Future work will include more annotation of complete meetings and an evaluation thereof. The manual has already received some update, and we expect this to happen again. We plan to publish the annotations along with the complete AMI/AMIDA corpus.

References

Jaime G. Carbonell and Philip J. Hayes. 1983. Recovery strategies for parsing extragrammatical language. *Comput. Linguist.*, 9(3-4):123–146.

Ann Copestake. 2002. Implementing typed feature structure grammars. CSLI Publications, Stanford, CA.

Philippe Boula de Mareüil, Benoît Habert, Frédérique Bénard, Martine Adda-Decker, Claude Barras, Gilles Adda, and Patrick Paroubek. 2005. A quantitative study od disfluencies in french broadcast interviews. In *Proceedings od DiSS '05*, pages 27–32, Aix-en-Provence, France, September.

Wolfgang Finkler. 1997. *Automatische Selbstkorrektur bei der inkrementellen Generierung gesprochener Sprache unter Realzeitbedingungen*. Ph.D. thesis, Saarland University.

Kilem Gwet. 2002. Kappa Statistic is not Satisfactory for Assessing the Extent of Agreement Between Raters. Series: Statistical Methods For Inter-Rater Reliability Assessment, No. 1. `http://www.stataxis.com`.

Peter A. Heeman and James F. Allen. 1999. Speech Repairs, Intonational Phrases and Discourse Markers: Modeling Speakers' Utterances in Spoken Dialogue. *Computational Linguistics*, 25(4):527–571.

Yang Liu, Elizabeth Shriberg, and Andreas Stolcke. 2003. Automatic Disfluency Identification in Conversational Speech Using Multiple Knowledge Sources. In *Proceedings EUROSPEECH*, pages 957–960, Geneva.

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 Measuring Behaviour 2005 symposium on Annotating and Measuring Meeting Behavior*, Wageningen, The Netherlands.

Elizabeth Shriberg. 1994. *Preliminaries to a Theory of Speech Disfluencies*. Ph.D. thesis, University of Berkeley, California.

Elizabeth Shriberg. 1999. Phonetic Consequences of Speech Disfluency. In *Proceedings of the International Congress of Phonetic Sciences*, pages 619–622, San Francisco.

Stephanie Strassel. 2004. Simple Metadata Annotation Specification V6.2. Linguistic Data Consortium. http://www.ldc.upenn.edu/Projects/MDE.

Klaus Zechner. 2001. *Automatic Summarization of Spoken Dialogues in Unrestricted Domains*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, November.

Experimental modeling of human-human multi-threaded dialogues in the presence of a manual-visual task

Alexander Shyrokov, Andrew Kun
Electrical and Computer Engineering
University of New Hampshire
{shirokov, andrew.kun}@unh.edu

Peter Heeman
Center for Spoken Language Understanding
Oregon Health and Science University
heeman@cslu.ogi.edu

Abstract

We discuss the design and preliminary results of an experiment for modeling human-human multi-threaded dialogues. We found that participants tend to complete adjacency pairs in dialogues before switching to a new dialogue thread. We also have indications that, in the presence of a manual-visual task, the difficulty of the task influences switching between dialogue threads.

1 Introduction

Humans can carry on multi-threaded spoken dialogues in which several dialogue threads overlap in time. Humans can do this while they are involved in manual-visual tasks, such as driving. For example a driver can discuss the weather with one passenger in the car, while periodically talking to another passenger about directions. However, it is an unsolved problem how to enable human-computer spoken multi-threaded interaction, especially while the human participant is involved in a manual-visual task. Our major hypothesis is that this problem can be solved by applying models of human-human interactions to human-computer interactions.

In this paper, we describe an experimental approach to model human-human spoken interactions in the presence of a manual-visual task, specifically driving a simulated car. We performed experiments with pairs of participants who were involved in an ongoing task but periodically needed to switch to an interrupting task. In the ongoing task one of the participants drove a simulated car and received verbal navigation instructions from the other participant who had a map of the simulated world but was not in the driving simulator. The interrupting task was initiated by a visual stimulus presented to the driver in the simulator and it had to be completed verbally. The driver had to initiate the switch to the new dialogue thread verbally.

We were interested in three elements of the model of this human-human interaction. First, we investigated how the urgency of the interrupting task affects the timing of the interrupting task. We hypothesized that more urgent interruptions will be dealt with more quickly.

Next we looked in which dialogue state participants choose to initiate a switch to the interruption dialogue thread. We define the state of the dialogue in terms of whether the speakers are in the midst of an adjacency pair.

Finally, we explored the relationship between driving task difficulty and how quickly participants initiated an interruption. From our previous experiments we know that driving task difficulty has a significant influence on the performance of spoken tasks in the simulator. Therefore, we expect that driving task difficulty (and in general manual-visual task difficulty) has to be incorporated in our model. We hypothesized that participants will respond to interruptions more quickly when the driving task is less difficult.

2 Related Research

We investigate the use of multi-threaded dialogues similarly to cognitive load studies in which participants switch between two separate manual-visual tasks (McFarlane, 1999). In our prior work we explored the timing of switches between dialogue threads in human-human conversations, depending on the urgency of the interrupting task (Heeman, 2005). We found that some participants varied the place within a dialogue where they switch to the

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 190–193,
Antwerp, September 2007.

interrupting task, depending on the urgency of the interrupting task. However, the tasks were artificial, that of playing a card game and determining whether a player has a certain colored shape on their computer screen. Furthermore, only gross discourse structure was examined, rather than the local discourse phenomena of adjacency pairs.

3 Experiment

Two participants took part in each session. One was assigned the role of a police officer, and the other was the dispatcher. The police officer operated the driving simulator, while the dispatcher sat in another room. Participants used headsets and microphones to communicate with each other. This task was related to the ongoing work at the University of New Hampshire on the Project54 system. The Project54 system integrates devices in police cruisers and provides a speech user interface to these devices (Kun, 2004). Our use of navigation as the ongoing task was inspired by the Map Task experiments (Anderson, 1991).

3.1 Ongoing task

We conducted our experiments in a high-fidelity driving simulator with a 180º field of view and a motion base, as shown in Figure 1. The simulator presented a city scenario with two-lane (one lane for each direction) roads (7 meters wide). The city consisted of sixteen intersections organized in a four-by-four grid, as shown in Figure 2. The limits of the area were marked with construction barrels. The officer was instructed not to drive past the barrels. Participants were not allowed to travel faster than 30 mph and they were required to stop at every stop sign, in order to lower the possibility of motion sickness (Mourant, 2000).

The dispatcher had a map with four marked locations that the officer had to visit. In order to ensure that the officer and the dispatcher engaged in a dialogue with each other, some city streets were also blocked with construction barrels, as shown in Figure 2. The barrel locations changed dynamically depending on the officer's location. The officer had to explain to the dispatcher if a street was open, so the dispatcher could make corrections to his/her instructions.

Figure 1. Driving simulator.

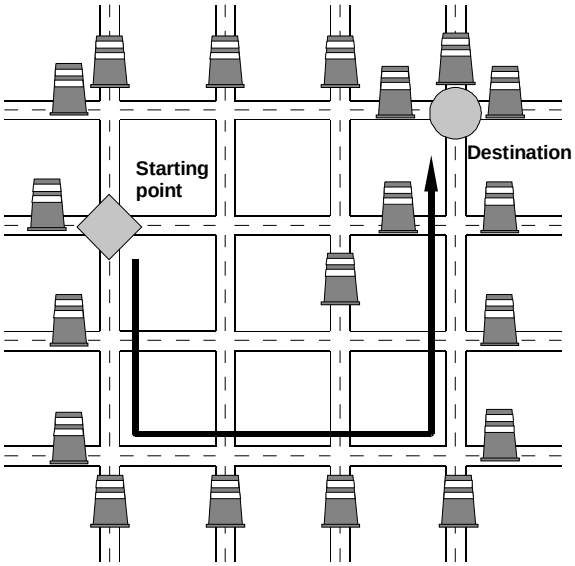


Figure 2. Blocked streets and possible path.

3.2 Interrupting task

Periodically the officer was presented with a visual stimulus. The officer then had to tell the dispatcher about the visual stimulus. Visual stimuli consisted of a text message and a progress bar. We used two different text messages for the interrupting task to make sure that the participants shift their attention from the ongoing task.

A progress bar was used to inform the officer about the urgency of the stimulus. Visual stimuli had one of two urgency levels. Officers had to respond to "urgent" visual stimuli (47% of all visual stimuli) within 10 seconds. For "non-urgent" visual stimuli officers had 20 seconds to respond. If the officer failed to inform the dispatcher about a visual stimulus within these time limits, the car would stop moving for 10 seconds (these car break-downs were controlled by the experimenter). Participants were told to complete the ongoing task as fast as

possible, and car break-downs provided an additional incentive to inform the dispatcher about visual stimuli quickly.

3.3 Procedure and participants

Participants were given an overview of the simulator, and were trained to perform the ongoing task and then both tasks. Training took about 10 minutes. Participants then performed the actual experiment which lasted about 30 minutes. At the end, participants completed questionnaires and received a debriefing. The experiment was completed by ten participants (five pairs) between 20 and 43 years of age. The average age of the participants was about 30 years and 30% were female.

4 Analysis and Results

We recorded the speech of all participants, as well as the car position. Vehicle data was collected at 10 Hz, resulting in about 90,000 vehicle data points for 2.5 hours of driving. We also recorded the time the visual stimuli appeared and synchronized these times with the audio recording of the participants. The five pairs of participants were presented with a total of 286 visual stimuli.

We analyzed three aspects of the data. First we looked at the average response time of the officer to urgent and non-urgent visual stimuli. We found no significant difference in response time depending on the urgency of the interruption (one tail t-test p=0.434), possibly because participants did not realize that some interruptions were more urgent than others.

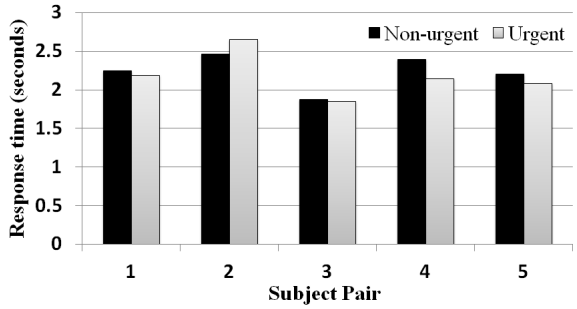


Figure 3. Average response times.

Figure 3 shows the plot of average response times for different participants. The response times are slower (average around 2.8 seconds for all cases) than reported by Tsimhoni et al. (2001) (average 1.3 seconds), who investigated reading messages on a heads-up display while driving. A reasonable explanation for this is that in our experiment the officer was engaged in verbal communication with the dispatcher and did not pay as close attention to the messages as the participants in the study of Tsimhoni et al. Even more likely, the officer was complying with established conventions in human-human dialogue, and so waited for a suitable point in the interaction. This waiting for an opportunity to speak slowed down his/her response.

We next analyzed what dialogue states allow people to initiate a dialogue thread switch. Figure 4 shows a model of the local dialogue state of the ongoing task, based on sequences of adjacency pairs (Schegloff, 1973). In the first part of an adjacency pair, either the dispatcher or the police officer speaks (e.g. poses a question). We denote the first part with "a" when the dispatcher speaks and with "e" when the officer speaks. After a pause (denoted with "b" after the dispatcher speaks and "f" after the officer speaks), the dialogue continues with the second part of the adjacency pair. The second part is denoted with "c" when the officer speaks and with "g" when the dispatcher speaks. Finally, when the second part ends, and before the next first part begins, we have a pause in the dialogue, denoted with "d."

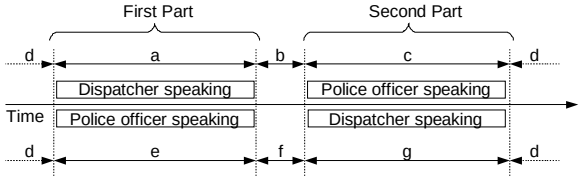


Figure 4. Adjacency pairs.

We coded each presentation of a visual stimulus with "a" through "g" based on where it happened with respect to the model in Figure 4. Each presentation resulted in the eventual initiation of an interruption (switch to the interrupting task). We also coded the interruption initiated by the officer based on where it happened with respect to the model in Figure 4. Note that the officer could have ignored the visual stimulus, but this happened only 5 out of 286 times, hence we did not further consider these cases. This left us with 7 x 7 = 49 possible types of interruption. In this paper, we decided to focus on interruptions in which the stimulus occurred during the first part of an adjacency pair ("a" or "e") as this is the point in the local discourse structure that is the most embedded.

When a stimulus is presented during the officer's first part ("e") 10% of the time the officer interrupts his/her own first part ("ee"). In 25% of the cases he/she completes the first part and then introduces the interruption ("ef"). In about 1% of the cases the officer introduces the interruption during the dispatcher's second part ("eg"). Most often, in 51% of the cases, the officer waits until after the adjacency pair is over ("ed"). In about 11% of the cases the officer introduces the interruption during the first part of the next adjacency pair when the dispatcher is speaking ("ea"). Finally, in 3% of the cases he/she interrupts after the dispatcher's first part in the next adjacency pair ("eb").

When the stimulus is presented while the dispatcher is speaking the first part ("a"), the officer interrupts immediately in about 23% of the cases ("aa") and after the first part in about 26% of the cases ("ab"). Again, most often, 40% of the time, the interruption came after the adjacency pair was over ("ad"). In about 2% of the cases each, the interruption came in the next adjacency pair during or after the officer's first part.

The above data shows that the officer often waited to initiate the interrupting task until after the adjacency pair was done. This might account for the difference between the average response times in this study and the one reported by Tsimhoni et al (2001).

Finally, we also looked at the average response time of officers during difficult and easy driving conditions. We defined difficult driving as driving within a radius of 10 meters of the center of an intersection. We found that officers on average responded slower under difficult driving conditions, however, our findings were not statistically significant. Note that the officers spent only about 8% of their time driving through the intersections and thus, on average this resulted in 5 visual stimuli out of 57 being presented in difficult driving conditions.

5 Conclusion and Future Directions

In this paper, we tried to determine the conventions that humans follow in initiating a switch to a new dialogue thread. We found that when the stimulus to signal the interruption was in the first part of an adjacency pair, participants either immediately interrupted the first part, or waited until the conclusion of the adjacency pair. This might indicate that participants were trying to avoid having the first part of an adjacency pair pending during a thread switch, so that there is a simpler discourse context to resume to. However, more analysis is needed to fully explore this issue, including examining other stimulus points, and distinguishing between different types of adjacency pairs.

Our analysis also shows that we need to further revise our task setup. We need to revise the experimental setup so that the urgency of the interrupting task is more realistic. We also need to better balance the easy with the difficult driving segments in order to better understand the impact of driving difficulty.

Acknowledgements

This work was funded by the National Science Foundation under grant IIS-0326496.

References

A. Anderson, M. Bader, E. Bard, E. Boyle, G. M. Doherty, S. Garrod, S. Garrod, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thompson and R. Weinert. 1991. *The HCRC Map Task Corpus,* Language and Speech, 34:351-366.

Peter Heeman, Andrew L. Kun, Fan Yang and Alexander Shyrokov. 2005 *Conventions in Human-Human MultiThreaded Dialogues: A Preliminary Study,* IUI'05.

Andrew L. Kun, W. Thomas Miller, III and William H. Lenharth. 2004. *Computers in police cruisers*, IEEE Pervasive Computing, 3(4):34-41.

D. McFarlane. 1999. *Coordinating the Interruption of People in Human-Computer Interaction*, Angela Sasse and Chris Johnson, Eds. Human–Computer Interaction.

Ronald R. Mourant and Thara R. Thattacherry. 2000. *Simulator Sickness in a Virtual Environments Driving Simulator*, Proceeding of the 44th Annual Meeting of the Human Factors and Ergonomics Society, 534-537.

O. Tsimhoni, P. Green and H. Watanabe. 2001. *Detecting and Reading Text on HUDS: Effects of Driving Workload and Message Location*, Paper presented at the 11th Annual ITS America Meeting, Miami, FL.

E. A. Schegloff and H. Sacks. 1973. *Opening up closings*, Semiotica VIII: 4: 290-327.

Modeling Vocal Interaction for Text-Independent Classification of Conversation Type

Kornel Laskowski
interACT
Universität Karlsruhe
Karlsruhe, Germany
kornel@ira.uka.de

Mari Ostendorf
Dept. of Electrical Engineering
University of Washington
Seattle WA, USA
mo@ee.washington.edu

Tanja Schultz
interACT
Carnegie Mellon University
Pittsburgh PA, USA
tanja@cs.cmu.edu

Abstract

We describe a system for conversation type classification which relies exclusively on multi-participant vocal activity patterns. Using a variation on a well-studied model from stochastic dynamics, we extract features which represent the transition probabilities that characterize the evolution of participant interaction. We also show how vocal interaction can be modeled between specific participant pairs. We apply the proposed system to the task of classifying meeting types in a large multi-party meeting corpus, and achieve a three-way classification accuracy of 84%. This represents a relative error reduction of more than 50% over a baseline which uses only individual speaker times (i.e. no interaction dynamics). Random guessing on this data yields an accuracy of 43%.

1 Introduction

An important and frequently overlooked task in automatic conversation understanding is the characterization of conversation type. In particular, search and retrieval in multi-participant conversation corpora stands to benefit from indexing by broad conversational style, as tending towards one or more speech-exchange prototypes (Sacks et al, 1974) such as interactive seminar, debate, formal business meeting, or informal chat. Current state-of-the-art speech understanding systems are well-poised to tackle this problem through up-stream fusion of multiparticipant contributions, following automatic speech recognition and dialog act classification. Unfortunately, such reliance on lexical information limits the ultimate application of conversational style classification to only a handful of languages with well-developed lexical components, notably English.

In the current work, we attempt to address this limitation by characterizing conversations in terms of their patterns of on-off vocal activity, referred to as *vocal interaction* by the psycholinguistic community (Dabbs and Ruback, 1987). In doing so, we rely only on the joint multi-participant vocal activity segmentation of a conversation (Renals and Ellis, 2003), and ignore other features. The text-independent features we explore here can of course be combined with text-dependent cues, and prosodic and/or speaker cues, depending on the reliability of these components.

To the best of our knowledge, there is currently little if any work on the continuous modeling of vocal interaction for conversations with arbitrary numbers of participants. Some very recent research exists with goals related to those in this work, most frequently focusing on the classification of time-dependent, evolving phenomena. Examples include the recognition of meeting states and participant roles (Banerjee and Rudnicky, 2004), the detection of interaction groups in meetings (Brdiczka et al., 2005), the recognition of individual and group actions in meetings (McCowan et al, 2005), and the recognition of participant states (Zancanaro et al, 2006). Modeling multi-participant vocal interaction to improve vocal activity detection in meetings was first explored in (Laskowski and Schultz, 2006) and elaborated in (Laskowski and Schultz, 2007); it has

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 194–201,
Antwerp, September 2007.

since been explored for privacy-sensitive data collection in more general settings (Wyatt et al, 2007). The rare examples of time-independent characterization of conversations in their entirety, as pursued in the current work, include the detection of conversational pairs (Basu, 2002) and the classification of dominance in meetings (Rienks and Heylen, 2005).

We begin this paper by proposing a computational framework which allows for the modeling of interactions among specific participants. We propose several time-independent interaction features, together with a robust means for computing them. Finally, we apply the proposed text-independent classification system to the task of meeting type classification. Our results show that features extracted from the multi-participant segmentation of a conversation can be successfully used for classifying meeting type through the observed conversational style.

2 Bayesian Framework

We introduce the notion of a *group* of participants, which we denote as $\mathcal{G}$ and which we define to be a specific ordering of all $K \equiv \|\mathcal{G}\|$ participants in a particular conversation $\mathcal{C}$. Each conversation is of exactly one type $\mathcal{T}$, from among $N_{\mathcal{T}}$ possible types. Participants are drawn without replacement from a potentially unknown population $\mathcal{P}$, of size $\|\mathcal{P}\|$. In general, $\|\mathcal{P}\| > \|\mathcal{G}\|$.

$\mathcal{G}\,[k]$, for $1 \leq k \leq K$, is an attribute of the kth participant; k represents a particular cardinal ordering of participants in group $\mathcal{G}$, which is immutable for the duration of a meeting (in this work, k is the channel number). $\mathcal{G}$ may be unique in $\mathcal{P}$, i.e. it may represent a specific participant; alternately, it may represent a category of participant, such as age group, social standing, or vocalizing time rank. When participants are unique in $\mathcal{P}$, the number of unique groups $N_{\mathcal{G}} = \|\mathcal{P}\|!/\left(\|\mathcal{P}\| - \|\mathcal{G}\|\right)!$ is simply the number of permutations of $\|\mathcal{P}\|$ taken $\|\mathcal{G}\|$ at a time.

Our observation space is the complete vocal interaction on-off pattern description for conversation $\mathcal{C}$, a discretized version of which we denote as $\mathbf{q}_t$ for $1 \leq t \leq T$, where T is the duration of the conversation. Our goal in the present work is to extract from $\mathbf{q} = \{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T\}$ a feature vector $\mathbf{F} \equiv f(\mathbf{q})$ which will discriminate among the $N_{\mathcal{T}}$ different conversation types under study.

We classify the type $\mathcal{T}$ of conversation $\mathcal{C}$, given observations $\mathbf{F}$, using:

$$\begin{aligned}
\mathcal{T}^* &= \arg\max_{\mathcal{T}} P\left(\mathcal{T} \,|\, \mathbf{F}\right) \\
&= \arg\max_{\mathcal{T}} \sum_{\mathcal{G}} P\left(\mathcal{G}, \mathcal{T}, \mathbf{F}\right) \\
&= \arg\max_{\mathcal{T}} \sum_{\mathcal{G}} P\left(\mathcal{T}\right) \times \qquad (1) \\
&\qquad \underbrace{P\left(\mathcal{G} \,|\, \mathcal{T}\right)}_{\substack{\text{Membership} \\ \text{Model}}} \times \underbrace{P\left(\mathbf{F} \,|\, \mathcal{G}, \mathcal{T}\right)}_{\substack{\text{Behavior} \\ \text{Model}}} .
\end{aligned}$$

The behavior model in Equation 1 is responsible for the likelihood of $\mathbf{F}$, describing the behavior of the participants of $\mathcal{G}$ during a conversation of type $\mathcal{T}$. The membership model provides a prior distribution for participant presence in conversations of type $\mathcal{T}$.

3 Vocal Interaction Features

We propose to extract interactional aspects of multiparticipant conversations by studying the presence of vocal activity for all participants at a fixed analysis frame rate. After some limited initial experimentation, we have chosen to use a frame shift of 100 ms. We consider two mutually exclusive vocal activity states, vocalizing ($\mathcal{V}$) and not vocalizing (i.e. silent, $\mathcal{N}$). Figure 1 graphically depicts the discretization of a multichannel segmentation, which allows us to treat a particular conversation as the output of a simple Markov process $\mathbf{q}$ over an alphabet of 2^K symbols, with

$$\mathbf{q}_t \;\in\; \mathbf{\Psi} \times \mathbf{\Psi} \times \mathbf{\Psi} \times \ldots \times \mathbf{\Psi} \qquad (2)$$

of K products, where $\mathbf{\Psi} \equiv \{\mathcal{N}, \mathcal{V}\}$, and t is the time index of the frame.

3.1 Feature Design

In the current work, we assume $\mathbf{q}$ to be a first-order Markov process which can be described by symbol transition probabilities

$$a_{ij} \;=\; P\left(\mathbf{q}_{t+1} = \mathbf{S}_j \,|\, \mathbf{q}_t = \mathbf{S}_i\right) . \qquad (3)$$

Furthermore, we assume that participants behave independently of each other, given their immediately preceding joint vocal activities,

$$a_{ij} \;=\; \prod_{k=1}^{K} P\left(\mathbf{q}_{t+1}\,[k] = \mathbf{S}_j\,[k] \,|\, \mathbf{q}_t = \mathbf{S}_i\right) . \qquad (4)$$

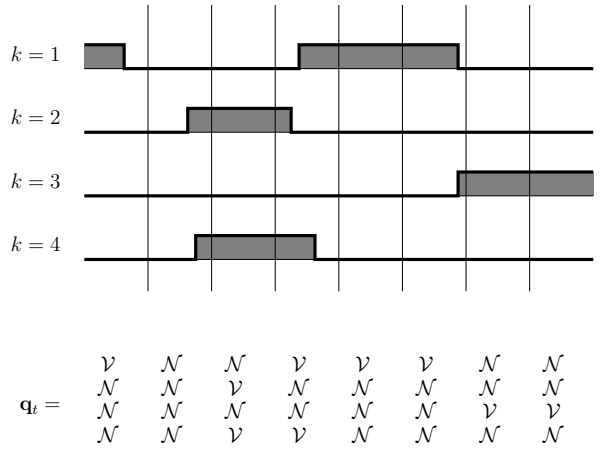


Figure 1: Discretization of multichannel segmentation references by assigning $\mathcal{V}$ for participant k at time t if that participant vocalizes for more than 50% of the duration of the frame centered at t, and $\mathcal{N}$ otherwise.

We propose to characterize the vocal behavior of participants over the entire course of conversation $\mathcal{C}$ using a subset of the probabilities a_{ij}. The features we explore, shown in Equations 5 to 8, represent the probability that participant k initiates vocalization during silence (VI), the probability that participant k continues non-overlapped vocalization (VC), the probability that participant k initiates overlap (OI) while only participant j vocalizes, and the probability that participant k continues vocalizing in overlap (OC) with participant j only, respectively. For this work, we neglect cases where more than one participant (other than j) is vocalizing at time t before participant k starts vocalizing, since such instances are rare.

The probabilities in Equations 5 to 8 can be estimated directly using a maximum likelihood (ML) criterion by accumulating bigram counts matching the event classes in each equation. For simplicity, we set the probabilities for which the conditioning context is never observed to 0.5.

In characterizing an entire conversational group of K participants, the feature vector $\mathbf{F}$ consists of K one-participant features of type f_k^{VI} and K one-participant features of type f_k^{VC}, as well as $K^2 - K$ two-participant features of type $f_{k,j}^{OI}$ and $K^2 - K$ two-participant features of type $f_{k,j}^{O}C$. This results in a total of $N_{\mathbf{F}} = 2K^2$ features per conversation; we note that conversations vary in the participant number K and therefore in their feature vector size.

3.2 Feature Estimation using the Ising Model

We contrast ML estimation of features with estimation which relies on a particular form of parameter tying, under an asymmetric infinite-range variant of the Ising model (Glauber, 1963). Canonically, the Ising model is used to study an ensembles emergent macroscopic properties, which are due to the microscopic interactions among its very large number of binary particles; we apply it here to study the emergent vocal interaction patterns of K participants. The modified Ising model is easily implemented as a single-layer neural network (Hertz et al., 1991) of K input units, K output units, and a sigmoid transfer function,

$$y_k(\mathbf{x}) \;=\; \frac{1}{1 + e^{-\beta\left(\sum_{j=1}^{K} w_{k,j} x_j + b_k\right)}} \quad , \quad (9)$$

where β is a parameter which is inversely proportional to the pseudo-temperature; we set it here to unity for convenience. x_j are the elements of vector $\mathbf{x}$, $w_{k,j}$ are the elements of a weight matrix $\mathbf{W} \in \Re^{K \times K}$, and b_k are the elements of a bias vector $\mathbf{b} \in \Re^{K}$. We show this network in Figure 2. When presented with an input vector $\mathbf{q}_t$, the network produces at each output unit the quantity

$$P(\,\mathbf{q}_{t+1}[k] = \mathcal{V} \,|\, \mathbf{q}_t = \mathbf{S}_i\,) \;=\; y_k(\mathbf{S}_i) \;. \quad (10)$$

In computing $y_k(\mathbf{S}_i)$, $\mathcal{V}$ and $\mathcal{N}$ are mapped to 1 and 0, respectively.

The network is characterized by the parameters $\mathbf{W}$ and $\mathbf{b}$, which can be learned from $\mathbf{q}_t$, $1 \leq t \leq T$, using a standard first-order or second-order gradient descent technique, for example. At each time frame, the current $\mathbf{q}_t$ binary vector can be used as a "pattern", with the subsequent $\mathbf{q}_{t+1}$ binary vector as the "target"; there are a total of $T-1$ such pattern-target pairs. The appropriate objective function for outputs representing multiple (conditionally) independent attributes is the binomial error (Bishop, 1995). To distinguish from features estimated using ML, as described in the previous section, we henceforth refer to features estimated using the Ising model as "NN".

$$f_k^{VI} = P(\mathbf{q}_{t+1}[k] = \mathcal{V} \mid \mathbf{q}_t[i] = \mathcal{N} \quad \forall \quad 1 \leq i \leq K) , \tag{5}$$

$$f_k^{VC} = P(\mathbf{q}_{t+1}[k] = \mathcal{V} \mid \mathbf{q}_t[k] = \mathcal{V}, \mathbf{q}_t[i] = \mathcal{N} \quad \forall \quad i \neq k, 1 \leq i \leq K) , \tag{6}$$

$$f_{k,j}^{OI} = P(\mathbf{q}_{t+1}[k] = \mathcal{V} \mid \mathbf{q}_t[j] = \mathcal{V}, \mathbf{q}_t[i] = \mathcal{N} \quad \forall \quad i \neq j, 1 \leq i \leq K) , \quad j \neq k , \tag{7}$$

$$f_{k,j}^{OC} = P(\mathbf{q}_{t+1}[k] = \mathcal{V} \mid \mathbf{q}_t[k] = \mathbf{q}_t[j] = \mathcal{V}, \mathbf{q}_t[i] = \mathcal{N} \quad \forall \quad i \neq j, i \neq k, 1 \leq i \leq K) , \quad j \neq k . \tag{8}$$

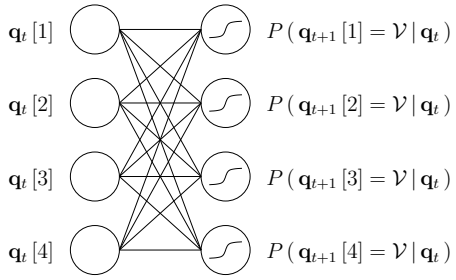


Figure 2: Infinite-range Ising model for predicting conditionally independent probabilities of activation at time $t+1$ given activations at time t, for a conversation with four participants; for clarity, bias connections are elided.

In closing this section, we note that the proposed interaction features have a particularly prosaic form under this model, when $\mathcal{N} = 0$ and $\mathcal{V} = 1$:

$$f_k^{VI} = \frac{1}{1 + e^{-b_k}} , \tag{11}$$

$$f_k^{VC} = \frac{1}{1 + e^{-b_k - w_{k,k}}} , \tag{12}$$

$$f_{k,j}^{OI} = \frac{1}{1 + e^{-b_k - w_{k,j}}} , \tag{13}$$

$$f_{k,j}^{OC} = \frac{1}{1 + e^{-b_k - w_{k,j} - w_{k,k}}} . \tag{14}$$

Furthermore, the total number of parameters to be estimated from segmentation data is $K(K+1)$, rather than $2K^2$ for the bigram ML model.

4 Modeling Groups

In this section we describe the structure, parameter estimation, and probability evaluation for the membership and the behavior models as introduced in Equation 1.

4.1 Behavior Model

We assume conditional independence among the elements of the feature vector $\mathbf{F}$,

$$\mathbf{F} = \bigcup_{k=1}^{K} \left\{ f_k^{VI}, f_k^{VC}, \bigcup_{j \neq k}^{K} \left\{ f_{k,j}^{OI}, f_{k,j}^{OC} \right\} \right\} , \tag{15}$$

such that

$$P(\mathbf{F} \mid \mathcal{G}, \mathcal{T}) = \tag{16}$$
$$\prod_{k=1}^{K} P\left(f_k^{VI} \mid \theta^{VI}_{\mathcal{T},\mathcal{G}[k]} \right) P\left(f_k^{VC} \mid \theta^{VC}_{\mathcal{T},\mathcal{G}[k]} \right) \times$$
$$\prod_{j \neq k}^{K} P\left(f_{k,j}^{OI} \mid \theta^{OI}_{\mathcal{T},\mathcal{G}[k],\mathcal{G}[j]} \right) P\left(f_{k,j}^{OC} \mid \theta^{OC}_{\mathcal{T},\mathcal{G}[k],\mathcal{G}[j]} \right) .$$

In the above, each θ represents a single one-dimensional Gaussian mean μ and variance Σ pair. These parameters are maximum likelihood estimates from the f_k and $f_{k,j}$ values in a training set of conversations, smoothed towards their global values.

4.2 Membership Model

Equation 1 allows for the inclusion of a prior probability on the presence and arrangement of participants with respect to channels. Although participants may have tendencies to sit in close proximity to certain other participants, we ignore channel preference in the current work. We employ the simple membership model

$$P(\mathcal{G} \mid \mathcal{T}) = \frac{1}{Z_{\mathcal{G}}} \prod_{k=1}^{K} P(\mathcal{G}[k] \mid \mathcal{T}) , \tag{17}$$

where $Z_{\mathcal{G}}$ is a normalization constant which ensures that $\sum_{N_{\mathcal{G}}} P(\mathcal{G} \mid \mathcal{T}) = 1$. We set each factor $P(\mathcal{G}[k] \mid \mathcal{T})$ to the ML estimate for participant $\mathcal{G}[k]$ in the training data. For example, if $\mathcal{G}[k]$ represents an identifier unique in $\mathcal{P}$, i.e. a name, then $P(\mathcal{G}[k] \mid \mathcal{T})$ is simply the proportion of meetings

of type $\mathcal{T}$ attended by the participant with that name. To allow the model to hypothesize rarely observed participants in the training material, we set this probability no lower than 0.1, a factor selected empirically without extensive validation.

4.3 Search

Equation 1 calls for the exhaustive enumeration of all possible groups $\mathcal{G}$. As mentioned in Section 2, there are $N_{\mathcal{G}} = \|\mathcal{P}\|! / (\|\mathcal{P}\| - \|\mathcal{G}\|)!$ different groups, which may make such enumeration intractable. Since we are not interested in automatically classifying participants, clustering participants in the training material and thereby reducing $\|\mathcal{P}\|!$ offers a simple means of limiting the magnitude of $N_{\mathcal{G}}$.

In the current work, we choose to cluster participants by training models not for specific participants, but for participant rank in terms of vocalizing time proportion. This makes the attribute $\mathcal{G}[k]$ unique in $\mathcal{G}$ rather than in $\mathcal{P}$. For each training conversation, we rank participants in terms of the overall proportion of time spent in state $\mathcal{V}$, in descending order, such that participant rank 1 refers to that participant who vocalizes most often during the conversation in question. This form of clustering also eliminates the problem of estimating models for specific participants which appear in only a handful of conversations.

Since a test conversation of K participants contains participant ranks $\{1, 2, \ldots, K\}$ and no others, the enumeration of $N_{\mathcal{G}}$ unique participant groups $\mathcal{G}$ in Equation 1 is replaced by an enumaration of $K! = \|\mathcal{G}\|!$ unique rank groups. However, we note that under this simplification, the membership model has only a small impact.

5 Classification Experiments

5.1 Data

In our experiments, we use the ICSI Meeting Corpus (Janin et al., 2003), consisting of 75 unscripted, naturally occurring multi-party meetings. There are 3 aspects which make this corpus attractive for the current work. First, it is larger than most multi-party conversation corpora. This is important because, in our framework, each meeting represents one data point. Second, meeting participants are drawn from a pool of 52 speakers, several of whom occur in more than one meeting type. Finally, meetings are not fixed in participant number, allowing us to demonstrate the generalization of our methods to arbitrary conversational group sizes.

$\mathcal{T}$	#	$\|\mathcal{P}\|$	$\|\mathcal{G}\|$		
			mod	min	max
Bed	15	13	6	4	7
Bmr	29	15	7	3	9
Bro	23	10	6	4	8

Table 1: Characteristics of the three ICSI meeting types considered: number of meetings (#); size of population from which participants are drawn ($\|\mathcal{P}\|$); mode (mod), minimum (min) and maximum (max) number of participants ($\|\mathcal{G}\|$) per meeting type $\mathcal{T}$.

67 of the meetings in the corpus are of one of three distinct meeting types, Bed, Bmr, and Bro, representing different projects, with different purposes for holding meetings. This is reflected in differences between patterns of vocal interaction; for example, Bmr meetings consist of more free-form discussion, presumably among peers, than either Bed or Bro meeting types. In contrast, the latter two types exhibit more asymmetry in participant roles than do Bmr meetings, and therefore the more easily inferable social structure. Furthermore, there are three speakers in the corpus which attend both Bro and Bmr meeting types, and one speaker which attends both Bed and Bmr meeting types; Bro and Bed types, however, have disjoint attendee subpopulations. A participant which appears in multiple meeting types may affect the overall interaction styles of the two types to be less distinct. This is especially true if he or she attends the majority of meetings of both types, as is the case for two of the participants which attend both Bmr and Bro meetings.

We present several additional characteristics of these three meeting types in Table 1. We ignore the remaining 8 meetings in the corpus, representing types of which there are too few exemplars for modeling. As Table 1 shows, the prior distribution over the 3 considered types is such that random guessing yields a 43% three-way classification accuracy.

We obtain the vocal interaction record $\mathbf{q} =$

$\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_T\}$ for each of the 67 meetings by discretizing their reference segmentations. The latter were produced by: (1) generating a talk spurt segmentation through forced alignment of transcribed words (available as part of the ICSI MRDA Corpus (Shriberg et al, 2004)), and bridging of inter-word gaps shorter than 0.3 s; (2) inferring a segmentation for transcribed laughter from the forced alignment of surrounding words, and manually segmenting isolated bouts of laughter (as described in (Laskowski and Burger, 2007)); and (3) merging the talk spurt and laugh bout segmentations. Fully automatic inference of the vocal interaction record, from audio, is beyond the scope of the current work.

5.2 Baseline Performance

To assess the difficulty of the problem, we propose a baseline which relies only on the proportion of vocalizing time, f_k^T, for each participant k. This is a frequently studied quantity for describing conversational style (Burger et al., 2002) and for assessing the performance of speaker diarization systems (Jin et al., 2004) (Mirghafori and Wooters, 2006).

The classification accuracy of the baseline, using the framework described by Equations 1, 16 and 17, is 65.7%. This performance is achieved with leave-one-out classification, using 66 meetings for training and one for testing, 67 times. The accuracy figures in this and in the subsequent section should be treated as estimates on a development set; since the longitudinal nature of the ICSI corpus is relatively unique, it is has not been possible to construct a fair evaluation set without significantly depleting the amount of training material.

We note that, as mentioned in Subsection 4.3, the membership model has negligible impact when participant vocalizing rank is used as the clustering criterion during training. This condition identically affects all of the experiments which follow, allowing for an unbiased comparison of the proposed vocal interaction features.

5.3 Feature Comparison

We present several leave-one-out experiments in order to evaluate the utility of each of the VI, VC, OI, and OC feature types separately, without f_k^T, estimating them from the multichannel reference segmentation for each meeting using both maximum likelihood (column 2), and the proposed neural network model (column 4). The results show that classification using ML-estimated single-participant features f_k^{VI} and f_k^{VC} outperforms classification using NN-estimated features. However, NN estimation outperform ML estimation when it comes to the two-participant features $f_{k,j}^{OI}$ and $f_{k,j}^{OC}$. This result is not surprising, since vocalization in overlap is much more rare than vocalizing alone, rendering maximum likelihood estimation of overlap behavior uncompetitive without additional smoothing.

Feature(s)	ML Estimation		NN Estimation	
	w/o f_k^T	w/ f_k^T	w/o f_k^T	w/ f_k^T
baseline	—	65.7	—	65.7
f_k^{VI}	59.7	67.2	56.7	65.7
f_k^{VC}	62.7	77.6	56.7	71.6
$\langle f_{k,j}^{OI} \rangle_j$	35.8	52.2	64.2	67.2
$\langle f_{k,j}^{OC} \rangle_j$	53.7	67.2	64.2	80.6
$f_{k,j}^{OI}$	41.8	46.3	67.2	64.2
$f_{k,j}^{OC}$	61.2	68.7	73.1	79.1
all	61.2	64.2	74.6	82.1
opt	—	—	74.6	83.6

Table 2: Leave-one-out meeting type classification accuracy using various feature combinations within the proposed Bayesian framework. "opt" consists of the features f_k^{VI}, $f_{k,j}^{OI}$, and $f_{k,j}^{OC}$.

In addition to the two-participant interaction features $f_{k,j}^{OI}$ and $f_{k,j}^{OC}$ described in Section 3, we also show the performance of summary single participant features $\langle f_{k,j}^{OI} \rangle_j = \sum_{j=1}^{K} f_{k,j}^{OI}/K$ and $\langle f_{k,j}^{OC} \rangle_j = \sum_{j=1}^{K} f_{k,j}^{OI}/K$, which average the overlap behavior of participant k over the possible identities of the already vocalizing participant j. When these features are used alone, they are outperformed by the two-participant features. This suggests that average overlap behavior does not distinguish between the three meeting types as well as does the overlap interaction between participants of specific vocalizing time rank.

Columns 3 and 5 of Table 2 show the performance of the same 6 feature types, in combination with the f_k^T features. Due to space constraints, we mention only that most feature types appear to combine additively with f_k^T. We also show, in the last two lines

Estimated	Actual Type		
	Bed	Bmr	Bro
Bed	11	1	3
Bmr	2	26	1
Bro	3	1	19

Table 3: Confusion matrix among the three ICSI meeting types studied, for classification with NN-estimated "opt" feature set (f_k^{VI}, $f_{k,j}^{OI}$, and $f_{k,j}^{OC}$).

of the table, the performance of all feature types together, as well as of an "oracle" feature set derived using backward feature selection, by removing the worst performing feature one at a time from the "all" feature set. The best number achieved, 83.6%, was obtained using total vocalizing proportion f_k^T, NN-estimated single-participant f_k^{VI}, and NN-estimated two-participant features $f_{k,j}^{OI}$ and $f_{k,j}^{O}C$, which describe the overlap behavior of specific participant ranks with respect to specific other participant ranks. The accuracy represents a 52% relative reduction over the baseline (from 34.3% to 16.4%).

We show the confusion matrix of the "opt" NN-estimated feature set in Table 3. Although the amount of data is too small to draw statistically meaningful conclusions, the symmetrical misclassification of 3 Bro meetings as type Bed and 3 Bed meetings as type Bro suggests that in fact the Bro and Bed meeting types are more similar to each other than either is to the Bmr meeting type.

6 Conclusions

We have proposed a framework for the classification of conversational style in multi-participant conversation. The framework makes use of several novel elements. First, it relies exclusively on text-independent features, extracted from the multiparticipant vocal interaction patterns of a conversation; the technique is directly deployable for languages for which mature automatic speech recognition or dialog act classification infrastructure may be lacking. Second, we have made use of a well-studied model in stochastic dynamics, the Ising model, to improve estimates of the transition probabilities that describe the evolution of multiparticipant vocal interaction over the course of conversation. Third, we have introduced the concept of enumerable groups of participants, making it possible to include features which model the interaction between specific pairs of participants, for meetings with any number of participants. Finally, we have applied the framework to the task of classifying meeting types. Our experiments show that features describing the text-independent interaction between participants of specific vocalizing time rank, when used in conjunction with a feature which performs poorly on its own f_k^{VI}, lead to a relative error reduction of 52% over our baseline.

The key findings from the analysis of different interaction features are that having detailed 2-participant features is better than simply using the average for a given target speaker, and that using interaction features (conversation dynamics) gives better results than the static measure of relative speaking time. Of course, the best results are achieved with a combination of these types of features.

7 Future Work

In the future, we will apply the proposed classification system to automatically generated multichannel segmentation and alternatives to the Gaussian classifier. It may also be interesting to investigate separately representing different types of vocalization (e.g. speech vs. laughter) and features related to overlaps of more than two speakers.

For resource rich languages, meeting type can be classified using lexical features from speech recognition. However, if one is interested in detecting meeting type independent of content, the choice of word features needs to factor out topic. It would be interesting to assess the relative importance of words vs. interactions, and the degree to which they are complementary, in the topic-independent context.

Finally, another important future direction is the application of the techniques to the dual of Equation 5,

$$\begin{aligned} \mathcal{G}^* &= \arg\max_{\mathcal{G}} P(\mathcal{G} \mid \mathbf{F}) \\ &= \arg\max_{\mathcal{G}} \sum_{\mathcal{T}} P(\mathcal{G}, \mathcal{T}, \mathbf{F}) \\ &= \arg\max_{\mathcal{G}} \sum_{\mathcal{T}} P(\mathcal{T}) \times \\ &\qquad P(\mathcal{G} \mid \mathcal{T}) \times P(\mathbf{F} \mid \mathcal{G}, \mathcal{T}) \end{aligned} \tag{18}$$

namely the problem of jointly characterizing participants rather than conversations.

8 Acknowledgments

This work was partly supported by the European Union under the integrated project CHIL (IST-506909), Computers in the Human Interaction Loop, while Mari Ostendorf was a Visiting Professor at the University of Karlsruhe.

References

S. Banerjee and A. Rudnicky. 2004. Using simple speech based features to detect the state of a meeting and the roles of the meeting participants. *Proceedings of INTERSPEECH*, Jeju Island, South Korea.

S. Basu 2002. Conversational Scene Analysis. doctoral thesis, MIT.

O. Brdiczka and J. Maisonnasse and P. Reignier. 2005. Automatic detection of interaction groups. *Proceedings of ICMI*, Trento, Italy.

S. Burger and V. MacLaren and H. Yu. 2002 The ISL Meeting Corpus: The impact of meeting type on speech style. *Proceedings of ICSLP*, Denver CO, USA, pp301–304.

J. Dabbs and R. Ruback. 1987. Dimensions of group process: Amount and structure of vocal interaction. *Advances in Experimental Psychology*, 20, pp123–169.

N. Fay and S. Garrod and J. Carletta. 2000. Group discussion as interactive dialogue or as serial monologue: The influence of group size. *Psychological Science*, 11(6), pp487–492.

R. Glauber. 1963. Time-dependent statistics of the Ising model. *Journal of Mathematical Physics*, 4(2), pp294–307.

J. Hertz and A. Krogh and R. Palmer. 1991. *Introduction to the Theory of Neural Compuations.* Addison-Wesley Longman.

A. Janin and D. Baron and J. Edwards and D. Ellis and D. Gelbart and N. Morgan and B. Peskin and T. Pfau and E. Shriberg and A. Stolcke and C. Wooters. 2003. The ICSI Meeting Corpus. *Proceedings of ICASSP*, Hong Kong, China, pp364–367.

Q. Jin and K. Laskowski and T. Schultz. 2004 Speaker segmentation and clustering in meetings. *Proceedings of ICASSP NIST RT-04s Spring Meeting Recognition Evaluation Workshop*, Montreal, Canada.

K. Laskowski and T. Schultz. 2006. Unsupervised learning of overlapped speech model parameters for multichannel speech activity detection in meetings. *Proceedings of ICASSP*, Toulouse, France, pp993–996.

K. Laskowski and T. Schultz. 2007. Modeling vocal interaction for segmentation in meeting recognition. *Proceedings of MLMI (to appear)*, Brno, Czech Republic.

K. Laskowski and T. Schultz. 2007. Analysis of the occurrence of laughter in meetings. *Proceedings of INTERSPEECH (to appear)*, Antwerpen, Belgium.

I. McCowan and S. Bengio and D. Gatica-Perez and G. Lathout and M. Barnard and D. Zhang. 2005. Automatic analysis of multimodal group actions in meetings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3), pp305–317.

N. Mirghafori and C. Wooters. 2006. Nuts and Flakes: A study of data characteristics in speaker diarization. *Proceedings ICASSP*, Toulouse, France, pp1017–1020.

S. Renals and D. Ellis. 2003. Audio information access from meeting rooms. *Proceedings ICASSP*, Hong Kong, China, pp744–747.

R. Rienks and D. Heylen. 2005. Dominance detection in meetings using easily obtainable features. *Proceedings MLMI*, Edinburgh, UK.

H. Sacks and E. Schegloff and G. Jefferson. 1974. A simplest semantics for the organization of turn-taking for conversation. *Language*, 50(4), pp696–735.

E. Shriberg and R. Dhillon and S. Bhagat and J. Ang and H. Carvey. 2004. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. *Proceedings SIGdial*, Cambridge MA, USA, pp97–100.

D. Wyatt and J. Bilmes and T. Choudhury and H. Kautz. 2007. A privacy-sensitive approach to modeling multi-person conversations. *Proceedings IJCAI*, Hyderabad, India, pp1769–1775.

M. Zancanaro and B. Lepri and F. Pianesi. 2006. Automatic detection of group functional roles in face to face interactions. *Proceedings ICMI*, Banff, Canada.

Introducing Utterance Verification in Spoken Dialogue System to Improve Dynamic Help Generation for Novice Users

Kazunori Komatani Yuichiro Fukubayashi Tetsuya Ogata Hiroshi G. Okuno

Graduate School of Informatics
Kyoto University
Yoshida-Hommachi, Sakyo, Kyoto 606-8501, Japan
{komatani,fukubaya,ogata,okuno}@kuis.kyoto-u.ac.jp

Abstract

A method is presented that helps novice users understand the language expressions that a system can accept, even from unacceptable utterances made that may contain automatic speech recognition errors. We have developed a method that dynamically generates help messages, which can avoid further unacceptable utterances from being made, by estimating a users' knowledge from their utterances. To improve the accuracy of the estimation, we developed a method to estimate a user's knowledge from utterance verification results. This method estimates whether a user knows an utterance pattern that the system considers acceptable, and suppresses useless help messages from being generated.

1 Introduction

We have developed a user friendly spoken dialogue system, even for novice users, that generates help messages dynamically (Fukubayashi et al., 2006). Since novice users do not necessarily know the language expressions that can be accepted by a system, help messages need to be generated to instruct them of acceptable expressions. Such messages can be generated by estimating each user's knowledge of the system through their interactions with the system.

Users often make out-of-vocabulary or out-of-grammar utterances. This is unavoidable because of the characteristics of speech, that is, speech interfaces do not provide enough affordance (Norman, 1988). A graphical user interface (GUI) provides users with a clear representation of the kind of input required by the system; however, users have difficulty in understanding the input required when speech interfaces are used. Unfortunately, the range of language expressions a spoken dialogue system can handle is inherently limited. Even when a statistical language model is used in automatic speech recognition (ASR) and large numbers of expressions can be handled, patterns of language expressions are limited in language understanding (LU) or dialogue management (DM) components. This problem is compounded when novice users do not know what utterances can be accepted by a system. This is the very situation in which help messages should be generated, but ASR results for this type of utterance are unreliable because the utterances are often considered unacceptable. Even from such erroneous ASR results, systems have to estimate a user's knowledge accurately.

We addressed this problem by introducing an utterance verification technique. Since utterance verification does not use ASR results but uses acoustic scores of ASR, information about a user's utterances can be obtained, even from utterances that are considered unacceptable. By using its result, we can measure how close an utterance is to the grammar of a system.

Several studies have focused on generating help messages (Gorrell et al., 2002) (Hockey et al., 2003). Since they did not consider changes in the user's knowledge during the dialogue, the same help messages were generated when the same speech recognition results were obtained. Furthermore, these

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 202–205,
Antwerp, September 2007.

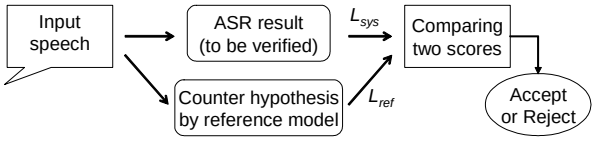


Figure 1: Overview of utterance verification

studies used ASR results from a "secondary" statistical language model when the primary grammar-based ASR failed. We ensured a user's knowledge state was updated appropriately, even when their utterances did not perfectly match utterances expected by a system developer, by detecting them as out-of-grammar utterances.

2 Generating Dynamic Help Messages Using Utterance Verification

2.1 Utterance Verification Using Differences in Acoustic Likelihoods

Utterance verification is generally performed by comparing log-scaled scores between an ASR output to be verified and a counter hypothesis based on a reference model. Same acoustic models were used in both recognizers. An outline of this process is shown in Figure 1. We denote the acoustic likelihood of the reference recognizer as $L_{\rm ref}$, the acoustic likelihood of the target-domain recognizer as $L_{\rm sys}$, the duration of the utterance as T (sec.), and the threshold as θ_{score}. The verification is assessed by using the following equation:

$$\begin{cases} S = (L_{\rm ref} - L_{\rm sys})/T & < \quad \theta_{score} \quad (Accept) \\ & \geq \quad \theta_{score} \quad (Reject) \end{cases} \tag{1}$$

The difference in the scores between the two recognizers indicates how close the user's utterance is to the system's grammar, which provides different information from conventional confidence measures (CMs) that are calculated for each word (Komatani and Kawahara, 2000).

Various studies have investigated the different reference models used in utterance verification (Sukkar et al., 1995; Kawahara et al., 1998). We used a simple utterance verification method in which the difference between log-scaled acoustic scores of the two recognizers is calculated. This is because we are now focusing on how utterance verification results can be used in spoken dialogue systems. The utterance verification method itself can be replaced if more accurate methods become available.

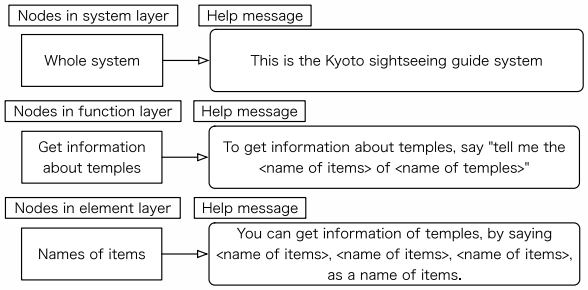


Figure 3: Example of help messages for each node

2.2 Generating Dynamic Help Messages

We have developed a method to generate help messages that fills the gap between a user's knowledge and the actual structure considered acceptable by the system. A detailed explanation of an algorithm we developed has been presented in our previous paper (Fukubayashi et al., 2006). The following is a concise explanation of that algorithm.

A *domain concept tree* was designed to represent a concept structure of a system, which represents the hierarchical layers of the target domain. This tree consists of four layers: "system", "function", "element", and "content word". The domain concept tree of the Kyoto sightseeing guide system is shown in Figure 2 as an example. *Known degrees* of each node in the domain concept tree are estimated. The degree represents how well a user knows a concept corresponding to the node. Known degrees are updated after each user's utterance; for example, a known degree of a node in the content word layer is increased if the content word is contained in an ASR result, and the effect is propagated to the known degrees of its ancestors. Lastly, a help message is generated after searching for a node having the lowest known degree. The message is generated by using templates, as shown in Figure 3.

The domain concept tree was updated on the basis of the ASR results from the user's utterances and the generated help messages. A user's knowledge, however, must be updated correctly, even when the content words in the user's utterances contain ASR

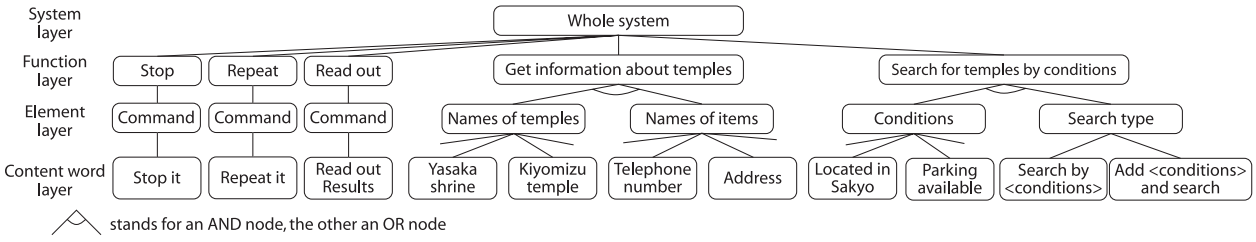


Figure 2: Domain concept tree of Kyoto sightseeing guide system

errors. For example, if a user says "Please tell me *an approach* to Yoshida Shrine." Even if *an approach* is unknown to the system, the system should be able to estimate that this user knows the utterance pattern considered acceptable but that they do not know the content words considered acceptable by the system.

2.3 Updating Domain Concept Tree Using Utterance Verification Results

Two loss functions are defined as:

$$cost_1 = (FA + SErr)/2$$
$$cost_2 = (FA + SErr + (1 - Acc))/3$$

Ratio FA (false acceptance) is the ratio of incorrectly accepted utterances that should be rejected, and $SErr$ (slot error) is the ratio of correct utterances that are not accepted (Komatani and Kawahara, 2000). We also calculated the accuracy of the language understanding by using the following equation: $Acc = (N - D - S - I)/N$, where N is the number of correct content words, and D, S and I are the numbers of deletion, substitution, and insertion errors, respectively. The accuracy of utterance verification is represented as $cost_1$, and $cost_2$ takes the accuracy of the language understanding results into consideration.

We defined two thresholds, θ_1 and θ_2, to minimize $cost_1$ and $cost_2$, respectively. Therefore, thresholds θ_1 and θ_2 focus on whether a whole utterance is in-grammar and whether content words in an utterance are correct. As a result, user utterances can be classified into one of the following three categories:

1. $S < \theta_1$: in-grammar and correct language understanding result,
2. $\theta_1 \leq S < \theta_2$: in-grammar but incorrect language understanding result, and
3. $S \geq \theta_2$: out-of-grammar and incorrect language understanding result.

The known degrees can be updated based on the above classification. When $S < \theta_1$, known degrees are ordinarily updated on the basis of the content words in the ASR results. Utterances whose S is greater than θ_2 are normally rejected. When $\theta_1 \leq S < \theta_2$, this utterance is estimated to be an in-grammar utterance, but its language understanding result seems to be incorrect. That is, the utterance seems to match to the system's grammar, even though it may contain incorrect content words. Then, the known degrees of the nodes in the function layer increase. This update allows the system to acquire information as to the user's knowledge regarding the system's grammar for the domain concept tree, even for utterances whose content words are not correctly recognized, and consequently suppresses unnecessary help messages from being generated regarding grammars.

3 Experimental Evaluation

We used dialogue data collected from users when they operated the Kyoto sightseeing guide system (Fukubayashi et al., 2006) in our evaluation. The dialogue data consists of 1,518 utterances from 12 subjects, none of which had previously used the system. Therefore, many user utterances were outside the range considered acceptable by the system and caused many ASR errors.

We used a grammar-based ASR engine, Julian[1], that has a vocabulary of 673 words. The average accuracy of the ASR was 42.9%. As a reference model for utterance verification, we used the outputs from a speech recognizer, Julius, which is based on statistical language models. Its language model was trained using newspaper articles and has a vocabulary of 20,000 words. The same acoustic model was

[1] http://julius.sourceforge.jp/

Table 1: Classification of utterances when setting θ_1 to 75 and θ_2 to 125

Correct answer of UV	LU results	$S < \theta_1$	$\theta_1 \leq S < \theta_2$	$S \geq \theta_2$
Accept	Correct (100%)	$454^{\dagger}$	$50^{(\star)}$	8
Accept	Some errors (<100%)	84	$29^{(\star\star)}$	34
Accept	No output	28	$13^{(\star\star\star)}$	23
Reject	Some errors (insertion error)	158	104	$185^{\ddagger}$
Reject	No output (correct rejection)	166	86	98

UV: utterance verification, LU: language understanding

used in both recognizers.

Loss functions $cost_1$ and $cost_2$ were minimized when θ_1 was 125 and θ_2 was 75. We counted the number of utterances in each category. The results are listed in Table 1. We compared the results from Table 1 with the results when the utterances with $S \geq \theta_2$ were simply rejected in which the performance was optimized by considering both the utterance verification and language understanding results. A value denoted by $^{(\star\star)}$ in Table 1 represents utterances that were incorrectly accepted despite some errors being contained in their language understanding results, and a value denoted by $^{(\star\star\star)}$ represents utterances that were rejected because no language understanding result was obtained. Therefore, the system obtained new information indicating that a user knows about the expression considered acceptable by the system, from 42 utterances which are denoted by $^{(\star\star)}$ and $^{(\star\star\star)}$. This enables the system to correctly update the domain concept tree at the function layer, even when correct language understanding results are not obtained.

In this case, correct language understanding results will be incorrectly rejected for 50 utterances denoted by $^{(\star)}$. Therefore, the performance needs to be improved. One reason for the inadequate performance is that the utterance verification algorithm used was very simple: only the differences in acoustic scores between the two recognizers were used. The performance of the classification is currently being improved, especially for short utterances whose differences in acoustic scores were not large enough, by considering other features.

4 Conclusion

We developed a method to update a user's knowledge that uses results from utterance verification, even when correct ASR results are not obtained. By using the utterance verification results, the system can estimate whether a user knows utterance patterns and can increase known degrees in the domain concept tree accordingly, which results in suppressing help messages from being generated regarding utterance patterns. Our future work includes improving the classification accuracy and using actual dialogues in experimental evaluations.

References

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. INTERSPEECH*.

Genevieve Gorrell, Ian Lewin, and Manny Rayner. 2002. Adding intelligent help to mixed-initiative spoken dialogue systems. In *Proc. ICSLP*.

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. EACL*.

Tatsuya Kawahara, Kentaro Ishizuka, Shuji Doshita, and Chin-Hui Lee. 1998. Speaking-style dependent filler phrase model for key-phrase detection and verification. In *Proc. ICSLP*, pages 3253–3256.

Kazunori Komatani and Tatsuya Kawahara. 2000. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proc. COLING*, pages 467–473.

Donald A. Norman. 1988. *The Psychology of Everyday Things*. Basic Books.

Rafid A. Sukkar, Anand R. Sethur, Mazin G. Rahim, and Chin-Hui Lee. 1995. Utterance verification of keyword strings using word-based minimum verification error (WB-MVE) training. In *Proc. IEEE-ICASSP*.

Making grounding decisions: Data-driven estimation of dialogue costs and confidence thresholds

Gabriel Skantze
Centre for Speech Technology, KTH
gabriel@speech.kth.se

Abstract

This paper presents a data-driven decision-theoretic approach to making grounding decisions in spoken dialogue systems, i.e., to decide which recognition hypotheses to consider as correct and which grounding action to take. Based on task analysis of the dialogue domain, cost functions are derived, which take dialogue efficiency, consequence of task failure and information gain into account. Dialogue data is then used to estimate speech recognition confidence thresholds that are dependent on the dialogue context.

1 Making grounding decisions

One of the greatest challenges when building spoken dialogue systems is to deal with *uncertainty*, which stems partly from the ambiguity of language itself and partly from the error-prone speech recognition process. Speakers handle uncertainty in dialogue by exchanging positive and negative evidence of understanding, a process commonly referred to as *grounding* (Clark, 1996). In order for a speaker to evaluate the addressee's understanding of the concepts involved, two kinds of evidence are especially useful: display of understanding (DISPLAY) and clarification requests (CLARIFY). The addressee can also choose not to provide any direct evidence, but to simply ACCEPT or REJECT the hypothesis. These options are exemplified in the following alternative system responses in a human-computer dialogue, where the system is uncertain about *the red building*:

(1) U: I can see a red building.
S (ACCEPT): *Ok, can you see a tree in front of you?*
S (DISPLAY): *Ok, a red building, can you see a tree in front of you?*
S (CLARIFY): *A red building?*
S (REJECT): *What did you say?*

In many dialogue systems, CLARIFY and DISPLAY are referred to as *explicit* and *implicit* verification.

The problem addressed in this paper can be described as follows: Given these different grounding options, how should a dialogue system choose what kind of evidence to give and which hypotheses to accept and reject? We will refer to this as the *grounding decision problem*. There are at least three important factors that speakers may take into account when making this decision:

1. Level of uncertainty
2. Task-related costs and utility
3. Cost of grounding actions

First, the more uncertain listeners are, the more evidence they provide. Second, as less evidence is given, the risk that a misunderstanding occurs will increase – thereby jeopardizing the task the speakers may be involved in. However, the cost of such a misunderstanding depends on the task at hand. Third, it would not be efficient to always display understanding or clarify everything that is said. Sometimes it may be more efficient to risk a misunderstanding and take the consequences.

A common approach to grounding decisions is to compare the speech recognition confidence score against a set of hand-crafted thresholds, and choose ACCEPT when the confidence is high, DISPLAY for middle-high scores, CLARIFY for middle-low scores and REJECT for low scores (see for example Bouwman et al., 1999). However, in this simple account, only Factor 1 above (level of uncertainty) is considered, and the thresholds used are typically only based on intuition.

In order to take Factor 2 (task-related costs and utility) into account, Bohus & Rudnicky (2001) uses a data-driven technique to derive actual costs from dialogue data, which showed that false acceptances were more costly than false rejections. Another aspect is that task costs are dynamic and often depend on the current state of the dialogue. To incorporate this aspect, Bohus & Rudnicky (2005) presents a method where binary logistic regression is used to determine the costs (in terms of task success) of various types of understanding errors involved in the rejection trade-off. Different regressions may then be calculated in different dialogue states, resulting in dynamic thresholds. However, these methods do not consider other grounding actions than ACCEPT

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 206–210,
Antwerp, September 2007.

and REJECT. To do this, Factor 3 above (cost of grounding actions) must also be considered.

Paek & Horvitz (2003) presents a decision theoretic approach to the grounding decision problem, based on the framework of *decision making under uncertainty*. According to this proposal, the optimal grounding action *GA* should satisfy the Principle of Maximum Expected Utility (MEU), which can be defined as follows: *Choose an action a, so that the expected utility EU(a) is maximized*. When making this decision, the world may be in one of the states h_1, h_2, $h_3 \ldots h_n$, and this state may have an impact on the effect of the action taken. This effect can be described by the function $Utility(a,h_i)$, which is the utility for action a under state h_i. Thus, for each action a, the probability for each possible state and the utility for taking action a, given that state, should be summed up:

$$(2) \quad GA = \arg\max_a EU(a) = \arg\max_a \sum_{i=1}^{n} P(h_i) \times Utility(a, h_i)$$

This approach is promising, in that it may account for all decision factors listed above. However, in Paek & Horvitz (2003), the utilities used in the model were estimated directly by the user (via a GUI) and were not derived from data.

2 The proposed model

In this paper, we will show how the utilities may be estimated directly from collected dialogue data. To do this, the problem will be described as that of minimising costs: *Choose a grounding action a, so that the sum of all task-related costs and grounding costs is minimised, considering the probability that the recognition hypothesis is correct.* Thus, the world may be in two states (*correct* and *incorrect* recognition), and a probability measure for these states is needed, as well as a cost function for calculating the costs of the different grounding actions, given these states. The problem is expressed in the following equation (where *P(incorrect)* equals 1-*P(correct)*):

$$(3) \quad GA = \arg\min_a \begin{pmatrix} P(correct) \times Cost(a, correct) + \\ P(incorrect) \times Cost(a, incorrect) \end{pmatrix}$$

To select the optimal grounding action according to equation (3), a probability measure of the state *correct* is needed, as well as a cost function for calculating the costs of the different grounding actions, given these states.

In this paper, we will assume that *P(correct)* can be derived from the speech recognition confidence score. While confidence scores typically delivered by speech recognisers should not be used as a direct measure of probability, it should be possible to derive probabilistic scores (Jiang, 2005).

3 Data

The model presented in this paper will be applied to data collected using the HIGGINS spoken dialogue system developed at KTH (Edlund et al., 2004). The initial domain for the system developed within the project is pedestrian city navigation and guiding. A user gives the system a destination and the system guides the user by giving verbal instructions. The system does not have access to the user's position. Instead, it has to figure out the position based on the user's descriptions of the surroundings. Since the user is moving, the system continually has to update its model of the user's position and provide new, possibly amended instructions until the destination is reached. For simulation, a 3D model of a virtual city is used. Example (1) above is typical for this domain. A typical dialogue consists of three main *phases* or *sub-tasks*: a *goal assertion phase*, a *positioning phase*, and a *guiding phase*.

A version of the HIGGINS system, with different sets of handcrafted confidence thresholds for making grounding decisions, was evaluated with users. The evaluation involved 16 participants, all native speakers of Swedish. The collected data consists of 2007 user utterances. A more detailed description of the data collection is provided in Skantze (in press).

4 Cost measure and functions

The model presented in this paper relies on a unified cost measure, which may be used for estimating both the task-related costs and the cost of grounding actions. The ultimate measure of cost would be the reduction of user satisfaction. However, user satisfaction is practically only obtainable on the dialogue level, and we need a much more detailed analysis. A cost measure that is relevant for both grounding actions and the task, and that is obtainable on all levels of analysis, is *efficiency*. This is reflected in the *principle of least effort* (Clark, 1996): "All things being equal, agents try to minimize their effort in doing what they intend to do". Thus, efficiency and user satisfaction should correlate to some degree, at least in a task-oriented dialogue setting as the one used in this paper. In the data collected here, the best predictor for user satisfaction was the *total number of syllables* uttered (from both the user and the system) ($R^2 = 0.622$). The impact of efficiency on user satisfaction in task-oriented dialogue has also been reported in other studies, such as Bouwman & Hulstijn (1998).

Using efficiency as a cost measure, we will analyse the consequences of different actions, given the correctness of the recognition hypothesis. The actions that will be considered are the ones listed in example (1): ACCEPT, DISPLAY, CLARIFY and REJECT. Table 1 summarises these costs based on a set of parameters, which are all average estimations over a set of dialogues.

Table 1: Costs for different grounding actions, given the correctness of the recognition (COR=Correct, INC=Incorrect).

Action,Hyp	Costs
ACCEPT,COR	No cost
ACCEPT,INC	The number of extra syllables the misunderstanding adds to the dialogue (*SylMis*).
DISPLAY,COR	Grounding dialogue *(SylDispCor)*.
DISPLAY,INC	Grounding dialogue (*SylDispInc)*. Risk that the user does not correct the system *(P(Fail|Disp,Inc))* times the consequences of a misunderstanding (*SylMis*).
CLARIFY,COR	Grounding dialogue *(SylClarCor)*. Risk that the user does not confirm the system *(P(Fail|Clar,Cor))* times the syllables for recovering the rejected concept *(SylRec)*.
CLARIFY,INC	Grounding dialogue *(SylClarInc)*
REJECT,COR	The number of syllables it takes to receive new information of the same value as the rejected concept *(SylRec)*.
REJECT,INC	No cost

The costs for DISPLAY and CLARIFY may need some explanation. In HIGGINS, a concept that is displayed is treated as correct unless the user initiates a repair. A concept that is clarified is treated as incorrect unless the user confirms it. Thus, they can be said to *fail* if the user does not correct a displayed misunderstanding or confirm a clarification of a correct concept. The number of syllables an average grounding dialogue takes involves both the grounding act and possible responses. For example, the following clarification dialogue involves 2 syllables (*SylClarCor*):

(4) S: *Red?*
U: Yes

Using these costs and equation (3) above, cost function may be defined for the different actions, as shown in Table 2.

Table 2: Cost functions for different grounding actions.

Action	Expected cost
ACCEPT	*P(incorrect)* x *SylMis*
DISPLAY	*P(correct)* x *SylDispCor* + *P(incorrect)* x *(SylDispInc* + *P(Fail|Disp,Inc)* x *SylMis)*
CLARIFY	*P(correct)* x *(SylClarCor* + *P(Fail|Clar,Cor)* x *SylRec)* + *P(incorrect)* x *SylClarInc*
REJECT	*P(correct)* x *SylRec*

5 Parameter estimation from data

To show how these parameters may be estimated from data, we will make a task analysis specific for the navigation domain presented here. We will start with the positioning phase of the dialogue, i.e., when the user describes her position, as in example (1) above.

The parameter *SylRec* describes the number of syllables it will take to get the same amount of information after a concept has been rejected. This parameter is highly context dependent – it depends on how much information the hypothesised concept provides (its *information gain*), compared to the average concept. This proportion will be referred to as *ConValueH*. The system and the user spent on average 15.0 syllables per important concept [1] accepted by the system. We will refer to this as *SylCon*. Based on these two parameters, *SylRec* can be calculated as follows:

(5) $SylRec = SylCon \times ConValueH$

How can *ConValueH* be estimated for the positioning phase? The purpose of the positioning phase is to cut down the number of possible user locations. Thus, the value of a concept can be described as the proportion of the set of possible user locations that are cut down after accepting it, compared to the average concept. The proportion of possible locations that are reduced on average after a single concept is accepted can be estimated from data (*CutDownA*). The dialogue system can then use the domain database to calculate the proportion of possible locations that would be cut down if the hypothesised concept would be accepted (*CutDownH*). By accepting *ConValueH* number of average concepts, each leaving a proportion of 1 - *CutDownA* possible locations, a proportion of 1 - *CutDownH* locations should be left. This is expressed in the following formula:

(6) $(1 - CutDownA)^{ConValueH} = (1 - CutDownH)$

By combining equations (5) and (6), *SylRec* can be calculated with the following formula:

(7) $$SylRec = SylCon \times \frac{\log(1 - CutDownH)}{\log(1 - CutDownA)}$$

We will now turn to the parameter *SylMis,* which describes the number of extra syllables a misunderstanding adds to the dialogue. The risk of accepting an incorrect concept during the positioning phase is that the set of possible user positions may be erroneously constrained. If this happens, the positioning often has to start all over again. Thus, *SylMis* should reflect the number of syllables a complete positioning takes (on average 97.0, which we will refer to as *SylPos*). However, the set of possible user locations does not *need* to be erroneously constrained when accepting an incorrect concept (the user may actually see a red building, even if this was not what she said). The probability that the correct position actually is lost can be described by the parameter *CutDownH* defined above, i.e., the proportion of possible locations that is reduced if the hypothesised concept is accepted. Thus *SylMis* can be calculated as follows:

[1] By important concept, we mean concepts that contribute in the current task. In this example, RED is important, but not BUILDING, since there are buildings everywhere.

(8) *SylMis* = *SylPos* x *CutDownH*

The rest of the parameters can be calculated from the data by counting the number of syllables spent on the grounding sub-dialogues and the number of times they failed. These parameters are shown in Table 3. *SylGA* is the number of syllables involved in the grounding act (in the case of DISPLAY or CLARIFY).

Table 3: Estimation of parameters.

Parameter	Value
SylClarCor	*SylGA* + 1.4
SylClarInc	*SylGA* + 2.1
SylDispCor	*SylGA* + 0.1
SylDispInc	*SylGA* + 1.2
P(Fail|Clar,Cor)	0.33
P(Fail|Disp,Inc)	0.82

The high value of *P(Fail|Clar,Cor)*, and especially *P(Fail|Disp,Inc)*, might be explained by the fact that the system did not use an elaborate prosodic model for the realisation of fragmentary DISPLAY and CLARIFY acts. Also, the use of such fragments is still very uncommon in dialogue systems, which often resulted in that the users did not recognise their function.

We will now consider two examples where the concept information gain differs a lot (the concepts under question are underlined):

(9) I can see a <u>mailbox</u> (*CutDownH* = 0.782; *SylGA* = 2)

(10) I can see a <u>two</u> storey building (*CutDownH* = 0.118; *SylGA* = 1)

CutDownA can be estimated from data as 0.336. Using these parameters, the cost function for the different grounding actions, depending on *P(correct)*, can be calculated to find out which action has the least cost for each value of *P(correct)* and thus derive confidence thresholds, as shown in Figure 1 and Figure 2. As can be seen in these figures, example (9) has a much higher information gain and thus a wide confidence interval where a clarification request is optimal, whereas example (10) has less information gain and is optimally either accepted or rejected, but never clarified.

In the previous examples, we have only considered the positioning phase of the dialogues. However, there is another important phase, the goal assertion phase:

(11) U: I want to go to an <u>ATM</u> (*SylGA*=3)

If this hypothesis would constitute a misunderstanding, it would lead to much higher costs than a misunderstood positioning statement. In this case, we can define *SylMis* as the number of syllables it takes on average until the user has reached the (incorrect) goal or restated the goal, which can be estimated to 261.6 from the data. We will assume that *SylRec* is equal to *SylCon* (15.0), and that the other parameters are the same as in the positioning phase. The cost functions and thresholds for grounding

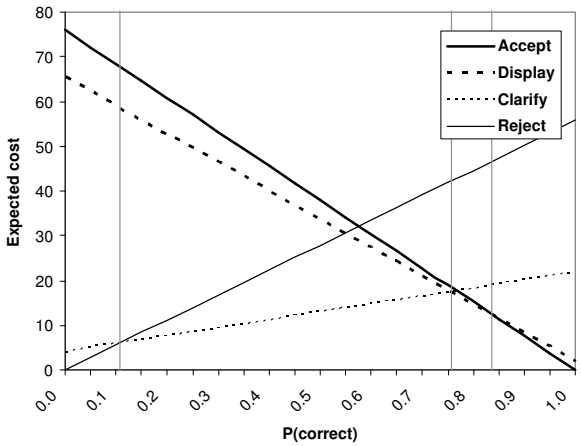


Figure 1: Cost functions and confidence thresholds for grounding the concept MAILBOX after "I can see a mailbox".

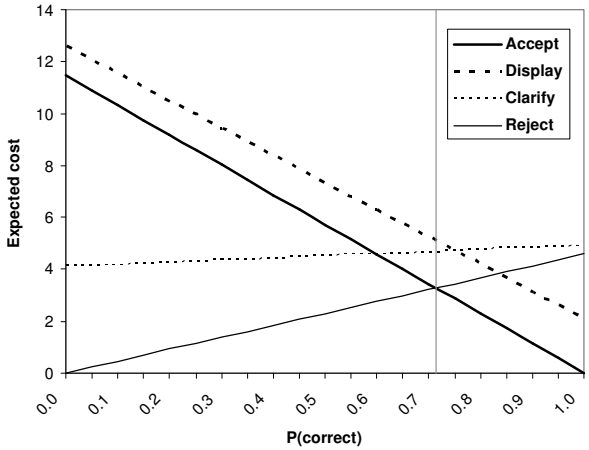


Figure 2: Cost functions and confidence thresholds for grounding the concept TWO after "I can see a two storey building".

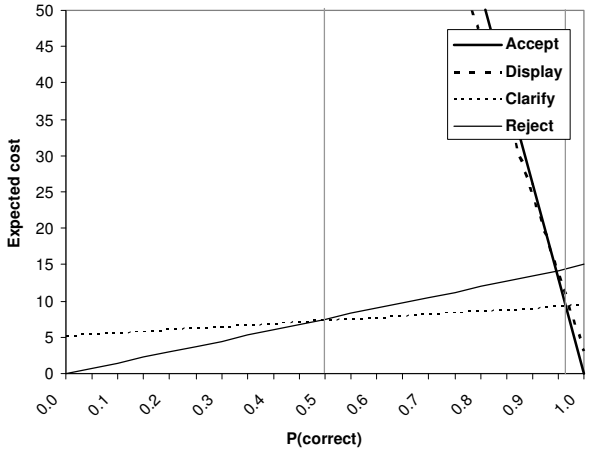


Figure 3: Cost functions and confidence thresholds for grounding the concept ATM after "I want to go to an ATM".

"ATM" in the example above are shown in Figure 3. Due to the high cost of misunderstandings, a simple accept requires a very high confidence, and goal assertions will therefore most often be clarified.

6 Discussion

The graphs presented above, and the calculation of thresholds, are of course only useful for illustrative purposes. A dialogue system would just calculate the most optimal action, given the value of *P(correct)*. It should be noted that these estimations are based on the data collected with hand-crafted confidence thresholds. If the derived thresholds would be applied to the system, the parameters values would change, thus affecting the thresholds. This means that the presented model should be derived iteratively, using bootstrapping, and the parameter values presented here are just the first step in such an iteration. To estimate the parameters, transcription of the dialogues and some annotation is needed. However, given that the logging is adapted for this, we believe that this can be done rather efficiently.

The functions presented in Table 2 describe general characteristics of the grounding actions and should be applicable to many different dialogue domains. However, the parameter estimation presented here is specific for the navigation domain. For some domains, it may be more problematic to use syllables as a general measure.

There are some simplifying assumptions in the model presented above. First, only one concept in the hypothesis is considered as correct or incorrect. It would of course also be possible to consider some concepts as correct and some concepts as incorrect. In such concept-level error handling (Skantze, in press), it is for example possible to clarify one concept while silently accepting or rejecting another. The model presented here could be extended to also cope with several concepts in an utterance with different probabilities, as in the following example (with probabilities in parenthesis):

(12) U: I can see a red building to the left
[RED (0.8) LEFT (0.2)]

In this case, we should consider 4 possible states instead of 2, 16 actions instead of 4, and 64 costs instead of 8. Here are some examples of the actions that should be considered:

Red? (CLARIFY RED, ACCEPT LEFT)
Do you have the red building on your left?
(DISPLAY RED, CLARIFY LEFT)
A red building on your left?
(CLARIFY RED, CLARIFY LEFT)

Another simplification is that temporal modelling of grounding (as discussed in Paek & Horvitz, 2003) is not considered, i.e., the fact that the utility of grounding actions change when they are repeated subsequently. However, it should be possible to account for this by conditioning the parameters, depending on the order in which the grounding action is taken. An elaborate model of *P(correct)* could also take this into account.

A more complex approach to grounding decisions is to use POMDP models (Williams & Young, 2007). The strength of such models is that they account for parallel recognition hypotheses and planning. The model presented here is much simpler and includes more bias. However, it requires less resources and is easier to apply and scale.

Of course, the presented model also remains to be evaluated, for example by comparing the performance of a system using this model with a system based on handcrafted thresholds.

References

Bohus, D., & Rudnicky, A. (2001). Modeling the cost of misunderstandings in the CMU Communicator dialog system. In *Proceedings of ASRU-2001*. Madonna di Campiglio, Italy.

Bohus, D., & Rudnicky, A. (2005). A principled approach for rejection threshold optimization in spoken dialog systems. In *Proceedings of Interspeech-2005*. Lisbon, Portugal.

Bouwman, G., & Hulstijn, J. (1998). Dialogue strategy redesign with reliability measures. In *Proceedings of the First International Conference on Language Resources and Evaluation*. Granada, Spain.

Bouwman, G., Sturm, J., & Boves, L. (1999). Incorporating confidence measures in the dutch train timetable information system developed in the Arise project. In *Proceedings of ICASSP'99*.

Clark, H. (1996). *Using language*. Cambridge University Press.

Edlund, J., Skantze, G., & Carlson, R. (2004). Higgins - a spoken dialogue system for investigating error handling techniques. In *Proceedings of ICSLP*. Jeju, Korea.

Jiang, H. (2005). Confidence measures for speech recognition: A survey. *Speech Communication*, 45(4).

Paek, T., & Horvitz, E. (2003). On the utility of decision-theoretic hidden subdialog. In *ISCA Workshop on Error Handling in Spoken Dialogue Systems*.

Skantze, G. (in press). Galatea: A discourse modeller supporting concept-level error handling in spoken dialogue systems. To be published in Dybkjær, L., & Minker, W. (Eds.), *Recent Trends in Discourse and Dialogue*. Springer.

Williams, J. D., & Young, S. (2007). Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2).

On the training data requirements for an automatic dialogue annotation technique*

Carlos D. Martínez-Hinarejos
Departamento de Sistemas Informáticos y Computación
Instituto Tecnológico de Informática
Universidad Politécnica de Valencia, Cno. de Vera s/n, Valencia, 46022
cmartine@dsic.upv.es

Abstract

When constructing a task-oriented dialogue system, it is usual to perform an acquisition of dialogues for the system's task. This acquisition can be used to define the behaviour of the dialogue system, and it can be rule-based or corpus-based. In the corpus-based case, the models that define the behaviour are automatically inferred from annotated dialogues. The annotation process is time-consuming and error-prone, and the use of assistant tools for the annotation can reduce the effort in this process. In this work, the data requirements of a previously presented annotation tool are presented, and the results show that the technique obtains its maximum performance even with a relative small amount of annotated dialogues.

1 Introduction

A dialogue system (Kuppevelt and Smith, 2003) is usually defined as an automatic system that interacts with a human user using dialogue, with the objective of solving a certain problem. Tasks such as timetable consultation (Aust et al., 1995) are common examples of dialogue system applications.

A dialogue system is defined by its behaviour, which tries to imitate a real dialogue situation. The most common method to define this behaviour is to acquire a corpus of dialogues on the task to be solved. In this acquisition, a set-up known as Wizard of Oz (Fraser and Gilbert, 1991) is used.

Then, the behaviour of the system is defined by analysing the acquired corpus of dialogues. Two main approximations have been used in the system's behaviour definition: rule-based (Gorin et al., 1997) and corpus-based (Stolcke et al., 2000). In the corpus-based approach the behaviour is determined by statistical models that are automatically inferred and updated. Therefore, in the corpus-based approach it is easier to adapt the system behaviour to new tasks and situations by inferring a new model.

The corpus-based approach needs huge amounts of data (dialogues) conveniently annotated to estimate the parameters of the statistical models. The most widely used annotation scheme is the Dialogue Act (DA) labelling (Searle, 1969). In this scheme, every turn of the dialogue is segmented into a utterance (Stolcke et al., 2000) and annotated with one DA, which defines its function in the dialogue.

The annotation of the corpus implies the definition of the set of DA and the annotation rules (Alcacer et al., 2005; Jurafsky et al., 1997), followed by the annotation itself, which is a very time-consuming process. Therefore, the development of automatic annotation techniques is very useful in the development of corpus-based dialogue systems.

Some automatic annotation techniques have been proposed in previous works (Stolcke et al., 2000). These techniques use part of the annotated dialogue corpus to infer the automatic annotators. These annotators are statistical models that, given the sequence of words, return the utterances with their corresponding DA labels. The automatic annotators are not error-free, and they improve their error rate as long as more training data is provided.

In this work, the influence of the amount of training dialogues on the automatic annotator error rate is presented. The results show that when using more than a certain number of dialogues, no significant improvements in the annotation error rate are no-

*Work partially supported by VIDI-UPV under PAID06-20070315 program.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 211–214,
Antwerp, September 2007.

ticed. This allows to determine the size of the corpus that must be manually annotated to obtain the highest automatic annotation performance with the lowest manual annotation cost.

The paper is organised as follows. In Section 2, the annotation technique is presented. In Section 3, the used dialogue corpora are described. In Section 4, the performed experiments and their results are discussed. In Section 5, conclusions and future work lines are presented.

2 GIATI based annotation technique

The automatic annotation technique which is analysed in this work is based on a general Stochastic Finite-State Transducer (SFST) inference technique known as GIATI (Casacuberta et al., 2005). This technique has been successfully used in Machine Translation tasks and in dialogue annotation (Martínez-Hinarejos, 2006).

GIATI infers a SFST from a parallel corpus using a re-labelling process of input-output pairs of sentences. From the re-labelled corpus, a smoothed n-gram is inferred and then it is converted into the final SFST by reverting the initial re-labelling. A modification of the GIATI annotation was proposed in (Martínez-Hinarejos, 2006) to perform the annotation directly using the n-gram instead of the SFST.

In dialogue is easy to find a re-labelling scheme because no cross-alignments are usually present (a DA label is attached to a complete utterance in a linear manner). For example, the DA label can be attached to all the words in the corresponding utterance, or only to the last word of the utterances. In this work, this last re-labelling strategy is used, following the steps presented in (Martínez-Hinarejos, 2006).

After the inference from the re-labelled corpus, the n-gram can be used as an annotator model. For the annotation, a Viterbi n-gram implementation was used following the ideas of (Martínez-Hinarejos, 2006). Intensive beam-search was applied in the implementation to avoid the problems with large exploration trees in the Viterbi process.

3 Dialogue corpora

In the experiments, two different dialogue corpora, with very different features, were used to assess the performance of the automatic annotation technique.

3.1 Dihana corpus

Dihana (Benedí et al., 2004) is a task-oriented corpus which is composed of computer-to-human dialogues. The main aim of the task is to answer telephone queries about timetables, fares, and services for long distance trains. The language of the corpus is Spanish.

The corpus is composed of 900 different dialogues that were acquired using the Wizard of Oz technique and semicontrolled scenarios. The total set of dialogues comprises 6,280 user turns and 9,133 system turns, with a vocabulary of 980 words. All the dialogues were annotated by human experts. The annotation scheme used in Dihana was presented in (Alcacer et al., 2005). The labels are organised in three different levels. The total number of labels which are present in the corpus is 248 (153 for user turns and 95 for system turns). If only the first and second level are taken into account, 72 different labels (45 for user and 27 for system) are present.

3.2 SwitchBoard corpus

SwitchBoard (Godfrey et al., 1992) is a well-known speech corpus which was obtained from human-to-human telephone conversations. These conversations were not task oriented, and both speakers were allowed to express themselves in a free manner and to interrupt the other speaker, discussing a general topic, but with no task to accomplish.

The corpus is composed of 1,155 conversations, with a total number of 126,754 different turns of spontaneous speech. The vocabulary size is 42,672 words. The corpus was annotated using a simplification of the DAMSL annotation scheme (Jurafsky et al., 1997) which comprises a total number of 42 different labels.

4 Experiments and results

The objective of the experiments is to determine which amount of labelled dialogues is enough to obtain the best possible GIATI-based dialogue labeller with the minimum annotation effort. It is clear that the quality of the labellers should be assessed with respect to a set of dialogues which is not included in the training corpus and that it must be fixed for all the variable-size training corpora. In our case, a set of 100 dialogues of the Dihana corpus and a set of 155 dialogues of the SwitchBoard corpus were taken

as the test corpora. The sizes of the training corpora were from 100 up to 800 in the Dihana corpus, and up to 1,000 in the case of the SwitchBoard corpus (with increments of 100 dialogues).

Some common preprocessing steps were performed in order to reduce data sparseness: case unification (all the words were transcribed in lowercase) and punctuation marks treatment (the punctuation marks were separated from the words).

For the Dihana corpus, two more preprocessing steps were applied: a categorisation (it was performed for categories such as town names, the time, dates, etc.) and the addition of a speaker identifier. These preprocessing steps reduced the vocabulary to 705 user and 190 system words. For this corpus, the annotation with only the two first levels was used.

In the case of the SwitchBoard corpus, one more preprocessing step was applied. It was utterance joining: the interrupted utterances (which were labelled with '+') were joined to the correct previous utterance. No categorisation was performed because of the no-task oriented nature of the SwitchBoard corpus. After these preprocessing steps, the vocabulary consisted of 21,797 different words, which reveals that the annotation of this corpus is more difficult because of the data sparseness.

For both different annotations (Dihana two-level and SwitchBoard), the set of incremental training corpora were defined. From both training corpora, three different GIATI-based models were trained: for 2, 3 and 4-grams. These automatic labellers were used to annotate the different test dialogue corpora (Dihana two-level and SwitchBoard). The automatic annotation was compared with the reference one with the Dialogue Act Error Rate (DAER) measure. DAER (which is similar to the Word Error Rate) computes which rate of the assigned labels are correct and do not have to be revised or corrected.

Absolute results on DAER for both corpora are presented in Figure 1. As it was expected, the results are worse as the complexity of the corpus increases: Dihana is the less complex, because of the reduced vocabulary and set of labels, and SwitchBoard is the most complex (with a large vocabulary). Another clear inference from the graphics is that the larger the training set size, the better the results.

This general tendency is quite more clear with the SwitchBoard corpus, and could be related to the decrement of out of vocabulary (OOV) words as the training corpus comes larger. In Dihana the OOV reduction rate is really small for a medium-size corpus, but in SwitchBoard this reduction is higher even for a large training corpus.

One more interesting observation is that there are no significant differences between the 3-gram and 4-gram results in the Dihana corpus, but the results with 4-grams with the SwitchBoard corpus are the worst of all, while there is no significant difference between the 2-gram and 3-gram results with this corpus. The explanation is that the high complexity of the SwitchBoard corpus makes association between words and DA too sparse to appropriately infer such a complex model as a 4-gram.

In order to assess the improvement as the corpus increases, the relative improvement of passing from one training corpus to the next one in the sequence was calculated. In both corpora, these results showed that using more than 300 dialogues for training did not provide any significant improvement (lower than 5%).

Some error analysis was performed on the results with 3-grams and 300 dialogues as training corpus. The analysis revealed that most of the errors in Dihana were substitutions between similar labels or labels which annotate similar sentences but with different dialogue meaning depending on the context. This indicates that the high locality of the models do not allow to distinguish between some situations (e.g., a question and an answer). Meanwhile, in the SwitchBoard corpus most errors involved the ambiguous *statement-opinion* (sv) and *statement-non-opinion* (sd) labels, which are difficult to determine even for human annotators (Stolcke et al., 2000).

With respect to the speed of the process, the annotation technique revealed itself as really fast. In the Dihana corpus, no more than 2 seconds per whole turn on average were needed. In the case of SwitchBoard, although is quite more complex, similar times were obtained.

5 Conclusions and future work

This work shows the behaviour of an automatic dialogue annotation technique, studying the effect of the amount of training data on the accuracy of the obtained models. The experiments were carried out with very different corpora, but the results show

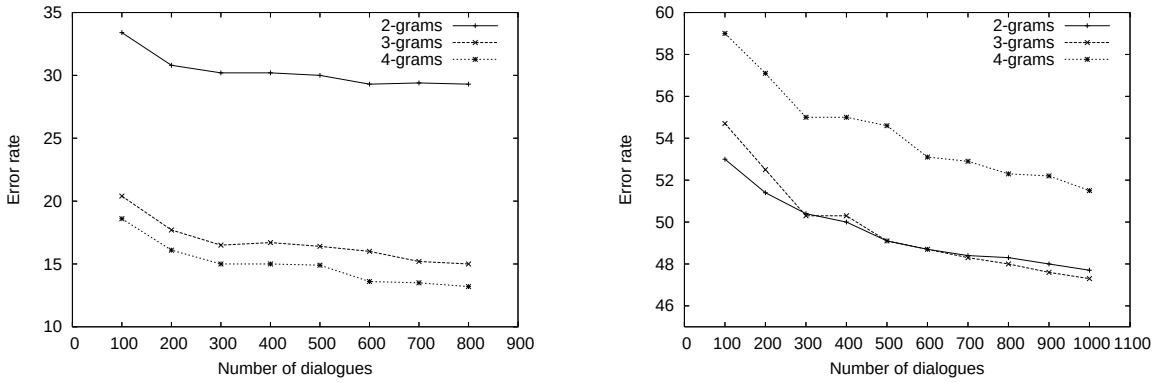


Figure 1: Absolute DAER rates for Dihana two-level and SwitchBoard.

the same behaviour: an amount of 300 dialogues is enough to obtain an appropriate annotation model. From this point, adding more dialogues to the training set does not improve significantly the accuracy of the models. Therefore, when applying this annotation technique in a dialogue corpus annotation, no new models should be inferred after the correct annotation of a relatively small number of dialogues (in this experiment, 300 dialogues). This speeds up the process, because the only task from this point is correcting the automatically annotated dialogues.

The results were obtained using the GIATI-based technique, but other annotation and identification techniques are available (Grau et al., 2004). Therefore, the same experimental framework should be applied on these techniques in order to know if they have the same limitations as the GIATI-based one. One interesting thing is the combination of several models for different tasks. Finally, although these conclusions were obtained from experiments with two corpora, experiments with more corpora could generalise these conclusions.

References

N. Alcacer, J. Benedí, F. Blat, R. Granell, C. D. Martínez, and F. Torres. 2005. Acquisition and labelling of a spontaneous speech dialogue corpus. In *Proceeding of 10th SPECOM*, pages 583–586, Patras, Greece.

H. Aust, M. Oerder, F. Seide, and V. Steinbiss. 1995. The philips automatic train timetable information system. *Speech Communication*, 17:249–263.

J. M. Benedí, A. Varona, and E. Lleida. 2004. Dihana: Dialogue system for information access using spontaneous speech in several environments tic2002-04103-c03. In *Reports for Jornadas de Seguimiento - PNTI*, Málaga, Spain.

F. Casacuberta, E. Vidal, and D. Picó. 2005. Inference of finite-state transducers from regular languages. *Pattern Recognition*, 38(9):1431–1443.

M. Fraser and G. Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5:81–99.

J. Godfrey, E. Holliman, and J. McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Proc. ICASSP-92*, pages 517–520.

A. Gorin, G. Riccardi, and J. Wright. 1997. How may i help you? *Speech Communication*, 23:113–127.

S. Grau, E. Sanchis, M. J. Castro, and D. Vilar. 2004. Dialogue act classification using a Bayesian approach. In *Proceedings of SPECOM'2004*, pages 495–499, Saint-Petersburg, Russia, September.

D. Jurafsky, E. Shriberg, and D. Biasca. 1997. Switchboard swbd-damsl shallow- discourse-function annotation coders manual. Technical Report 97-01, University of Colorado Institute of Cognitive Science.

J. Van Kuppevelt and R. W. Smith. 2003. *Current and New Directions in Discourse and Dialogue*, volume 22 of *Text, Speech and Language Technology*. Springer.

C.D. Martínez-Hinarejos. 2006. Automatic annotation of dialogues using n-grams. In *Proceedings of TSD 2006*, LNCS/LNAI 4188, pages 653–660, Brno, Czech Republic, Sep. Springer-Verlag.

J. R. Searle. 1969. *Speech acts*. Cambridge University Press.

A. Stolcke, N. Coccaro, R. Bates, P. Taylor, C. van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin, and M. Meteer. 2000. Dialogue act modelling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):1–34.

Practical dialogue manager development using POMDPs

Trung H. Bui, Boris van Schooten, and Dennis Hofs
University of Twente
PO Box 217 Enschede, The Netherlands
{buith, schooten, hofs}@ewi.utwente.nl

Abstract

Partially Observable Markov Decision Processes (POMDPs) are attractive for dialogue management because they are made to deal with noise and partial information. This paper addresses the problem of using them in a practical development cycle. We apply factored POMDP models to three applications. We examine our experiences with respect to design choices and issues, and compare performance with hand-crafted policies.

1 Introduction

Partially Observable Markov Decision Processes (POMDPs) are attractive for dialogue management in the cases where the dialogue manager has to make choices which depend on statistical information. They can determine optimal strategies in the face of error and partial information. POMDPs can take advantage of statistical information about behavior or error to the fullest extent, and take into account extensive hidden information.

Using POMDPs for spoken dialogue management has been examined thoroughly in (Williams and Young, 2007). Current POMDP-based dialogue managers model a complete slot-filling dialogue including all slots with all values. Large numbers of slots and values lead to a large state space, which is not tractable for current POMDP solvers. Usually, this restricts us to toy problems. Recent effort to scale up POMDP-based models is reported in (Williams and Young, 2007; Bui et al., 2007).

It is not yet clear enough how to employ POMDPs in a systematic development cycle. A number of practical issues with POMDPs has not really been addressed yet. How do you obtain the user model and the probability distributions? How do you test and debug POMDPs? How do you tweak reward values? How do you evaluate and compare performance of the POMDP policy which other approaches? We address these questions by using the factored POMDP models (Williams and Young, 2007; Bui et al., 2007) as a basis, and applying them to three dialogue management systems.

2 Methodology

Design guidelines. The state space represents the user's state and action. It is defined as a set of features. We should keep it compact. This can be done by specifying only features which are relevant in selecting the system action and by pruning all unreachable states. For example, when analyzing the Williams's 1945-state travel problem (Williams and Young, 2007), we found that could increase tractability by pruning 1626 states[1], leaving only 319 reachable states. The system actions are not only the actions toward the user but also actions for other dialogue manager tasks such as querying the database. Similar to the state space, the observation space is also defined as a set of observation features such as user's action with noise (from the ASR) and observed user's emotional state.

Designing a reward model that leads to a good policy is a very challenging task. The typical parameters used to design a reward model are task success, the number of turns, and dialogue act appropriateness (for example, the system should not confirm a value if it has not yet been provided by the user). The precise numerical values used may have significant impact on the policy and convergence behaviour.

Evaluation setup and toolset. From the literature, the typical approach is first to test the quality of the POMDP-based dialogue policy with a simulated user. The real-user evaluation is considered at the

[1] For example, the states which the user's goal feature is ab and the user's action feature is c

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 215–218,
Antwerp, September 2007.

final step. An advantage of modeling dialogue as a POMDP is that we can use the POMDP environment model itself as a simulated user model. The probability distributions of the simulated user (testing model) might be varied with the ones of the dialogue manager (training model). The probability distributions of all the user models used in our three applications are handcrafted. We have developed a software toolkit to conduct our experiments, which includes a factored POMDP to flat POMDP translator, and an interactive simulator for both the user and the system. The POMDP problem is first solved with a POMDP solver (we used Perseus (Spaan and Vlassis, 2005) and ZMDP (Smith and Simmons, 2005)). The generated alpha file is then used to carry out the performance test with simulated user models. Section 3 shows our test results on three different problems. We conducted a large number of dialogue episodes ($\geq 10,000$) to guarantee the statistical significance.

3 Evaluation

Ritel QA dialogue system. Ritel (Galibert et al., 2005) is a telephone-based question answering (QA) dialogue system. Dialogue functionality includes confirmation of key phrases and the type of the answer sought, and handling follow-up questions. In our model, we focus on confirmation, modeling key phrases and answer type as slots. In the real system, there are thousands of possible key phrases, but answer type only has a few possible values. To make it tractable, we simplified the model to one slot with between 3 and 10 values, suitable at least for modeling answer type fully.

The POMDP state space consists of the user goals and the user actions ($S = G_u \times A_u$). The user goals are the different questions or question types that the user may ask, $G_u = q_1, ..., q_n$. The user actions are composed of the questions, plus positive and negative feedback, a 'bye' utterance, and a 'hang-up' signal, $A_u = q_1, ..., q_n, pos, neg, bye, null$. The observation set Z is the same as Au. The system actions consist of confirming each question, answering it, and the 'ask' action, asking an open question to the user, $A = confirm_{q_i}, answer_{q_i}, ask$. When the system answers the correct question, the user poses a new question, otherwise the user either repeats or gives negative feedback. The user may hang up in any dialogue turn, with a fixed probability 0.1.

We made the reward model as simple as possible: give a reward of 1 for answering the right question, -1 for answering a wrong one, zero otherwise. We found that modelling dialogue state was not necessary, and it increases state space to intractable levels. This model yields the desired behaviour, though like Williams et al., we found that the system starts confirming even when the user has not yet said anything. This can be remedied by rewarding the ask action with a reward slightly more than 0. Note that it is not necessary to give an explicit penalty for dialogue length. The problem can be translated as: answer as many questions as possible before the user hangs up. The results of Perseus were not useable, so the experiments were done with ZMDP only. Convergence was good up till nine slot values. We observed that, when the ASR error becomes high, 0.7 or above, the system actually wants to hear a question multiple times in a row before answering it. The policy was compared to a hand-crafted policy (figure 1), similar to the actual Ritel policy, which is based on counting the number of times a particular keyword was heard. It was optimised to each particular problem by determining the optimal number of times a question should be heard before confirmation is sufficient, just as the POMDP does.

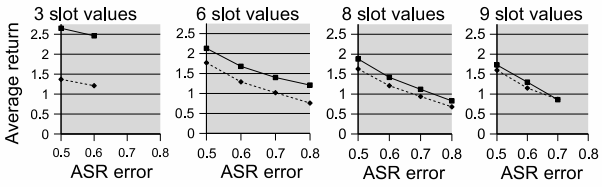


Figure 1: Performance comparison of POMDP and optimised hand-crafted models for different problem sizes and ASR error rates. The solid line is the POMDP, the dashed line is the hand-crafted model. For three values, an error more than 0.6 would result in the probability of hearing the wrong question being higher than the right one. For nine values and error=0.8, no sensible policy could be calculated.

ICIS route navigation system. In the ICIS project[2], we are developing a multimodal human-

[2]http://www.icis.decis.nl/

computer framework for crisis management (Fitrianie, 2007). A subtask of the system is to assist rescuers to find a route description to evacuate victims from an unsafe tunnel. This task has been implemented as a multimodal route navigation dialogue system (Bui et al., 2007).

The simplified POMDP for this problem (one slot case) is represented by $S = \langle G_u \times A_u \times E_u \times D_u \rangle = \langle \{v_0, ..., v_m\} \times \{v_0, ..., v_m, yes, no\} \times \{stress, nostress\} \times \{notstated, stated\} \rangle$, $A = \{ask, confirm\text{-}v_0, ..., confirm\text{-}v_m, ok\text{-}v_0, ..., ok\text{-}v_m, fail\}$, and $Z = \langle OA_u \times OE_u \rangle = \langle \{v_0, ..., v_m, yes, no\} \times \{stress, nostress\} \rangle$. The full flat-POMDP model is composed of $(4m^2 + 8m + 1)$ states (including a special `end` state), $(2m+2)$ actions, and $(2m+4)$ observations.

The transition and observation models are generated from the two time-slices Dynamic Decision Network (Bui et al., 2007). We assume that the observed user's action only depends on the true user's action (i.e. $P(oa_u|a_u) = (1 - p_{oa})$ if $oa_u = a_u$, otherwise $P(oa_u|a_u) = 1/(m + 1) \times p_{oa}$). The observed user's emotional state is computed in a similar way. The reward model is defined as follows: if the system `confirms` when dialogue state is `notstated`, the reward is `-2`, the reward is `-5` for action `fail`, the reward is `10` for action `ok-x` where g_u = `x` (`x` $\in v_0, ..., v_m$), otherwise the reward is `-10`. The reward for any action taken in the absorbing `end` state is `0`. The reward for any other action is `-1`.

We set different values for parameters m, p_e, p_{oa}, p_{oe}[3] and use two POMDP solvers Perseus and ZMDP to compute the near-optimal policy. Previous research showed that the optimal policy depends on the user's stress level in case $p_e > 0$ and the POMDP policy outperforms hand-crafted policies (Bui et al., 2007). The size of the state space of POMDP model increases as the square of slot numbers and computing the optimal policy is not possible when the number of slot values is greater than 30 because the POMDP parameter file size rapidly increases (for example with $m = 30$, the size is bigger than 200MB). Therefore, the POMDP solver got stuck in initializing the problem. An alternative solution is to use DDN-POMDP (Bui et al., 2007) or summary POMDP (Williams and Young, 2007). However, when the number of slot values is greater than 100, the belief update task is not tractable. Therefore, a further research on the POMDP problem representation is necessary. A practical issue is that ZMDP is more suitable for the more complex problem ($10 \leq m \leq 30$). This is because ZMDP is able to handle a larger state space by more effective use of sparsity (Smith and Simmons, 2005). On the other hand, Perseus solves small problems very well. The reason for this was not theoretically indicated in the Perseus paper, but they found the same result when testing with the standard POMDP problems from the literature.

[3] p_e is the probability of the user's action error being induced by stress. p_{oa} and p_{oe} are the probabilities of the observed user's action and observed user's stress errors.

Virtual Guide application. The Virtual Guide is a character in a Virtual Reality model of the Music Centre in Enschede (Hofs et al., 2003). The character can help users find their way in the building. It encompasses a multimodal dialogue system that allows users to refer to locations and objects with spoken or written language or by pointing at a location on a map. The system uses clarification questions and implicit confirmations. The user can continue a dialogue with follow-up questions.

It is currently impossible to create a tractable POMDP model for the system. In our simplified models the user can only ask for the route between two objects, and the world is limited to three or eight objects. Moreover we made a closed model where follow-up questions are not allowed. We have fit the problem into a POMDP dialogue model of Williams. A reward is given when the system gives the correct route and the user provided both locations.

Evaluations were performed for four models. For each of them we compared the solutions of Perseus and ZMDP and an adapted hand-crafted system. The solvers were run for ten minutes, when convergence was usually slowing down, although it had not always reached a desirable level. We then ran dialogues with an automatic user simulation based on the user model of the POMDP.

The first model stops after giving any answer, has observation error 0.2, and three locations. We varied the observation error of the simulator. The results in figure 2 show that with increasing errors the POMDP solutions produced higher returns than the

hand-crafted system.

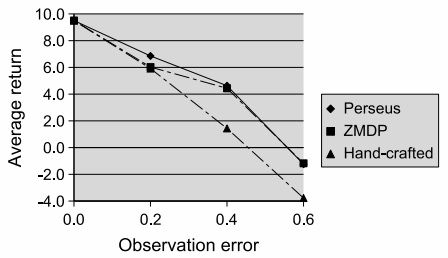


Figure 2: Average returns for simulation with different observation errors

For the second model, we increased the observation error to 0.6. The Perseus solution contained a state from which the dialogue never ended. ZMDP did not converge acceptably. Therefore its solution performed worse than the hand-crafted system.

The third model has observation error 0.2 again, but the dialogue only stops after giving a correct answer. The average returns for Perseus, ZMDP and the hand-crafted system were 8.08, 6.84 and 6.69 (higher than the first model, because of a reward for an extra system action).

In the last model we increased the number of locations to eight, resulting in 729 POMDP states instead of about 80. Perseus was not able to load this problem. The average returns obtained with ZMDP and the hand-crafted system were 5.08 and 4.04.

4 Conclusions

Although our experiments indicate that POMDP-based dialogue systems can perform better than hand-crafted ones, we identified several problems with modelling them. One of the major problems remains tractability. It is not possible to obtain useful solutions for any but strongly simplified models, which may bear little relation to the original problem. For example, when reducing the number of slot values, the strategy of trying them one by one can be employed, something that may not have been feasible for the original number of values. Another example was the need to simplify an open model, where the end of a dialogue is determined by the user, to a closed model.

The definition of a good reward model is another hard problem. While the reward model models psychological factors such as user satisfaction, which cannot easily be quantified precisely, the POMDPs proved very sensitive to small changes in the reward model, in particular the relative magnitude of different types of reward. In practice we had to experiment with different reward values.

The POMDP policies sometimes came up with surprising strategies. For example, some policies decided to confirm multiple times in a row, something which our original hand-crafted models did not. We could significantly improve performance of the hand-crafted policies by adapting them according to the strategies found by the POMDP policies. This shows how POMDPs could be used to improve hand-crafted systems.

Acknowledgements. This work is part of the ICIS and IMIX programs. ICIS is sponsored by the Dutch government under contract BSIK 03024. IMIX is funded by the Netherlands Organization for Scientific Research (NWO).

References

T.H. Bui, M. Poel, A. Nijholt, and J. Zwiers. 2007. A tractable ddn-pomdp approach to affective dialogue modeling for general probabilistic frame-based dialogue systems. In *Proc. of the 5th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 34–37.

S. Fitrianie. 2007. A multimodal human-computer interaction framework for research into crisis management. In *Proc. of the Intelligent Human Computer Systems for Crisis Response and Management*, pages 149–158.

O. Galibert, G. Illouz, and S. Rosset. 2005. Ritel: an open-domain, human-computer dialog system. In *Interspeech 2005*, pages 909–912.

D. Hofs, R. op den Akker, and A. Nijholt. 2003. A generic architecture and dialogue model for multimodal interaction. In *Proc. of the 1st Nordic Symposium on Multimodal Communication*, pages 79–92.

Trey Smith and Reid Simmons. 2005. Point-based pomdp algorithms: Improved analysis and implementation. In *Proc. of the Conference on Uncertainty in Artificial Intelligence*, pages 542–549.

M.T.J. Spaan and N. Vlassis. 2005. Perseus: Randomized point-based value iteration for pomdps. *Journal of Artificial Intelligence Research*, 24:195–220.

J.D. Williams and S. Young. 2007. Partially observable markov decision processes for spoken dialogue systems. *Computer Speech and Language*, 21:393–422.

Problem-Sensitive Response Generation in Human-Robot Dialogs

Petra Gieselmann*
Universität Karlsruhe, Germany
petra@ira.uka.de

Mari Ostendorf
University of Washington, Seattle USA
mo@ee.washington.edu

Abstract

We develop a new mechanism to detect and respond to miscommunications in human-robot dialogs, distinguishing between computer misunderstandings vs. human inexperience. Problem indicators drive an error/help state machine, which augments the dialog state and is used in tailoring response generation. A user study shows that the task success rate and user satisfaction is improved substantially by the two-part miscommunication model.

1 Introduction

A major challenge in spoken dialog systems is miscommunication: speech recognition errors and misunderstandings often result in error spirals from which the user can hardly escape. This leads to user frustration and task failure. The problem of error handling has been the subject of several studies, and errors often cause system designers to use a dialog management strategy based on system initiative and explicit confirmation. In human-robot dialogs, miscommunication can arise not only from imperfect speech recognition and understanding, but also from user uncertainty about the robot's knowledge and capabilities. Moreover, in most applications for humanoid robots, there is a need for a mixed-initiative dialog strategy to support more natural communication.

To handle human-robot miscommunication in this context, we have developed a strategy to detect and track two types of problems: computer misunderstanding vs. user need for help. Computer misunderstandings may be due to speech recognition or understanding errors. Users may need help, either because they are unsure how to proceed in a dialog or because they are asking for something that is beyond the current capability/knowledge of the robot. From the robot's perspective, these problems can sometimes be hard to distinguish, but they require different responses in order to resolve miscommunications.

Consider the following extract of an actual dialog with our robot, where "U" is the human user and "R" is the robot. The robot knows about things in the kitchen; it can help with meals by reciting recipes and bringing things to the human; but it cannot actually cook.

U: Robbi, I am very hungry. Could you please prepare something to eat?
R: In the fridge, there are tomatoes, eggs, milk, and butter. With these ingredients you can make different dishes, such as Spaghetti Napoli or a pancake. What do you want to eat?
U: Spaghetti Napoli please.
R: Sorry, I didn't catch that.
U: Robbi, you just told me that it is possible to cook some Spaghetti Napoli. I would like to have that please.
R: Okay.
U: Fine, thank you.
R: Thank you.
U: Could you prepare something, a dessert?
R: [silence]
. . .
U: Don't you want to cooperate with me?

In the first utterance, the robot failed to detect that "prepare" indicated that the user was asking it to do something beyond its capability. Thus, the user assumes that it can cook, which seems to be confirmed when the robot says "okay" to "it is possible to cook...," though for the robot this means that the ingredients are available. Recognition errors further complicate the misunderstanding, and not surprisingly lead to frustration.

To deal with such problems, variables indicative of different types of errors are tracked, based on analyses of the recognized user utterance. Together with the current dialog state and "problem state," these variables are used to predict whether the conversation is functioning normally vs. in an error spiral or help-needed condition. The response generation strategy is then adjusted accordingly, both in terms of the type of response and its wording. The details of this strategy and experimental validation are described below, following a review of related work.

2 Related Work

There have been several analyses of communication failures in human-computer dialog, looking at characteristics of utterances where speech understanding errors occur, as well as those of attempted corrections of errors. Studies have found that the speech recognition error rate increases with increasing depth into the error correction subdialog (Swerts et al., 2000), as does user frustration

*Now with Lucy Software and Services GmbH, Munich, Germany

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 219–222,
Antwerp, September 2007.

(Bulyko et al., 2005). There are studies showing that error corrections have acoustic and prosodic features different from normal user utterances (Swerts et al., 2000), and combining acoustic and lexical cues to detect corrections, e.g. (Kirchhoff, 2001). Such studies inform speech recognizer design as well as automatic error (correction) detection.

Other studies have focused on factors that impact the dialog management strategy. Shin et al. (Shin et al., 2002) analyzed 161 dialogs from the NIST 2000 Communicator Evaluation (Walker et al., 2001) in terms of system behavior in order to find out how users discover that an error occurred. The results revealed the need for more explicit confirmations, since users need more time to get back on track and fail more often when they discover errors through implicit (vs. explicit) confirmations. Results from human-human communication also stress the need for explicit confirmations in error subdialogs (Gieselmann, 2006). An approach for using error correction detection output to decide between different degrees of system initiative in the generation strategy is outlined in (Bulyko et al., 2005), together with generation wording variations motivated by studies showing effects on user frustration. The goal of this work is to extend the results on dialog strategy and response wording to problems that include not only error handling but also human inexperience, with the goal of shortening miscommunications and increasing user satisfaction.

Within the robotics community, new application domains such as taking care of old people, delivering hospital meals, etc., are driving the development of robots that can interact with humans. It is considered important that people can communicate with these robots as to another human (Sidner and Dzikovska, 2005). Most research in this field concentrates on designing the robot as similar as possible to a human in terms of both its appearance and its communicative behavior (Breazeal, 1999). The focus here on help responses is consistent with this view.

3 Task and Baseline System

Our robot can accomplish different tasks in the household environment; e.g. it can deliver and retrieve kitchen items, switch on or off lights, and provide information about recipes or about the contents of the refrigerator (Gieselmann et al., 2003). The robot should be able to interact with inexperienced and older users, e.g. in assisted living situations, so it is important that the communication be as comfortable as possible for the user. In addition, since the robot does not yet have all of the capabilities of a human and an inexperienced user will not know its limits, it is important that the robot can inform the user about its capabilities.

For speech recognition, we use the JANUS Recognition Toolkit with the IBIS decoder which decodes using a grammar controlled by the dialog manager, which penalizes specific rules depending on the situational context (Fügen et al., 2004). The recognizer grammar also provides a parse for interpreting the utterance. It is a context-free grammar enhanced by information from the ontology defining all the objects, tasks and properties about which the user can talk. The parse tree is converted into a semantic representation and added to the current discourse. The semantic representation consists of the speech act and the objects/properties expressed within the user utterance.

For dialog management, we use the TAPAS dialog tools (Holzapfel, 2005) based on the language- and domain-independent dialog manager ARIADNE (Denecke, 2002), which uses typed feature structures to represent semantic input and discourse information. If all the information necessary to accomplish a goal is available in discourse, the dialog system calls the corresponding service. Otherwise, clarification questions are generated using a template-based approach.

4 Mixed Initiative Dialog Management

Our strategy is to try distinguish between problems due to system errors vs. human inexperience, using different indicators of possible communication problems and a separate problem state model with problem-sensitive response generation, as described next.

4.1 Factors Indicating Problematic Situations

Computer misunderstandings can occur for a variety of reasons. The system has to cope with high variability in spontaneous speech, self corrections, segmentation errors, and barge-in, for example. An utterance may include words that are out of the recognizer's vocabulary, either an infrequent wording of a known concept or a totally new concept. Since the recognizer will hypothesize words that are consistent with its vocabulary and language model, the robot can only detect these problems indirectly. Implicit error indicators we use include:

- *The utterance is not parsed or only partly parsed.*
- *No speech act can be found, neither in the user utterance nor in the discourse.*
- *The user utterance is inconsistent with the current discourse or with the robot's expectations.*
- *The user repeatedly asks for the same information.*

In addition, some problems are explicitly indicated:

- *The user explicitly asks for help.*
- *The user tries to correct a preceding utterance.*
- *The user asks for something that the robot knows it cannot yet do, such as cleaning.*

4.2 Problem State Model

For representing different problems, we developed a 4-state finite-state automaton on top of the dialog manager:

- **Start state**: Used at the start of a dialog and between tasks as an idle state; the discourse history is empty.
- **Error state**: Information needs to be corrected.
- **Help state**: The user does not know how to proceed and needs help by the robot about its capabilities.
- **Normal state**: No known problematic situation.

The transitions between the states are rule-based, determined by the information in the discourse history and the user utterances, and the problem indicators. The robot is initially in the start state and goes to the normal state as long as no problems occur. Implicit problem indicators trigger a transition to either error state or help state, depending on whether there is information available in the discourse that the user might want to correct. The user stays in the help state (or in the error state) as long as the problems persist. After a non-problematic utterance the user returns to the normal state. To switch from the help state to the error state, a user utterance must contain some information which is put in the discourse. The user can also put the system into the help state or the error state directly by uttering an explicit help request or error correction, respectively. In addition, whenever a user asks for a known task the robot cannot accomplish, such as cleaning, the user is also transferred to the help state. When an error is resolved, the user goes back to the normal state. The system goes back to the start state and the discourse is cleared whenever a user request to the robot has been met or the user explicitly clears the discourse using an utterance such as ”start over” or ”abort”.

4.3 Problem-Sensitive Response Generation

In order to appropriately respond to the user, we have the following features to keep track of the ongoing situation:

- HELP NECESSITY: a variable that increases with each problem indicator, and decreases with a transition to the normal state (to some minimum).
- ERROR SPIRAL: count of the number of successive turns in the error state, cleared after a transition to the normal or start states.
- USER KNOWLEDGE: a list that contains the information already given to the user and how many of times it was given.[1]

Within the help state, the user will get information about the robot's capabilities. The full set of robot capabilities is too large to describe in one response, so we use a set if responses organized according to a task hierarchy. At the highest level, the most general capabilities are described, i.e. for a dialog with a new user, and details related to those capabilities are covered in lower level responses. The user knowledge list is used to determine whether the user has already been given a particular help message. If so, the user is either given a different help message for that dialog state or the robot asks the user if s/he would like to hear again about the robot capabilities. When the help necessity gets above a given threshold, the robot asks the user to speak some predefined sentences to better adapt to the user's voice, and the problem state is reset to the "start" state.

Within the error state at the beginning of the dialog, the user is asked to check microphone placement. Later, potential errors are handled by a repeat request, with different wording depending on the error spiral as in (Bulyko et al., 2005). In cases of repeated requests that are out of scope, the robot explicitly tells the user tasks that it cannot do.

[1] We track only the current interaction; long term knowledge from multiple interactions is not addressed here.

	Baseline	V1	V2
No Predefined Task			
Concept Error Rate	68%	52%	49%
No. of new Concepts	5.0	2.8	4.6
With Predefined Task			
Concept Error Rate	50%	42%	25%
No. of new Concepts	3.0	1.4	2.1
Task Completion Rate	57%	70%	96%
Turns per Task	8.4	5.1	2.7

Table 1: Results of the User Study

5 Experimental Details and Results

We conducted a user study to assess the impact of using a general help/error state vs. separating the help and error correction modes. Two different development cycles of the dialog system were tested and compared to a baseline system that had no explicit error handling. Version 1 (V1) used a dialog management and generation strategy with a single state for errors and help together, and version 2 (V2) includes a division of the problem handling into error vs. help states.

We tested V1 and V2 each with 8 users, with no overlap of people in these groups. The baseline system was tested with 3 trials, with 1 person running two trials of the baseline system and 1 trial of V1. Of the 16 people participating, half were native speakers of English and half were fluent English speakers with another native language. All subjects were familiar with computers, but only six had talked to a dialog system before.

The user study consists of three parts. The first part was a free interaction with the robot: users were told that

they had a new household robot that can support them in the kitchen. This situation is more realistic, but also harder for the users because they have limited knowledge of what the robot can do. In the second part, each user was given (the same) 10 tasks to accomplish with the robot. Using specified tasks, we can assess task completion, but we get less information on the types of capabilities that users expect. After the dialog with the robot, users fill in a short questionnaire. They answered three directed questions about how much they liked the system, how successful they were, and how much they would like to use such a robot again. In additional open questions, participants could report their problems and suggestions for further improvements.

To evaluate the dialogs, we measured concept error rate (percent of semantic concepts not understood by the system for whatever reason) and tracked the number of new concepts introduced. The semantic concepts include actions (e.g. bring, report) and objects (e.g. cup, cabinet) in the robot's ontology. For the predefined tasks, we also tracked task completion and number of turns per task.

The fact that the concept error rate decreases with each design cycle (cf. Table 5) confirms the usefulness of error handling in general, and specifically the separation of error and help needs. As expected, the concept error rate and the average number of new concepts decrease when the subjects are given predefined tasks. Note that, even when the tasks are predefined, users still invent new concepts that the robot does not know, so the help functionality is still useful. (The drop in the number of new concepts between the baseline and V1 may be due to user learning; all users of V2 were new to the task.) For the dialogs with predefined tasks, there is an increase in the task completion rate and a decrease in the number of turns per task for each step in the design cycle, with bigger changes in moving from V1 to V2. Differences found in the condition without predefined tasks do not reach statistical significance, but all the differences within the predefined task condition are significant (p-value smaller than .008 for concept error rate and p-value smaller than .005 for turns per tasks and task completion rate).

The results of the user survey revealed that in V2 the users liked the robot more and felt they had been more successful in their interactions, compared to V1. The differences in responses related to whether they would like to use such a robot again were not significant, possibly because problem handling does not impact the robot's actual capabilities. Within the free-text answers, some users mentioned the nice recovery after misunderstandings and stressed that it was very clear about its capabilities.

6 Conclusion

In summary, we developed a new dialog management strategy which is sensitive to different types of miscommunications in human-robot dialogs. We use several types of problem indicators to drive state transitions in a 4-state indicator of error/help modes. The generation strategy is modified according to the type of problem, if any. The results of a user study showed that the task success rate, concept accuracy, and user satisfaction are improved substantially by these changes.

In the future, the error handling component can be improved by expanding the problem state space, and including new features such as word confidence, out-of-vocabulary word detection, acoustic cues, and new problem indicators. Another potential direction is to use the problem indicators as input to a Markov decision process for controlling the dialog state. Finally, we note that automatic learning of new concepts and skills on the robot's side will require dynamic update of the problem tracking and help response generation mechanisms.

Acknowledgment: This research is partially funded by the German Research Foundation (DFG) under SFB 588.

References

C. Breazeal. 1999. Robot in society: Friend or appliance? *Proc. Workshop on Emotion-based Agent Architectures*.

I. Bulyko *et al.* 2005. Error correction detection and response generation in a spoken language dialogue system. *Speech Communication*, 45:271-288.

M. Denecke. 2002. Rapid prototyping for spoken dialogue systems. *Proc. ACL*, pp. 1-7.

C. Fügen, H. Holzapfel, and A. Waibel. 2004. Tight coupling of speech recognition and dialog management. *Proc. ICSLP*.

P. Gieselmann *et al.* 2003. Towards multimodal communication with a household robot. *Proc. HUMANOIDS*.

P. Gieselmann. 2006. Comparing error-handling strategies in human-human and human-robot dialogues. *Proc. KONVENS*, pp. 24-31.

H. Holzapfel. 2005. Towards development of multilingual spoken dialogue systems. *Proc. LTC*.

K. Kirchhoff. 2001. A comparison of classification techniques for the automatic detection of error corrections in human-computer dialogues. *Proc. NAACL Workshop on Adaptation in Dialogue Systems*.

J. Shin *et al.* 2002. Analysis of user behavior under error conditions in spoken dialogs. *Proc. ICSLP*, pp. 2069-2072.

C. Sidner and M. Dzikovska. 2005. A first experiment in engagement for human-robot interaction in hosting activities. N.O. Bernsen *et al.* (Eds.), *Advances in Natural Multimodal Dialogue Systems*.

M. Swerts, J. Hirschberg, and D. Litman. 2000. Corrections in spoken dialogue systems. *Proc. ICSLP*.

M. Walker *et al.* 2001. DARPA Communicator dialog travel planning systems: the June 2000 data collection. *Proc. Eurospeech*, 2:1371-1374.

Rapid Development of Dialogue Systems by Grammar Compilation*

Björn Bringert
Department of Computer Science and Engineering
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden
bringert@cs.chalmers.se

Abstract

We propose a method for rapid development of dialogue systems where a Grammatical Framework (GF) grammar is compiled into a complete VoiceXML application. This makes dialogue systems easy to develop, maintain, localize, and port to other platforms, and can improve the linguistic quality of generated system output. We have developed compilers which produce VoiceXML dialogue managers and ECMAScript linearization code from GF grammars. Along with the existing GF speech recognition grammar compiler, this makes it possible to produce a complete mixed-initiative information-seeking dialogue system from a single GF grammar.

1 Introduction

In current industrial practice, dialogue systems are often constructed using VoiceXML for dialogue management, context-free speech recognition grammars for input, with semantic tags for interpretation, and concatenation of canned text and output data for system responses. Developing several components which all need to cover the same concepts increases development costs. Having multiple interdependent components in formalisms with few automatic correctness and consistency checks also complicates maintenance, since any change in the coverage of one component may require changes in the others. Since all the components are language-specific, much effort is needed to port the system to a new language, and to keep the implementations for different languages in sync. The lack of a powerful method for output realization makes it hard to generate high-quality output, especially for languages with a more complex morphology than English.

We address these problems by specifying the system in a single high-level formalism, which is then compiled into the existing lower-level formalisms. The developer writes a GF abstract syntax module which defines the user input and system output semantics, and a concrete syntax module which describes how each construct in the semantics is represented in natural language. The GF grammar is then compiled to a complete VoiceXML application. The dialogue flow is determined by the abstract syntax (ontology) of the grammar. This is based on the idea by Ranta and Cooper (2004) that a proof editor for constructive type theory can be used to implement the information gathering phase of information-seeking dialogue systems.

In contrast to earlier rapid dialogue system development approaches, such as CSLU's RAD (McTear, 1999) and the GEMINI AGP (Hamerich et al., 2004), we use a compiler-like model instead of a graphical design environment. In addition, our development model is focused on the specification and realization of the inputs and outputs of the system, rather than on the dialogue flow or the underlying database. Compared to existing dialogue systems built with GF (Ericsson et al., 2006), our approach does not require an external dialogue manager.

*This work has been partly funded by the EU TALK project, IST-507802, and Library-Based Grammar Engineering, Swedish Research Council project dnr 2005-4211.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 223–226,
Antwerp, September 2007.

2 Grammatical Framework

Grammatical Framework (GF) (Ranta, 2004) is a grammar formalism based on constructive type theory. GF separates grammar into *abstract syntax* and *concrete syntax*.

2.1 Abstract Syntax

The abstract syntax defines the ontology of the application, that is, *what* can be said. An abstract syntax contains category (**cat**) and function (**fun**) definitions. This is an example of a small abstract syntax:

cat Order; Size;
fun *pizza* : Size → Order; *small* : Size;

This allows us to construct an abstract syntax term *pizza small* of type Order. In addition to functions, abstract syntax terms can also contain *metavariables*, written ?. For example, the term *pizza* ? contains a metavariable of type Size.

2.2 Concrete Syntax

A concrete syntax defines *how* each abstract syntax construct is realized in a particular language. From a concrete syntax, the GF system can derive both parsing and realization components. A concrete syntax contains linearization type (**lincat**) and linearization (**lin**) definitions. The linearization type of a category is the type of the concrete syntax terms produced for abstract syntax terms in the given category. A linearization definition is a function from the linearizations of the arguments of an abstract syntax term to a concrete syntax term. Terms in concrete syntax can be records, strings, tables, and parameters. This is an example of a concrete syntax for the abstract syntax above:

lincat Order, Size = $\{s : \mathsf{Str}\}$;
lin *pizza* $x = \{s =$ "a" ++ $x.s$ ++ "pizza"$\}$;
small $= \{s =$ "small"$\}$;

3 An Example Dialogue System

This section shows a GF grammar from which a complete dialogue system (excluding the domain resources) can be derived automatically. For reasons of brevity, this system is very small. An extended version of this system is available online[1].

[1] http://www.cs.chalmers.se/~bringert/xv/pizza/

3.1 Abstract Syntax

The abstract syntax in Figure 1 describes the possible things that the user can say, in a semantic form. There is one category for each kind of input. In this application, the main input object is an Order. A order can in this small example only be for a number of pizzas, all of the same size and with the same topping. A number is "one" or "two", the sizes are "small" and "large", and the toppings are "ham" and "cheese". An example abstract syntax term in the Order category is: *pizza two small cheese*.

abstract Pizza = {
flags startcat = Order;
cat Order; Number; Size; Topping;
fun *pizza* : Number → Size → Topping → Order;
one, *two* : Number;
small, *large* : Size;
cheese, *ham* : Topping;
}

Figure 1: Abstract syntax for the example system.

3.2 Concrete Syntax

The concrete syntax in Figure 2 defines how the terms in the abstract syntax are realized (and inversely, how concrete syntax terms can be interpreted as representations of abstract syntax terms). For example, the linearization type of Topping is $\{s : \mathsf{Str}\}$, that is, a record with a single field s which contains a string. The linearization for *cheese* is the concrete syntax term $\{s =$ "cheese"$\}$.

The linearization type of Number contains a field n, which is used for agreement. The type of n is Num, defined by a **param** definition to be either Sg or Pl. In the linearization of *pizza*, the n field of the Number is used to inflect the noun "pizza".

An important feature of this grammar is that it allows partially specified input. While the utterance "two small pizzas with cheese" results in the abstract syntax term *pizza two small cheese*, the partial versions "two pizzas with cheese" (*pizza two ? cheese*), "two small pizzas" (*pizza two small ?*), and "two pizzas" (*pizza two ? ?*) are also allowed. The intention is that the system will ask follow-up questions to replace all metavariables with complete terms. This process is type-directed: the system asks for a subterm of the appropriate type. Partial input, imple-

```
concrete PizzaEng of Pizza = {
lincat Number = {s : Str; n : Num};
       Order, Size, Topping = {s : Str};
param Num = Sg | Pl;
printname cat
  Order   = "What would you like to order?";
  Size    = "What size pizzas do you want?";
  Topping = "What topping do you want?";
lin pizza n s ts = {s =
      n.s ++ variants {s.s; []} ++ pizza_N.s ! n.n
      ++ variants {"with" ++ ts.s; []}};
    one     = {s = "one"; n = Sg};
    two     = {s = "two"; n = Pl};
    small   = {s = "small"};
    large   = {s = "large"};
    cheese  = {s = "cheese"};
    ham     = {s = "ham"};
oper pizza_N = {s = table {Sg => "pizza";
                           Pl => "pizzas"}};
}
```

Figure 2: Concrete syntax for the example system.

mented with suppression, is thus used to achieve a mixed-initiative dialogue.

The **printname** definitions are used as prompts for each category.

3.3 Example Dialogues

The system generated from the grammar in the previous section allows dialogues such as the examples below. After each system action we show the information state, i.e. the current state of the abstract syntax term that we are constructing.

S: What would you like to order?
U: two pizzas
pizza two ? ?
S: What size pizzas do you want?
U: small
pizza two small?
S: What topping do you want?
U: ham
pizza two small ham

Here, more information is given in the first answer:

S: What would you like to order?
U: two pizzas with ham
pizza two ?*ham*
S: What size pizzas do you want?
U: small
pizza two small ham

3.4 Extending the Example System

Recursive structures One possible extension to the example system is to use a recursive structure to allow more complex orders:

```
cat Order; Item; [Item];
fun order : [Item] → Order;
    pizza : Number → Size → Topping → Item;
printname cat [Item] = "Anything else?";
lin order is = {s = is.s};
    ConsItem x xs =
      {s = x.s ++ variants {"and" ++ xs.s; []}};
    BaseItem = {s = "nothing" ++ "else"};
```

While this can be done with subdialogues and scripting in VoiceXML (by essentially writing by hand the code that we generate), it appears to be beyond the scope of standard practice. If we also add drinks as a kind of Item, the system will support dialogues such as this one:

S: What would you like to order?
U: one large pizza
order [*pizza one large*?, ?]
S: What topping would you like?
U: cheese
order [*pizza one large cheese*, ?]
S: Anything else?
U: one beer
order [*pizza one large cheese*, *drink one beer*, ?]
S: Anything else?
U: nothing else
order [*pizza one large cheese*, *drink one beer*]

System output At the end of the dialogue, we would like the system to give a response based on the output of some domain resource. For example, the pizza ordering system might return the price of the order. This could be used to construct a confirmation using an addition to the grammar:

```
cat Output;
fun confirm : Order → Number → Output;
lin confirm o p =
  {s = o.s ++ "costs" ++ p.s ++ "euros"}
```

Multilinguality To port a dialogue system to a new language, all that needs to be done is to write a new concrete syntax. For many languages, writing speech recognition grammars and realization functions is more complicated than for English. For example, Swedish adjectives agree with the gender and number of the noun they modify. GF's expressive concrete syntax makes it possible to implement such features with little effort, and if the GF Resource Grammar Library is used, it is as easy to write the Swedish grammar as the English.

Multimodality GF can be used to write multimodal grammars (Bringert et al., 2005). The extended online version of the example system uses a concrete syntax which linearizes pizza and drink orders to vector drawings to display graphical representations of the completed orders.

4 Implementation

The concrete syntax is compiled (Bringert, 2007) to an SRGS speech recognition grammar, with SISR semantic interpretation tags. This grammar has one category for each GF category.

The abstract syntax and the prompts from the concrete syntax are compiled to a VoiceXML application with one form for each GF category. Each such form takes an argument, which the caller sets to the currently known abstract syntax term. If the given term is a metavariable, input is requested in the appropriate speech recognition grammar category. For each subterm of the abstract syntax term returned by the semantic interpretation, a subdialogue call is made to the corresponding VoiceXML form.

The concrete syntax is also compiled to an ECMAScript program which can be used to linearize system outputs.

5 Future Work

Currently, the dialogue model is quite limited. For real-world use, more flexible dialogue management would be needed. The Trindi tick list (Bohlin et al., 1999) could be used to guide such work. Other possibilities could include support for dependently typed abstract syntax (Ranta and Cooper, 2004), a help system with automatically generated examples for each category, and context-sensitive prompt generation.

6 Conclusions

We have shown that GF grammars can be used to implement mixed-initiative information-seeking dialogue systems. From the declarative and linguistically powerful specification that a GF grammar is, we generate the interconnected components needed to run dialogue systems using industry standard infrastructure. Hopefully, this method can reduce the development and maintenance costs for dialogue systems, and at the same time improve their linguistic quality. The methods described in this paper are implemented as part of the open source GF system[2].

References

Peter Bohlin, Johan Bos, Staffan Larsson, Ian Lewin, Colin Matheson, and David Milward. 1999. Survey of Existing Interactive Systems. D 1.3, TRINDI.

Björn Bringert, Robin Cooper, Peter Ljunglöf, and Aarne Ranta. 2005. Multimodal Dialogue System Grammars. In *Proceedings of DIALOR'05, Ninth Workshop on the Semantics and Pragmatics of Dialogue*.

Björn Bringert. 2007. Speech Recognition Grammar Compilation in Grammatical Framework. In *Proceedings of the Workshop on Grammar-Based Approaches to Spoken Language Processing*.

Stina Ericsson, Gabriel Amores, Björn Bringert, Håkan Burden, Ann-Charlotte Forslund, David Hjelm, Rebecca Jonson, Staffan Larsson, Peter Ljunglöf, Pilar Manchón, David Milward, Guillermo Pérez, and Mikael Sandin. 2006. Software illustrating a unified approach to multimodality and multilinguality in the in-home domain. D 1.6, TALK.

Stefan Hamerich, Volker Schubert, Volker Schless, Ricardo de Córdoba, José M. Pardo, Luis F. d'Haro, Basilis Kladis, Otilia Kocsis, and Stefan Igel. 2004. Semi-Automatic Generation of Dialogue Applications in the GEMINI Project. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*.

Michael F. McTear. 1999. Software to support research and development of spoken dialogue systems. In *Proceedings of Eurospeech'99*.

Aarne Ranta and Robin Cooper. 2004. Dialogue Systems as Proof Editors. *Journal of Logic, Language and Information*, 13(2):225–240.

Aarne Ranta. 2004. Grammatical Framework: A Type-Theoretical Grammar Formalism. *Journal of Functional Programming*, 14(2):145–189.

[2]See `http://www.cs.chalmers.se/~aarne/GF/`

Resolving *"You"* in Multi-Party Dialog*

Surabhi Gupta
Department of Computer Science
Stanford University
Stanford, CA 94305, US
surabhi@cs.stanford.edu

John Niekrasz, Matthew Purver
Center for the Study
of Language and Information
Stanford University
Stanford, CA 94305, US
{niekrasz,mpurver}@stanford.edu

Dan Jurafsky
Department of Linguistics
Stanford University
Stanford, CA 94305, US
jurafsky@stanford.edu

Abstract

This paper presents experiments into the resolution of *"you"* in multi-party dialog, dividing this process into two tasks: distinguishing between generic and referential uses; and then, for referential uses, identifying the referred-to addressee(s). On the first task we achieve an accuracy of 75% on multi-party data. We achieve an accuracy of 47% on determining the actual identity of the referent. All results are achieved without the use of visual information.

1 Introduction

This paper discusses the disambiguation and resolution of second-person pronoun *"you"* in dialog. Our objectives are to automatically disambiguate the *referential* and *generic* senses, and to resolve the referential instances through *addressee identification* using linguistic features.

This work is motivated by the effort to automatically extract useful information from multi-party human-human conversation. For example, Purver et al. (2006) attempt to extract action items: concrete decisions in which one (or more) individuals makes a commitment to perform a given task. One critical roadblock is that of personal deixis: identification of the individuals in question usually involves personal pronouns – either singular or plural, as (1) and (2) – rather than explicit naming.

(1)
A: and um if **you** can get that binding point also maybe with a nice example that would be helpful for **Johno** and **me**.
B: Oh yeah uh O_K.

(2)
A: So y- so **you guys** will send to **the rest of us** um a version of um, this, and - the - uh, description -
B: With sugge- yeah, suggested improvements and -

However, to complicate the matter, many second-person pronoun uses[1] do not refer to the addressee and are generic, as in example (3):

(3)
B: Well, usually what **you** do is just wait until you think it's stopped,
and then **you** patch them up.

The problem of *"you"* resolution can therefore be seen as two sub-problems: (1) determining whether a *"you"* is referential, and if so, (2) determining the number and identity of the addressee(s) referred to.

2 Related Work

Prior linguistic work has recognized that *"you"* is not always addressee-referring – e.g. Jurafsky et al. (2002) distinguish three cases (generic, referential, and the conventional phrase *"you know"*) in an investigation of phonological form – but little work exists on automatic disambiguation.[2] In (Gupta et al., 2007), we described such a study (to our knowledge, the first) on two-person telephone conversations, achieving 84% accuracy on disambiguating generic and referential senses. Here, we extend this to multi-party data, and add addressee resolution.

In multi-party dialog (unlike two-party dialog), addressee identification is an important problem, and there is a growing body of research addressing this (Jovanovic et al., 2006b; Katzenmaier et

*This work was supported by the CALO project (DARPA grant NBCH-D-03-0010) and ONR (MURI award N000140510388).

[1] There are five second-person pronouns in English: *you*, *your*, *yours*, *yourself*, and *yourselves*.

[2] Although there is related work on classifying *"it"*, which also takes various referential and non-referential readings (Müller, 2006).

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 227–230,
Antwerp, September 2007.

al., 2004). Jovanovic et al. (2006b) achieve an accuracy of 83.74% when including visual features such as gaze information, along with more complex information such as meeting action types (e.g. discussion, presentation, white-board). Galley et al. (2004) showed some success using only speech-based information for a related problem – identifying the first halves of adjacency pairs (whose speakers will in many cases be the addressees of the second halves) – achieving an accuracy of 90.2%. Our approach in this paper is closer to the latter in using only non-visual information, in order to support a solution in environments lacking video.

3 Data

We used the AMI Meeting Corpus (McCowan et al., 2005), a multi-modal dataset of 4-party meetings. The meetings are scenario-driven – participants are assigned roles in a loosely scripted collaborative design task, averaging about 30 minutes in duration. All meetings are hand-transcribed and annotated for dialog acts; we used a 15-meeting subset which is also annotated for addressee (Jovanovic et al., 2006a), with each utterance labeled with the set of addressees. Jovanovic et al. (2006a) report that 34.2% of utterances were addressed to all participants, 61.7% were addressed to single individuals, with <2% being addressed to 2-person subgroups.

We randomly selected a subset of utterances containing *"you"* to annotate. Only text and/or audio were made available to annotators – no videos were provided during annotation. The result was a 4-way classification on a per-utterance basis using the following classes: *generic*, *referential*, *reported referential*, and *discourse marker*. Examples of the first three of these classes are given above. The *reported referential* class was used to mark when speakers are quoting other speakers' referential uses, as in example (4). Finally, the *discourse marker* class was used to mark instances of the commonly-occurring, semantically bleached version of *"you know"*.

(4)
B: Well, uh, I guess probably the last one I went to I met so many people that I had not seen in probably ten, over ten years.
It was like, don't **you** remember me.
And I am like no.
A: Am I related to **you**?

The reliability of our annotations was acceptable, with kappa of .84 and raw inter-tagger accuracy of .92 (assessed over a subset of 108 instances tagged by two authors). The resulting dataset for generic versus referential consisted of 952 utterances for training and 374 for test; overall, 47.4% of cases were generic. Since the addressee annotations do not cover all utterances in the meetings, the dataset for addressee detection had only 291 utterances for training and 176 utterances for testing (this set of experiments were performed for the utterances marked as referential); 59.7% of the utterances were addressed to one person.

For the experiments below, we excluded the *reported referential* and the *discourse marker* class since they both occurred in less than 2% of the dataset. Note also that the author performing classification experiments annotated the training set, reserving the test set for annotation by another author.

4 Referentiality Detection

We first investigate the disambiguation of generic versus referential uses. In our earlier work on the two-party Switchboard corpus, we achieved an accuracy of 84.4%, significantly above the baseline performance of 54.6% (always predicting the dominant class). The best classifier made use of a diverse set of features including lexical, part-of-speech, and dialog act features, together with a set of oracle context features (which assumed perfect knowledge of the classes of the preceding utterances).

Here, as well as applying the approach to more complex multi-party data, we wanted to remove the requirement for these unrealistic oracle context features. We therefore used a sequence classifier — conditional random field (CRF), first introduced by Lafferty et al. (2001) — allowing us access to the same contextual information, but via the output of the classifier. The full set of features is shown in Table 1.

Note that in the absence of an available DA tagger for this data, we use manually produced DA tags. This is also unrealistic; we therefore investigated the substitution of the full DA tagset features with a single Q_DA feature which indicates the presence of a questioning dialog act (the AMI `elicit` acts).

N	Features
	Sentential Features (Sent)
2	you, you know, you guys
N	number of you, your, yourself
2	you (say|said|tell|told|mention(ed)|mean(t)|sound(ed))
2	you (hear|heard)
2	(do|does|did|have|has|had|are|could|should|n't) you
2	"if you"
2	I|we
2	(which|what|where|when|how) you
	Part of Speech Features (POS)
2	Comparative JJR tag
2	you (VB*)
2	(I|we) (VB*)
2	(PRP*) you
	Dialog Act Features (DA)
16	DA tag of current utterance i
16	DA tag of previous utterance $i-1$
16	DA tag of utterance $i-2$
	Other Features (QM)
2	Question mark

Table 1: Features investigated (adapted from (Gupta et al., 2007)). N indicates the number of possible values (there are 16 DA tags).

Features	Accuracy
Baseline	57.9%
Sent + POS + QM	63.0%
DA	71.9%
Sent + POS + QM + Q_DA	70.6%
Sent + POS + QM + DA	75.1%

Table 2: CRF results: generic versus referential

4.1 Results & Discussion

A dominant class baseline on this data gives an accuracy of 57.9% (see Table 2). Our best set of features achieve an accuracy of 75.1% (see Table 2).

Our automatically extracted features (sentential, part of speech and question mark) achieve an accuracy of 63% which is above the baseline. Adding oracle dialog act information increases accuracy to 75.1%; substituting the more realistic Q_DA feature gives a smaller improvement, resulting in 70.6%. Note that accuracy is lower than the 84.4% achieved for two-person data, suggesting that referentiality in multi-party meetings is a harder task.

5 Reference Resolution

For referential cases, we must now identify the reference of *"you"* – in other words, the addressee. As our interest is in resolving *"you"*, we investigate this only for the referential utterances as marked by our annotators (not for all utterances). The AMI corpus has 4 meeting participants for each meeting. As 2-person subgroup addressing is rare (see above), we can model the problem as a four way classification task for each utterance – each of the 3 other participants and the entire group.

Since we have multiple meetings with possibly different participants, it makes little sense to index potential addressees by their real-world identity. Instead, for a given utterance, the potential addressee to speak next gets a label of 1; the other two are given labels of 2 and 3 based on the order in which they next speak. We use a label of 4 to represent addressing to the entire group.

Baseline. We can build two baselines. The *Next Speaker* baseline always predicts the addressee to be the next (different) speaker (i.e. a label of 1). The *Previous Speaker* baseline predicts the addressee to be the most recent previous different speaker.

Features. We expect that the structure of the dialog gives the most indicative cues to addressee: forward-looking dialog acts are likely to influence the addressee to speak next, while backward-looking acts might address a recent speaker. We therefore use similar features to those of Galley et al. (2004) for the related task of identifying the first half of an adjacency pair. However, since their task was retrospective, their features all involve facts about the previous discourse context. We therefore adapt the approach to examine features of subsequent as well as preceding utterances.

For each utterance and potential addressee, we examine the pair made up of the original utterance A and the next (or previous) utterance B spoken by that potential addressee. We then extract features of the pair which might indicate the degree of relatedness of the utterances, including their overlap, separation and lexical similarity, as shown in Table 3.

We also added a feature for the number of speakers that talk during the next 5 utterances to allow for better prediction of group addressing. In addition we included the features from Table 1, to test whether the features found useful for generic vs. referential disambiguation would be useful for the task of addressee detection.

Structural Features
. number of speakers between A and B
. number of utterances between A and B
. number of utterances of speaker B between A and B
. number of speakers that talk during the next 5 utterances
. do A and B overlap?
Durational Features
. duration of A
. if no overlap, time separating A and B
. if overlap, duration of overlap
. time of overlap with previous speaker
. time of overlap with next speaker
. speech rate of A
Lexical Features
. number of words in A
. number of content words in A
. ratio of words in A that are also in B
. ratio of words in B that are also in A
. number of cue words (Hirschberg and Litman, 1993) in A

Table 3: Features for addressee identification adapted from (Galley et al., 2004). We obtain a set of backward looking (BL) and forward looking (FL) features for an utterance.

Features	Accuracy
Baseline: Previous Speaker	23.0%
Baseline: Next Speaker	37.0%
FL + BL + Table 1	47.2%

Table 4: Addressee detection results.

Results & Discussion A CRF trained using all our features achieves an accuracy of 47.2%, which is a significant improvement on the baseline. Table 4 presents all the results.

The biggest confusion was found to be between utterances being classified as 1 or 4 (i.e. the next speaker or the entire group). Future work will therefore involve selecting features which can better discern between these two classes.

6 Conclusion

For generic vs. referential *you* disambiguation, our approach developed on two-party data transfers reasonably well to multi-party data. While accuracy is lower, it is significantly above the baseline. Use of a sequence model classifier has allowed us to operate without oracle context features, and a reduced dialog act tagset (question identification) provides reasonable (though reduced) accuracy. A next step here could be to use automatically classified dialog act tags.

Addressee detection is a hard problem, but we have shown promising results. We expect that investigation of further features, potentially including video information, will improve performance.

References

M. Galley, K. McKeown, J. Hirschberg, and E. 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)*.

S. Gupta, M. Purver, and D. Jurafsky. 2007. Disambiguating between generic and referential "you" in dialog. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*.

J. Hirschberg and D. Litman. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530.

N. Jovanovic, R. op den Akker, and A. Nijholt. 2006a. A corpus for studying addressing behaviour in multi-party dialogues. *Language Resources and Evaluation*, 40(1):5–23.

N. Jovanovic, R. op den Akker, and A. Nijholt. 2006b. Addressee identification in face-to-face meetings. In *Proceedings of the 11th Conference of the European Chapter of the ACL (EACL)*.

D. Jurafsky, A. Bell, and C. Girand. 2002. The role of the lemma in form variation. In C. Gussenhoven and N. Warner, editors, *Papers in Laboratory Phonology VII*, pages 1–34. Mouton de Gruyter.

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 ICMI*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*.

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 Measuring Behavior 2005, the 5th International Conference on Methods and Techniques in Behavioral Research*.

C. Müller. 2006. Automatic detection of nonreferential *It* in spoken multi-party dialog. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

M. Purver, P. Ehlen, and J. Niekrasz. 2006. Detecting action items in multi-party meetings: Annotation and initial experiments. In *MLMI 2006, Revised Selected Papers*.

SIDGrid: A Framework for Distributed, Integrated Multimodal Annotation, Archiving, and Analysis

Gina-Anne Levow, Sonjia Waxmonsky
Department of Computer Science
University of Chicago
levow,wax@cs.uchicago.edu

Bennett Bertenthal
Department of Psychology,
Indiana University, and
Computation Institute
University of Chicago
bbertent@indiana.edu

David McNeill
Department of Psychology
University of Chicago
dmcneill@uchicago.edu

Mark Hereld, Sarah Kenny, Michael E. Papka
Computation Institute
The University of Chicago
m-papka,m-hereld,skenny@uchicago.edu

Abstract

The SIDGrid architecture provides a framework for distributed annotation, archiving, and analysis of the rapidly growing volume of multimodal data. The framework integrates three main components: an annotation and analysis client, a web-accessible data repository, and a portal to the distributed processing capability of the TeraGrid. The architecture provides both a novel integration of annotation, analysis, and search for multimodal data and a powerful framework for web-based, distributed collaborative annotation and analysis. The flexibility and capabilities of the system have been demonstrated through archiving Talkbank and other spoken discourse and dialogue data and performing joint multimodal analysis of lexical, prosodic, turn-taking, and other multimodal factors.

1 Introduction

Recent research programs in multimodal environments, including understanding and analysis of multi-party meeting data and oral history recording projects, have created an explosion of multimodal data sets, including video and audio recordings, transcripts and other annotations, and increased interest in annotation and analysis of such data. However, multimodal data poses particular challenges including a broad range of annotation and analysis measures, large storage requirements for media data, and increased computational complexity of media data and multi-factor analyses. Furthermore, since this data is costly to collect and annotate, both in terms of time and money, there is additional incentive to share data and collaborate on annotation efforts. The wide range of annotations, from aligned transcripts to gaze to reference to gestural form, often leads to annotation by multiple expert groups, possibly geographically distributed, to fully exploit these resources.

A number of systems have been developed to manage and support annotation of multimodal data, including Annotation Graphs (Bird and Liberman, 2001), Exmeralda (Schmidt, 2004), NITE XML Toolkit (Carletta et al., 2003), Multitool (Allwood et al., 2001), Anvil (Kipp, 2001), and Elan (Wittenburg et al., 2006).

The framework described here, developed under the NSF Cyberinfrastructure Program, aims to extend the capabilities of such systems by focusing on support for large-scale, extensible distributed data annotation, sharing, and analysis. The system is open-source and multi-platform and based on existing open-source software and standards. The system greatly eases the integration of annotation with analysis though user-defined functions both on the client-side for data exploration and on the TeraGrid for large-scale distributed data processing. A web-accessible repository supports data search, sharing, and distributed annotation. While the framework is general, analysis of spoken and multi-modal discourse and dialogue data is a primary application.

The details of the system are presented below. Sections 2, 3, and 4 describe the annotation client,

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 231–234,
Antwerp, September 2007.

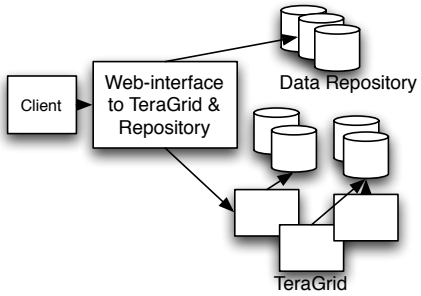


Figure 1: System Architecture

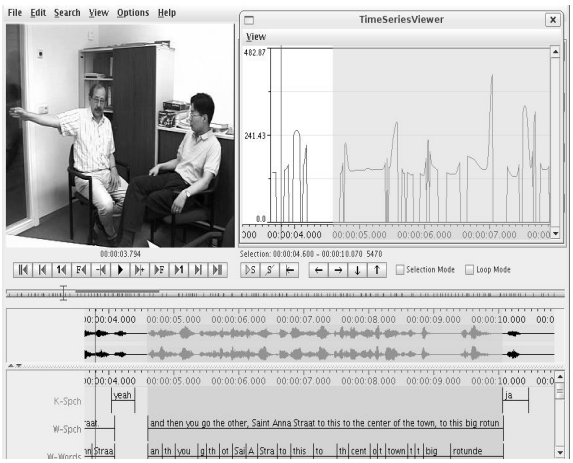


Figure 2: Screenshot of the annotation client interface, with video, time-aligned textual annotations, and time series displays.

the web-accessible data repository, and the portal to the TeraGrid, respectively, as shown in Figure 1 below. Section 6 describes system availability and planned extensions to system functionality.

2 The SIDGrid Client

The SIDGrid client provides the primary interactive multimodal annotation interface. A screenshot appears in Figure 2. The client extends the open-source ELAN annotation tool from the Max Planck Institute[1]. ELAN supports display and synchronized playback of multiple video files, audio files, and arbitrarily many annotation "tiers" in its "music-score"-style graphical interface. The annotations are assumed to be time-aligned intervals with, typically, text content; the system leverages Unicode to provide multilingual support. Time series such as pitch tracks or motion capture data can be displayed synchronously. The user may interactively add, edit, and do simple search in annotations. For example, in multi-modal multi-party spoken data, annotation tiers corresponding to aligned text transcriptions, head nods, pause, gesture, and reference can be created.

The client expands on this functionality in two main ways. First, the system allows the application of user-defined analysis programs to media, time series, and annotations associated with the current project, such as a conversation, to yield time series files or annotation tiers displayed in the client interface. Any program with a command-line or scriptable interface installed on the user's system may be added to a pull-down list for invocation. For example, to support a prosodic analysis of spoken dialogue data, the user can select a Praat (Boersma, 2001) script to perform pitch or intensity tracking. Currently a variety of Praat, R, and Matlab scripts are supported, and topic segmentation and reference resolution algorithms are being integrated. Also, the client provides integrated import and export capabilities for the central repository. New and updated experiments and annotations may be uploaded directly to the archive from within the client interface. Existing experiments may be loaded from local disk or downloaded from the repository for additional annotation.

3 The SIDGrid Repository

The SIDGrid repository provides a web-accessible, central archive of multimodal data, annotations, and analyses. This archive facilitates distributed annotation efforts by multiple researchers working on a common data set by allowing shared storage and access to annotations, while keeping a history of updates to the shared data, annotations, and analysis.

The browser-based interface to the archive allows the user to browse or search the on-line data collection by media type, tags, project identifier, and group or owner. A simple permission scheme, based on Unix-style group permissions, provides public access to freely available data while restricting access to more sensitive data to authorized users. Once se-

[1] http://www.mpi.nl/tools/elan.html

lected, all or part of any experiment may be downloaded. In addition to lists of experiment names or thumbnail images, the web interface also provides a streaming preview of the selected media and annotations, allowing verification prior to download. (Figure 3)

The repository also supports import of new data. To support interoperability with other annotation tools, conversion functions have been developed for a range of annotation formats, in collaboration with developers[2], using Annotation Graphs as an interchange format, in addition to the existing ELAN-based import capabilities.

All data is stored in a MySQL database. Annotation tiers are converted to an internal time-span based representation, while media and time series files are linked in unanalyzed. This format allows generation of ELAN format files for download to the client tool without regard to the original source form of the annotation file. The database structure further enables the potential for flexible search of the stored annotations both within and across multiple annotation types.

4 The TeraGrid Portal

The large-scale multimedia data collected for multimodal research poses significant computational challenges. Signal processing of gigabytes of media files requires processing horsepower that may strain many local sites, as do approaches such as multi-dimensional scaling for semantic analysis and topic segmentation. To enable users to more effectively exploit this data, the SIDGrid provides a portal to the TeraGrid (Pennington, 2002), the largest distributed cyberinfrastructure for open scientific research, which uses high-speed network connections to link high performance computers and large scale data stores distributed across the United States. While the TeraGrid has been exploited within the astronomy and physics communities, it has been little used by the computational linguistics community.

The SIDGrid portal to the TeraGrid allows the user to specify a set of files in the repository and a program or programs to run on them on the Grid-based resources. Once a program is installed on the Grid, the processing can be distributed automatically

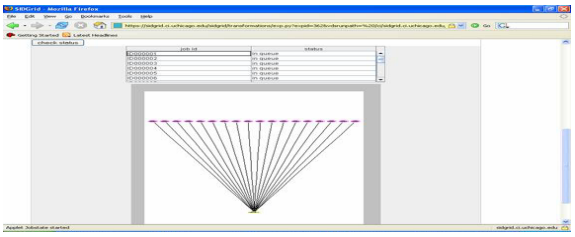


Figure 4: Progress of execution of programs on TeraGrid. Table lists file identifiers and status. Graph shows progress.

to different TeraGrid nodes. Software supports arbitrarily complex workflow specifications, but the current SIDGrid interface provides simple support for high degrees of data-parallel processing, as well as a graphical display indicating the progress of the distributed program execution, as shown in Figure 4. The results are then reintegrated with the original experiments in the on-line repository. Currently installed programs support distributed acoustic analysis using Praat, statistical analysis using R, and matrix computations using Matlab.

5 Prototype Use

The system has been applied to spoken and multimodal discourse and dialogue data ranging from recordings and annotations of multi-party interactions to oral history data to the Talkbank corpus[3], including child language data. This data served as a corpus for basic development of system capabilities. The developers converted the data from their original formats for integration into the repository. The publicly available Talkbank data, such as audio and video media files, can be viewed, browsed, and downloaded from the repository and manipulated in the client-side annotation tool. Prosodic extraction experiments have been performed both using the local client and on the Teragrid, using the dispatch procedures to concurrently analyze data and media files on widely distributed hardware resources. Pitch extraction processes, where analysis of a single file runs out of memory on 2GB, dual-processor Opteron machines, can be completed on 10 files in 1.5 hours with Grid-based servers. These tasks illustrate the scalability of large-scale computation-

[2] http://www.multimodal-annotation.org

[3] http://www.talkbank.org

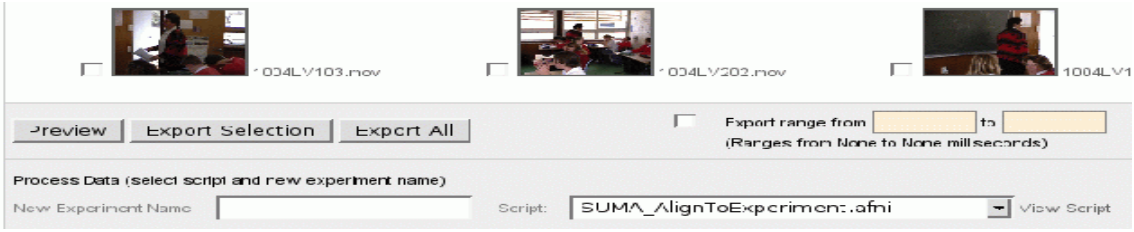


Figure 3: Screenshot of the archive download interface, with thumbnails of available video and download and analysis controls.

ally expensive analyses supported by the SIDGrid framework.

In addition, some preliminary experiments to assess multimodal search and analysis were conducted. These experiments considered the interaction of prosodic features, such as pitch, with other modalities such as gaze or head movement, within turns. These trials demonstrated the capability of search across multiple annotation tiers - including manual speech transcriptions and turn annotations - and time series data from pitch tracking.

6 Future Directions

The SIDGrid infrastructure provides a powerful and flexible environment for annotation, archiving, and analysis of multimodal data. The novel, extensible integration of annotation and analysis both in the client and in the Grid portal will support greater ease of data exploration and large-scale data analysis. The overall framework supports both local data access and distributed annotation and analysis via access to the repository and TeraGrid.

While the basic infrastructure developed thus far is already useful, many extensions to functionality are underway. A major focus is the enhancement of search functionality, for both data and meta-data search. We aim to support both aggregate search for text and annotations across sets of files in the repository under a range of user-specified constraints and search for images in the video recordings. Access to the SIDGrid software and systems is possible through http://sidgrid.ci.uchicago.edu. Future users of the system further will guide its development as new needs come to light.

Acknowledgments We thank other members of the SIDGrid group, including Rick Stevens, David Hanley, Kavithaa Rajavenkateshwaran, and Thomas Uram. This work was supported in part by NSF under Grant No. BCS-05-37849.

References

Jens Allwood, Leif Groenqvist, Elisabeth Ahlsen, and Magnus Gunnarsson. 2001. Annotations and tools for an activity based spoken language corpus. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, pages 1–10.

S. Bird and M. Liberman. 2001. A formal framework for linguistic annotation. *Speech Communication*, 33(1,2):23–60.

P. Boersma. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9–10):341–345.

J. Carletta, S. Evert, U. Heid, J. Kilgour, J. Robertson, and H. Voormann. 2003. The NITE XML Toolkit: flexible annotation for multi-modal language data. *Behavior Research Methods, Instruments, and Computers, special issue on Measuring Behavior*, 35(3):353–363.

M. 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.

Rob Pennington. 2002. Terascale clusters and the TeraGrid. In *Proceedings for HPC Asia*, pages 407–413. Invited talk.

T. Schmidt. 2004. Transcribing and annotating spoken language with EXMARaLDA. In *Proceedings of the LREC-Workshop on XML-based richly annotated corpora*.

P. Wittenburg, H. Brugman, A. Russel, A. Klassmann, and H. Sloetjes. 2006. Elan: a professional framework for multimodality research. In *Proceedings of Language Resources and Evaluation Conference (LREC) 2006*.

ScIML: Model-based Design of Voice User Interfaces

Jörn Kreutel
University of Potsdam,
Applied Computational Linguistics Lab
kreutel@ling.uni-potsdam.de

Abstract

We will introduce ScIML, a domain specific language for voice user interface (VUI) creation that is based on the generic expressive means of the Unified Modelling Language. In particular, we employ UML statecharts for interaction flow modelling.

1 Introduction

In the course of the last decade and beyond, significant research has been carried out in the field of dialogue management, in general, and spoken dialogue systems, in particular (see, e.g., (Traum, 1996; Seneff et al., 1998; Larsson et al., 1999; Bohus and Rudnicky, 2003). On the other hand, available approaches and technology for developing commercial dialogue systems – which we will term as *Voice User Interfaces* (VUIs) throughout this paper – have so far not advanced beyond the simple form-filling mechanism underlying VoiceXML (Oshry, 2004). The latter, at the same time, exhibits a severe lack of modularisation as far as a separation of concerns between dialogue management, on the one hand, and prompt and grammar creation, on the other, is concerned.

Missing transformation achievements between research and the 'voice industry' may partially be due to the fact that the latter strongly requires a *visual* representation format for VUIs that makes transparent the functionality of a VUI to all stakeholders in a project, be it technical or business staff. In addition, industry projects require that any aspect of a VUI, in particular its interaction flow, be principally subject to particular *design* decisions, i.e. it needs to be *hand-craftable*. Both requirements, however, are outside the primary scope of research on dialogue management. In fact, the concept of a generic dialogue management component which implements a range of domain independent interaction routines may even be seen as marking a contrary position, as long as its functionality is not foreseen to be at least overridable by domain specific implementations.

Given these findings, this paper will outline the basic ideas underlying the *Scene based Interaction Modelling Language* (ScIML),[1] which approaches the issue of voice user interface creation from the perspective of *model based user interface design*. The expressive means of ScIML are a range of VUI-specific concepts that are formalised as extensions of the UML meta model (Group, 2004), whose visualisations are well established within the IT industry. In particular, ScIML employs UML statecharts (Harel, 1987) for dialogue management purposes, i.e. it relies on a generic formalism for describing the behaviour of complex event-driven systems.

Methodologically, ScIML adheres to an account of user interface modelling that is known as OO&HCI (*Object Oriented Modelling and Human Computer Interaction*). It conceives of UI design as involving a range of different interrelated modelling activities. ScIML adopts this approach and supports the respective activities through appropriate expressive means which exploit the current state of the art in dialogue systems research. In particular it employs the two basic concepts of *dialogue acts* (Poesio and Traum, 1998; Bunt and Girard, 2005) and of

[1] The ScIML notion of *scene* is based on proposals for GUI modelling in (de Paula, 2002).

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 235–238,
Antwerp, September 2007.

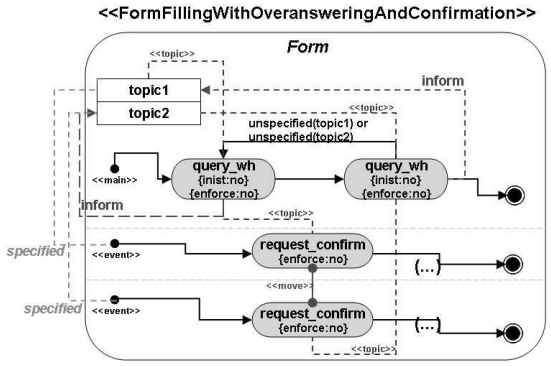


Figure 1: Integrated representation format for ScIML, depicting, main and event workflows, topicality relations, non-local response spaces and move formation

grounding (Clark and Brennan, 1991; Matheson et al., 2000).

2 ScIML Modelling Activities and Artefacts

ScIML VUI Referent Models provide, for each activity of an underlying **Task Model**[2], a structured description of the entities that are addressed by the VUI and/or the user in the course of the realisation of that activity. Referent models, hence, specify the potential *topics* of conversation for some VUI. The ScIML meta model assumes three abstract referent types: *activity*, *entity* and *event*, which are concretised as *domain activity* and *VUI activity*, *domain entity* and *VUI entity* and *domain event* and *VUI event*, respectively. Examples for the latter are, e.g., failures of speech recognition or missing inputs on the part of the user.

Over the set of referents of a ScIML referent model, we further assume the relations of *occurrence* between events and activities, and the one of *involvement* that specifies associations between entities, on the one hand, and activities or events, on the other, as well as between two activities. We further assume that entities may be *complex*, which is reflected by a *constituency* relation between entity referents. Activities and events, on their part, will be considered as *complex referents* by nature. Methodologically, these relations serve as a starting point for referent identification on the basis of a task model. Assuming that each activity of the latter corresponds to an activity referent in the referent model, it is possible to determine both the entities that are involved in some activity and the events that may occur in it.

For authoring a VUI referent model, we use UML class diagrams whose classes and associations are *profiled* on the basis of the assumed referent types and relations between them.

ScIML Interaction Structure Models define the *topical structure* of interactions over some given VUI referent model. For this purpose, an interaction structure model identifies, first of all, a set of *scenes* which can be conceived of as topically coherent contexts that span over sequences of *moves* by the user and the VUI. *Moves*, on their part, are modelled as sets of *dialogue acts*.[3] Both for scenes and for dialogue acts, the referents that serve as their respective topics are provided by the constituents of the VUI referent model. For *scenes*, it is additionally required that their topics be *complex* referents. Interaction structure models, further, describe which *domain functions* for accessing backend data and executing transactions are required for each scene's realisation.

Interaction structure models are authored as class diagrams that define the *topicality* association between the members of the referent model and the assumed *scenes* and *dialogue acts*. They further specify the association between dialogue acts and those scenes in whose realisations the acts are involved. Note that interaction structure models are *structural* models in the sense that they merely define the set of scenes for some VUI application, as well as, for each scene, the set of dialogue acts that may be involved in its realisation. Both the actual dialogue flow by

[2] ScIML Task Models describe, at a coarse-grained level, the *activities* that are actually or supposedly – from a user's perspective – supported by the VUI. Activities may be either *domain activities* or *VUI activities*, and are structured in the sense that some activity may involve the realisation of a range of subactivities.

[3] ScIML's notion of *dialogue acts* is based on the idea that for the purpose of VUI modelling, dialogue acts can be described by a generic *dialogue act type* and a domain specific referent that identifies the *topic* of the act. This proposal withdraws from thinking of the *propositional content* of dialogue acts as rich semantic representations of system and user utterances. Instead, it assumes that the *illocutionary force* of dialogue acts – or their *update effect*, in other terms, see e.g. (Poesio and Traum, 1998) – operates on assignments of referent values.

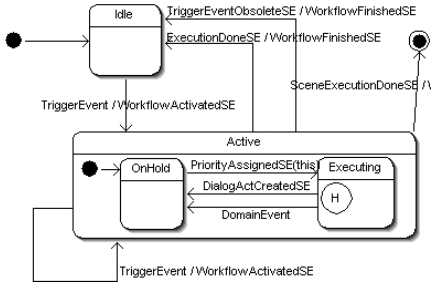


Figure 2: Generic Processing Model for a Workflow in ScIML

means of which a scene is realised and the clustering of elementary dialogue acts into *moves* is outside the scope of this model type.

ScIML interaction flow models define, for each scene of an interaction structure model, the realisation of this scene through a set of *event-triggered workflows*. A workflow is a sequence of state transitions between *VUI dialogue acts*, *domain functions* and *sub scenes*. In this approach, dialogue acts performed by the user are conceived of as a particular type of *event* that may trigger the initialisation of a workflow or a state transition within some active workflow.

The concepts underlying this type of model are described, in more detail, in the following section.

ScIML presentation models comprise, on the one hand, a *VUI move model* that defines *patterns* of VUI dialogue acts that constrain the move formation on the part of the VUI. On the other hand, they include the definition of a *response space model*. The latter assigns, for each occurrence of a VUI dialogue act in the interaction flow model, a set of user moves that are assembled, on their part, from dialogue acts. Presentation models are authored within an *integrated representation format* for ScIML that is exemplified by figure 1.

3 Statecharts-based interaction flow Modelling

Given our notion of *scene* as the domain with regard to which interaction flow will be specified, the purpose of an interaction flow model is to determine, for each realisation of a scene, whether it is in one of the following *activity states*:

- the performance of a dialogue act
- the performance of some domain function for backend data access or transaction execution
- the realisation of some subscene

For dialogue act states, ScIML assumes that the corresponding dialogue act will only be realised if its preconditions hold and as long as they *do* hold. This way, e.g., a *form filling algorithm* can be reconstructed – as in figure 1 – by a sequence of query_wh dialogue acts which will only be realised if their respective referents have not been specified before. Thus, ScIML's notion of dialogue act states is able to account for dialogue flow phenomena like Overanswering.

However, rather than specifying the control algorithm for a scene as a single FSM, we propose to think of it as being described by a *set* of *workflows* that define transitions over the above states and that are *triggered by events* of the following types:

- *VUI Events*, which indicate a failure to recognise a user's input, the missing of input by the user, or any other exceptional behaviour particular to the usage of a VUI.
- *Dialogue Act Events*, which signal the performance of a dialogue act by the user.
- *Grounding Events*, which express a change of the grounding status of some referent. Grounding events will be caused, on their part, by the performance of dialogue acts.[4]
- *Domain Events*, which may be thrown during the execution of some domain function. The call of a domain function may, on its part, be triggered by the occurrence of a grounding event, e.g. a specified event with regard to some referent or set of referents.

For each scene there will, further, be one *main workflow* that will be triggered upon entering the scene and that describes, e.g., a *form-filling* flow.

[4] We assume that a referent's grounding status may be either unspecified, specified, i-grounded, c-grounded, i-rejected or c-rejected. The notions of i-grounded and i-rejected, on the one hand, and c-grounded and c-rejected, on the other, reflect the two dimensions of *reliability* with regard to a user's intention and *validity* with regard to the given application domain.

Note, however, that triggering of a workflow may not immediately result in executing it. Instead, there is, additionally, a generic *workflow prioritisation* algorithm that determines the ordering in which workflows will be executed. E.g., in case a trigger event for some workflow occurs outside the context of the corresponding scene, this workflow will, further, generically be prioritised over the scene's main workflow. Particularly in voice portals that offer a variety of services under single entrance point, these processing routines allow that a user may not only identify a desired service, but may also provide more information with regard to the latter's referents.

As a complementary process to workflow triggering, SciML allows to discard workflows that have been made obsolete by events that occurred after they have been initialised. With regard to *workflow obsoletion*, SciML assumes that obsoletion conditions can be derived from the triggering conditions of a workflow.[5]

For authoring, SciML employs an abbreviation of the actual statechart representation. Using statecharts in their particular version as UML *activity diagrams*, only the particular flow inside the generic workflow execution model – i.e. the content of the Executing state – will be explicitly authored. As figure 1 shows, these workflows will be specified in the *parallel* regions of a scene state.

4 Outlook

The SciML execution model described in the previous section has been verified on the basis of the existing Apache reference implementation of an SCXML interpreter. SCXML is an XML syntax for statecharts and has recently been proposed by the W3C (Auburn et al., 2005) as a standard for UI interaction flow control. It is also meant to enhance VoiceXML towards the creation of more flexible 'advanced' VUIs. However, statecharts lack expressive means for this particular purpose. SciML, in contrast, shows how statecharts can be intuitively *profiled* for VUI modelling and thus means to contribute to the uptake of SCXML in the voice industry.[4]

[5] For example, if an i-rejected grounding event occurs for any referent value involved in a workflow trigger, or if a specified event specifies an alternative value for the latter, the affected workflows will be obsolete. This will not be the case, however, if the workflow contains a subscene state that, on its part, specifies a workflow for the respective event.

References

RJ Auburn, Jim Barnett, Michael Bodell, and T.V. Raman. 2005. State Chart XML (SCXML) scate machine notation for control abstraction. Working draft, W3C.

Dan Bohus and Alex Rudnicky. 2003. Ravenclaw. In *Proceedings of the Eighth European Conference on Speech Communication and Technology (Eurospeech 2003), Geneva, Switzerland*.

Harry Bunt and Yann Girard. 2005. Designing and open, multidimensional dialogue act taxonomy. In *Proceedings of Dialor 2005, the 9th Workshop on the Semantics and Pragmatics of Dialogue*. LORIA, Nancy/France, June 2005.

Herbert H. Clark and Susan E. Brennan. 1991. Grounding in communication. In L. B. Resnick, J. Levine, and S.D. Teasley, editors, *Pespectives on Socially Shared Cognition*. APA.

Maira Greco de Paula. 2002. Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica: Construção de um modelo de interação. Msc thesis, Pontificia Universidade Catolica do Rio de Janeiro.

Object Management Group. 2004. UML 2 Meta Model. Specification, OMG.

David Harel. 1987. Statecharts: A visual approach to complex systems. *Science of Computer Programming*, 8:231–274.

Staffan Larsson, Peter Bohlin, Johan Bos, and David Traum. 1999. Trindikit 1.0 manual. TRINDI Deliverable 2.2, University of Göteborg.

Colin Matheson, Massimo Poesio, and David Traum. 2000. Modelling grounding and discourse obligations using update rules. In *NAACL*.

Matt Oshry. 2004. Voice Extensible Markup Language (VoiceXML) 2.1. Recommendation 2.1, W3C consortium.

Massimo Poesio and David Traum. 1998. Towards an axiomatisation of dialogue acts. In *Twente Workshop on Language Technology*.

Stephanie Seneff, Ed Hurley, Raymond Lau, Christine Pao, Philipp Schmid, and Victor Zue. 1998. Galaxy-ii: A reference architecture for conversational system development. In *Proceedings of ICSLP 98*.

David Traum. 1996. Conversational agency: The trains-93 dialogue manager.

Tutoring in a Spoken Language Dialogue System

JAAKKO HAKULINEN, MARKKU TURUNEN, and KARI-JOUKO RÄIHÄ
Tampere Unit for Computer-Human Interaction
Department of Computer Sciences
33014 University of Tampere, Finland
Firstname.lastname@cs.uta.fi

Abstract

We have developed interactive multimodal software tutors to teach users how to use a spoken dialogue timetable by guiding users and monitoring their interaction. They feature a visual representation of the spoken dialogue to support error recognition and recovery and thus helping the users to learn the required interaction style. Two different versions of tutoring were compared to a static web manual in a between-subjects experiment (N=27).

1 Introduction

The challenges of designing spoken dialogue systems are well known, as are the usual solutions. How do the users know the functionality provided by a speech-based system? How do they know when to speak and what to say to the system? A well designed speech interface supports the users' natural way of speaking. However, in practice the interface must also guide users to speak in a way that the system is able to understand. Implicit and explicit prompts, hints, and tapering embed the guidance in the spoken interaction. (Yankelovich, 1996) When a system is used repeatedly, it is plausible that the users are willing to invest some effort into fully learning service.

How, then, are speech-based systems introduced to new users? When speech is an additional modality e.g. in the case of voice control systems in automobiles, the speech-based features can be described in the owner's manual or users can discover the voice control possibilities through the graphical part of the interface. Unimodal, telephone-based spoken dialogue systems need some auxiliary material to introduce them to the users. In addition to an introduction to the service, users are often provided with some instructions on how to use the system. Such a web-based tutorial can improve the user experience and users' perception of the system (Kamm, Litman, and Walker, 1998).

Another approach to introducing new applications to users is software tutoring. This is popular with graphical interfaces, particularly in video games, but it has been almost neglected in the case of speech-based applications. However, the tutorial type guidance can be embedded into a dialogue system, e.g., as a specific guided mode, which can make the system more transparent to users and thus help them, for instance, in knowing how to correct errors (Karsenty and Botherel, 2005). This kind of guidance can be extended by implementing a software tutor, a separate dialogue partner, which not only guides users but also monitors their interaction and makes sure that the users indeed learn to use the system. We have implemented such a tutor and found it reduced the amount of problems users have during the learning period (Hakulinen, Turunen, and Räihä, 2006).

Here we follow-up our previous work on unimodal tutoring by studying graphical tutoring in speech interface. The visual presentation can overcome the transient and linear nature of speech and its low output rate. The multimedia tutors are connected to the spoken dialogue system so that a user can try out the system under the supervision of the tutor. Different tutor concepts were developed (Hakulinen, Turunen, and Salonen, 2005) and two most promising ones were chosen for an experiment. The tutors introduce the spoken dialogue system to users, guide them through an elementary scenario, monitor users' interaction with a spoken

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 239–242,
Antwerp, September 2007.

dialogue system and provide guidance as necessary, for example, after recognition rejections.

We collected data on users' interaction with the tutor and the dialogue system and users' attitudes towards the guidance materials and the system. The data did not show significant differences in the task completion rates, but the most troublesome interactions occurred in the web guidance condition. The software tutor with more interaction possibilities was ranked highest in the subjective evaluations, while the other tutor was ranked the worst among the three conditions. Thus, the multimedia tutor can help in learning to interact with a spoken dialogue system, but only when designed properly. The graphical form used in the most interactive guidance helps users in understanding the functionality of the spoken dialogue system. The results point out the importance of constructing the guidance material in a manner that closely corresponds to the interaction model of the system: the interface is essentially a form-filling dialogue, and the highest ranked tutor is based on a graphical version of the form.

2 Guidance Materials

The tutors are graphical software applications run on a personal computer and they communicate with the spoken dialogue application running on a server. A web manual has been constructed based on the tutors by removing all interactivity and arranging the information into a static document. All material is in Finnish, figures and examples have been translated for the paper.

The spoken language dialogue system that the tutors guide users on is called Busman. It is a research prototype of a telephone-based service for Tampere area public transport timetables (Turunen et al. 2005). Typical utterances understood by the system include "Which line runs from University Hospital to the city center" and "When after six pm does a bus depart from Hervanta to university". The system uses form-based dialogue management. Implicit confirmations are used extensively and mostly the interaction is user initiative. System initiative prompts are used for obtaining missing information and after repeated error situations. A short and a rather exhaustive spoken help messages can be heard by giving respective commands.

The system uses the Finnish language ASR (Philips SDK with unisex Finnish acoustic models, about 1500 words per grammar) and TTS(Mikropuhe by Timehouse). The system does not support barge-in but telephone keypad can be used to interrupt the system.

2.1 Tutor Design

The goal of the tutors is to introduce the Busman system to new users and teach them how to interaction with it. In five to ten minutes, users will learn the functionality of the system and use it by following the instructions given by the tutor.

The tutors were presented to users as application windows as can be seen in Figures 1 and 2. The only aural component in the tutors is a notification sound that directs users' attention from the application context to tutoring when necessary.

The snapshot of Balloon tutor during the hands-on exercise part can be seen in Figure 1.

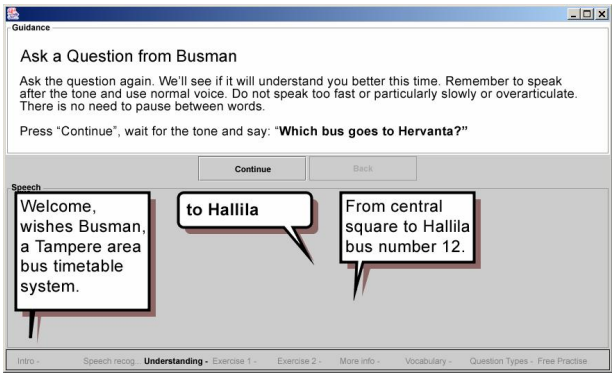


Fig. 1: A screenshot of the Balloon tutor.

The Form tutor includes all the functionality of the Balloon tutor. And a form consisting of graphical user interface components, which users can use to create queries that can be asked from the Busman system. The GUI form can be seen as a visual representation of the timetable system. The benefit of a graphical form is based on a finding by Terken and teRiele (2001) that a multimodal interface with a graphical query interface provided a mental model that can be useful with a speech only interface. The Form tutor is shown in Figure 2.

Guidance in both tutors is organized similarly into six segments, each consisting of one text screen. In addition, there is a hands-on exercise in the middle of the tutoring where users try out Busman under the supervision of the tutor. This part consists of calling Busman and making three

queries. In the end, there is free experimentation while the tutor is still active. The last text segment before the free experimentation is a summary.

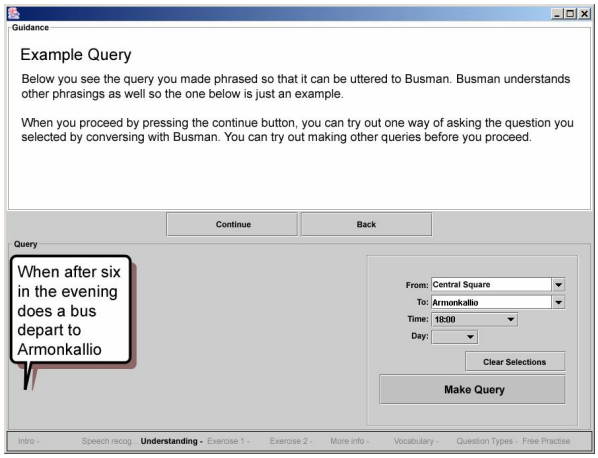


Fig. 2: A screenshot of the Form tutor.

Speech balloons are used to visualize spoken dialogue, i.e., speech recognition results and system outputs, both during the hands-on exercise and the free experimentation. They are also used to display an example dialogue before the exercise. The balloons use bold face font to emphasize keywords in user utterances. Furthermore, the balloons provide a short dialogue history.

The tutors guide the users step by step during the hands-on exercise. The users are told exactly what to say when they call the timetable system for the first time. The tutors monitor the speech recognition results for errors and by comparing ASR results to the requested input, the tutors can spot errors with certainty but not deduct their reason. They do not guess but provide guidance on how to remedy the situation. If the recognition results do not match the required input closely enough, help is given, and the user is asked to try again maximum three times, simplifying the requested input if some information has already been given successfully. The help provided includes instructions on how to speak, such as to use normal voice and talk after a tone. By pointing out errors and providing relevant guidance, the tutors can help users in learning to detect, diagnose, and correct errors.

In addition to the two tutors, a web based version of the same material was created. It contains the same texts and graphics as the tutors as far as possible.

3 Experiment

There were three conditions (named web, balloon and form), one for each guidance material, with 9 participants each. Age of participants ranged from 16 to 41 years with an average of 26. Ten of them were male and 16 female. Most of them had never used spoken dialogue systems and the remaining had had random usage. Participant's computer using skill ranged from inexperienced user to active hobbyist, most being common users. There were no significant differences on background variables between the conditions. The participants received a movie ticket for their participation. They were randomly assigned to the conditions.

The test consisted of a 15 minute learning period with the guidance and a 15 minute period for working with a set of 11 tasks without the guidance material. In the end, participants filled in two questionnaires where the timetable system and the guidance material were evaluated.

A SASSI questionnaire (Hone and Graham 2000) was used to gather opinions on the Busman timetable system. A set of questions developed by Hassenzahl et al. (2000) was used to gather opinions on the guidance. Both used seven-item Likert-scale questions and an additional field for open comments. The guidance questionnaire also included scales on the length, amount, and consistency of guidance. The questions were in Finnish.

In the tasks the participants were asked to find a bus line number for a given route and a departure time that was near a given time.

3.1 Results

Task completion rates were similar in all conditions. The telephone calls reveal a wider variety of error rates in the Web condition. Questionnaires and general observations made during the experiments raise the Form tutor as the most highly ranked guidance type and provide some insights into differences between different kinds of users.

Interaction with the System Users' interaction under the guidance of tutors seems to be more consistent while some users of a static manual do just fine and others have serious problems. While there were no statistically significant differences in the error rates between the conditions, the variances of utterance level error rates (i.e., percentage of utterances that did not result in correct system re-

sponse) between the three conditions were significantly different (Bartlett test of homogeneity of variances, df = 2, $p < 0.05$). The Web condition had the highest variance in error rates while the Balloon condition had the lowest. When the training part, i.e., the direct effect of the condition is removed, and only the interaction during the tasks is considered, the error rate distributions become more similar.

Questionnaires The guidance evaluation questionnaire resulted in different overall evaluations for the guidance materials. The differences are highly significant (Friedman rank sum test (of evaluation medians), df = 2, $p < 0.001$). Rank sums (higher value – better evaluation) were 55.5 for the Web condition, 39.5 for the Balloon condition, and 73.0 for the Form condition. There were no statistically significant differences between the conditions within single guidance evaluation questions.

There was no significant difference between the conditions on the SASSI evaluation of the Busman system. However, participants' backgrounds correlate with some evaluations. Computer skills is a variable that highly significantly correlated (Pearson's product-moment correlation df = 25, $p < 0.01$) with answers to five questions. In all cases more experienced computer users considered the timetable system worse, i.e., less pleasant and more irritating. Speech user interface experience correlates also with computer skills (Pearson's product-moment correlation, df = 25, $p < 0.05$). However, computer skills did not correlate with error levels or task completion rates. Furthermore, the correlations of computer skills were only with system evaluations. There was no significant correlation with the guidance evaluations, which suggests that the tutors, while not equally necessary to, were equally accepted by the different users.

In guidance questions age correlated (Pearson's product-moment correlation, df = 25, $p < 0.001$) negatively with answers to the question "Guidance was too long", i.e., younger participants considered the guidance too long more often than older ones.

4 Discussion

In this study, we compared different guidance materials to teach to use of a spoken dialogue system. The results indicate that interactive tutoring helps especially those people, who would have most problems learning the use with static guidance materials. While some users can learn to use a system just fine with just a static manual or even without any guidance, others have many problems in learning the style of interaction required in human-computer spoken dialogue. Unlike static guidance, tutors were able to take care of all users. It is worth mentioning, that especially those, who felt more insecure on using the system, reported that they felt comfortable when they received support from the tutor in the beginning. Tutoring can support users who could not learn the system otherwise, but not all users should be forced to use one.

5 References

Jaakko Hakulinen, Markku Turunen, and Kari-Jouko Räihä 2006. Evaluation of Software Tutoring for a Speech Interface. *International Journal of Speech Technology*, 8, 3, 283-293.

Jaakko Hakulinen, Markku Turunen, and Esa-Pekka Salonen. 2005. Software Tutors for Dialogue Systems. *Proceedings of Text, Speech and Dialogue, LNAI 3658*, Springer, 412-419.

Marc Hassenzahl, Axel Platz, Michael Burmester, and Katrin Lehner, 2000. Hedonic And Ergonomic Quality Aspects Determine a Software's Appeal. *Proceedings of CHI2000*, ACM Press, 201-208.

Kate Hone, and Robert Graham, 2000. Towards a Tool For The Subjective Assessment of Speech System Interfaces (SASSI), *Natural Language Engineering*, 6, 3 & 4, September 2000.

Candace Kamm, Diane Litman, and Marilyn Walker, 1998. From Novice to Expert: The Effect of Tutorials on User Expertise with Spoken Dialogue Systems. *Proceedings ICSLP*, ASSTA, 1211-1214.

Laurent Karsenty, and Valérie Botherel, 2005. Transparency Strategies to Help Users Handle System Errors. *Speech Communication*, 45, Pp. 305–324.

Jacques Terken, and Saskia te Riele, 2001. Supporting the Construction of a User Model in Speech-Only Interfaces by Adding Multi-Modality. *Proceedings of Eurospeech 2001 Scandinavia*, ISCA, 2177-2180.

Markku Turunen, Jaakko Hakulinen, Esa-Pekka Salonen, Anssi Kainulainen, and Leena Helin, 2005. Spoken and Multimodal Bus Timetable Systems: Design, Development and Evaluation. *Proceedings of SPECOM 2005*, 389-392.

Nicole Yankelovich. 1998. How Do Users Know What To Say? *Interactions*, 3, 6, 32-43.

Using Speech Acts in Logic-Based Rhetorical Structuring for Natural Language Generation in Human-Computer Dialogue

Vladimir Popescu[1,2], **Jean Caelen**[1], **Corneliu Burileanu**[2]
[1]Grenoble Institute of Technology, France
[2]University "Politehnica" of Bucharest, Romania
{vladimir.popescu, jean.caelen}@imag.fr

Abstract

Usually, human-computer dialogue systems rely on ad-hoc solutions for the component performing speech turn generation, in natural language. However, integration of task-specific and general world knowledge in order to provide a more reliable and natural interaction with humans also through more sophisticated language generation techniques becomes needed. In this paper we present performance improvements of a module simulating in first-order logic Segmented Discourse Representation Theory for language generation in dialogue. These improvements concern reductions in computational costs and enhancements in rhetorical coherence for the discourse structures obtained, and are obtained using speech-act related information for driving rhetorical relations computations.

1 Introduction

Most human-computer dialogue systems rely on handcrafted, usually template-based, language generation modules for producing machine's utterances (McTear, 2002). However, in the last decade or so, with the emergence of research results and ideas from the multi-agent systems domain dialogue systems became more sophisticated, aiming at better responses to user's requests, via a greater naturalness and relevance of the speech turns produced, in relation to the context of the dialogue and to the users involved (Caelen and Xuereb, 2007), (McTear, 2002). Hence, the natural language generation component itself should aim towards more contextualized and pragmatically situated language productions, involving consideration of rhetorical and actional aspects of language production. In this context, two research trends became distinguishable: (i) a rhetoric-based approach, using formal accounts of discourse originally designed for language interpretation: thus, theories such as Rhetorical Structure Theory or, more recently, Segmented Discourse Representation Theory - SDRT (Asher and Lascarides, 2003) have been adopted for natural language generators (Danlos *et al.*, 2003); (ii) a speech-act based approach, relying on speech act theory (Vanderveken, 1990-1991) or on extensions of it has been used in several systems (Stent, 2002).

In this paper, we show performance improvements for a SDRT-based rhetorical structuring component of a task-oriented spoken dialogue system; these, triggered by the usage of speech acts, consist in: (i) reductions in computational costs involved by discourse structure update, and (ii) improved selection capabilities for choosing the most coherent discourse structure, out of several possibilities.

The paper is structured as follows: the next section provides a brief overview of the baseline rhetorical structuring component, the third one advocates the usage of speech acts in rhetorical structure update, through a discourse update algorithm; then, a discourse update example is presented, allowing comparisons between the baseline approach and the one integrating speech acts; finally, conclusions and pointers to further research are put forward.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 243–246,
Antwerp, September 2007.

2 Logic-Based Rhetorical Structuring Component

Our team has designed a rhetorical structuring component integrated in a natural language generation module of a task-oriented spoken dialogue system. In this context, seventeen rhetorical relations have been chosen, in the framework of SDRT, namely:

- first-order rhetorical relations - *Q-Elab*, *IQAP*, *P-Corr* and *P-Elab*, with informal semantics as in (Asher and Lascarides, 2003), that are strongly related to *temporal* aspects in dialogue, hence used in an approximate manner, specific to the type of dialogue concerned (i.e., conversations involving negotiations on time intervals of resource availability);

- second-order rhetorical relations - $Background_q$, $Elab_q$, $Narration_q$, *QAP*, *ACK* and *NEI*, with informal semantics as in (Asher and Lascarides, 2003), that are less constrained by the temporal aspects of the dialogues concerned, hence used in a manner closer to that specified in vanilla SDRT;

- third-order rhetorical relations, specific to monologues and used to relate utterances within a speech turn, generated by one of the speakers (either the human or the machine) - *Alternation*, *Background*, *Consequence*, *Elaboration*, *Narration*, *Contrast* and *Parallel*, with semantics as in vanilla SDRT (Asher and Lascarides, 2003).

Each of these 17 rhetorical relations is expressed as a predicate in first-order logic; each such predicate is expressed in terms of other predicates instantiating actions, operations and relationships between entities. These entities are objects either in a task-independent discourse ontology, or in a task ontology, as described in (Popescu *et al.*, 2007); these predicates take as arguments objects either in the discourse ontology, or in the task ontology (the entities expressing the semantics of the two utterances due to be related via a rhetorical relation. The predicates expressing the semantics of the rhetorical relations are linked through the usual connectors in first-order logic, namely $\wedge$ ("and"), $\vee$ ("or"), $\neg$ or $\Rightarrow$ (implication); furthermore, each predicate in the discourse ontology is expressed in terms of several predicates in the same ontology and of objects in either of the two ontologies.

3 Speech Acts in Rhetorical Structure Computation

Previous studies of our team advocated for the correspondences that exist between pairs of speech acts (Vanderveken, 1990-1991) (customized for human-computer dialogue) and mapping tables have been proposed, using a spoken dialogue corpus, acquired via the Wizard-of-Oz method in the context of a meeting room reservation task (Caelen and Xuereb, 2007).

The taxonomy of speech act *types* proposed by our team supposes that human-computer dialogue is a coordination of actions according to some rules (in order to reach a present or future goal). Hence, the interaction proceeds through an exchange of acts, each one having two components: (i) a propositional content, expressing the semantics conveyed by the utterance produced, and (ii) an illocutionary act that characterizes the utterance in terms of language *use*. Certain acts are *performed* in order to determine changes in the state of things - F^A: performing an action (denoted by "DO"), F^F: determining (a speaker) to perform an action (denoted by "MAKE-DO"); other acts are epistemic in nature, that is, they aim at determining changes in the discourse state or mental states of the speakers - F^S: informing a speaker about certain facts (denoted "MAKE-KNOW"), F^{FS}: asking (a speaker) about certain facts (denoted "MAKE-DO-KNOW"). Finally, there are two act types that are deontic in nature, i.e., they create obligations (necessities) or give choices (possibilities) - F^D: compel (a speaker) to do something (denoted "MAKE-MUST"), F^P: give a speaker choices of doing something (denoted "MAKE-CAN"). Each utterance is characterized by one speech act type, computed, in our architecture, by the dialogue controller for machine turns and by the pragmatic interpreter for user turns (Caelen and Xuereb, 2007); for each pair of utterances one thus has a pair of speech acts and, from a rhetorical point of view, a set of rhetorical relations connecting them.

The point we make here is that the set of rhetorical relations connecting a pair of utterances is conditioned not only by the semantics of the utterances (expressed as logic forms), but also by the speech acts characterizing them from an illocutionary point of view; an extensive corpus study regarding this

F_U^{FS}:	Where can I find book "X"?	
Possible answers of M		
F_M^{S}:	It is at the end of this corridor	*QAP*
	The plan of the book shelves is down the entrance hall	*P-Elab*
F_M^{FS}:	Is it for a scientific report you have to write?	$Elab_q$
F_M^{P}:	You can take either the hardcover edition, or the DVD edition of this book	*P-Elab*

Figure 1: Speech acts and rhetorical relations: some examples.

for each utterance α to be added to the dialogue SDRS:

1. read its corresponding logic form $K(\alpha)$, through a query to the *dialogue controller* (Caelen and Xuereb, 2007);
2. for each utterance β already in the dialogue SDRS:
   (a) read its corresponding logic form $K(\beta)$;
   (b) **read the pair $(\gamma_\alpha, \gamma_\beta)$ of speech acts for this utterance and the utterance at step 1.;**
   (c) **retrieve the set P of rhetorical relations *authorized* by the pair of speech acts read at 2.(a);**
   (d) for each rhetorical relation ρ in set P:
      i. read the semantics Σ_ρ of rhetorical relation ρ;
      ii. compute the truth value γ of the proposition $\Sigma_\rho(K(\alpha), K(\beta))$;
      iii. if $\gamma = \mathsf{FALSE}$, then go to step 2.(c); else add ρ to the set of rhetorical relations in the SDRS and α to the set of utterances in the SDRS and go to 2.(c).

Figure 2: Dialogue SDRS updating algorithm.

problem is provided in (Caelen and Xuereb, 2007).

An illustrative example in this respect is given in Figure 1, where we have two speakers, the human subject (denoted by U) and the machine (denoted by M), and that U tries to reserve a book in a library.

Using corpus-drawn examples of the type presented in Figure 1, our team has shown that for each pair of speech acts in dialogue, only some (usually, two or three) rhetorical relations (out of all the 17 considered) are *authorized* to connect the utterances involved (Caelen and Xuereb, 2007).

These results are used for refining the set of candidate rhetorical relations in (segmented) discourse structure - SDRS update, according to an improved version of the algorithm presented in (Popescu *et al.*, 2007), by taking into account speech acts in rhetorical structure update.

A rather informal statement of this improved algorithm is presented in Figure 2; steps added in the present version of the algorithm are shown in boldface. A rough estimation of the reductions in the computational cost involved by discourse structure updating can be computed supposing that the SDRS to be updated already contains N utterances, that the total number of possible rhetorical relations between utterances is $R = 17$, and that the average number of rhetorical relations authorized by a certain pair of speech acts is M (usually, 3, according to our studies). Furthermore, assuming that the time needed to read or retrieve logic formulas or speech acts is a negligible constant (since these elements are computed by the dialogue controller, independent of the language generation component (Caelen and Xuereb, 2007)), the computational cost of updating the SDRS with one utterance is $N \times R$ proofs, since each of the R rhetorical relations needs to be checked for each of the N utterances in the dialogue SDRS. We suppose that the time needed to prove a rhetorical relation between two utterances is a constant, T, thus the computational cost could be evaluated at $N \times R \times T$ without speech acts, and at $N \times M \times T$ with speech acts, hence a reduction of R/M is achieved. For the average values of R and M, the computational cost is reduced around 6 times when using speech acts.

4 Discourse Structure Update Example

In order to illustrate the augmentation of the pertinence for an updated SDRS, we consider the dialogue below (here, π_i denotes the label of the i-th speech turn):

$U : \pi_1$: Where can I find some book about "F"?
$M : \pi_2$: You want a book on "F" written by whom?
$U : \pi_3$: What's available?

From this point on, the machine is supposed to answer that books by authors "A" and "B" are available on the subject "F" and to give the user the opportunity to choose between these two authors; this drives M to produce two utterances, as an act of informing the user (a F^S), and as an act of giving

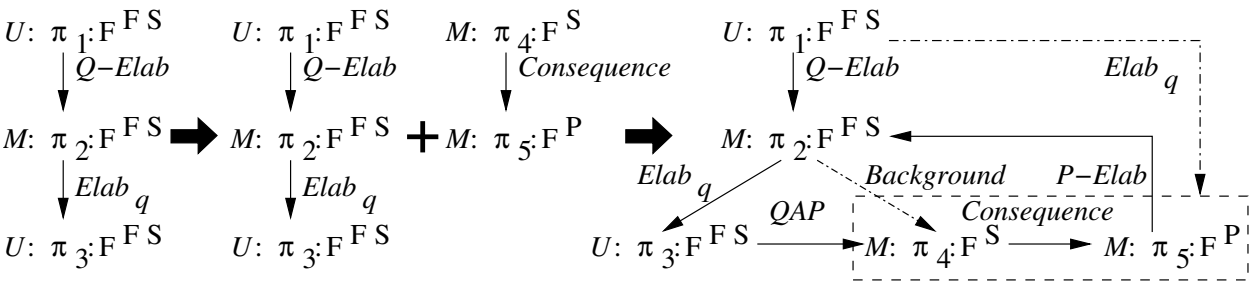


Figure 3: Discourse structure update process.

him a choice (a F^P); for these, only logic forms are available (from the dialogue controller (Caelen and Xuereb, 2007)); however, for the ease of comprehension, possible linguistic forms for them are given, in italics, below:

$M : \pi_4$: *We have books by authors "A" and "B".*

$M : \pi_5$: *Which one you like?*

Then, the machine builds a sub-SDRS using these two utterances, π_4 and π_5, and adjoins this substructure to the dialogue SDRS, formed with the utterances π_1 to π_3. This process is illustrated in Figure 3; the rhetorical relations between utterances are marked by directed labeled arrows. In da-dotted lines are marked the rhetorical relations computed as valid by the logic-based SDRS update module, but not authorized by the pair of speech acts. Thus, when the machine links π_4 and π_5 through a rhetorical relation, only *Consequence*, authorized by the pair of speech acts F^S and F^P in a monologue context, is found between these utterances. Next, the sub-SDRS thus obtained is connected to the dialogue SDRS containing utterances π_1 to π_3 via several rhetorical relations: (i) *QAP* (π_3, π_4), (ii) *Background* (π_4, π_2), (iii) *P-Elab* (π_2, π_5), (iv) $Elab_q$ (π_1, (π_4, π_5)). From these, $Background(\pi_4, \pi_2)$ and $Elab_q(\pi_1, (\pi_4, \pi_5))$ are not authorized by the pairs of speech acts, which corresponds to our intuitions and to the informal semantics of the rhetorical relations in SDRT (Asher and Lascarides, 2003).

5 Conclusions and Further Work

In this paper we have presented several improvements concerning a rhetorical structuring component for language generation in dialogue. These, based on speech act induced constraints, consisted in reduced computational costs for discourse structure update, and in greater agreement between the discourse structures obtained and human intuitions.

At present, a rhetorical structuring component prototype, integrating constraints induced by speech acts, is under development. In the near future, the discourse structuring module described in this paper will be coupled with other aspects relevant to *spoken* language generation in human-computer dialogue, namely illocutionary force control (Vanderveken, 1990-1991) and (pragmatically-motivated) anaphora generation.

References

N. Asher and A. Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

J. Caelen and A. Xuereb. 2007. *Interaction et Pragmatique - jeux de dialogue et de langage*. Editions Hermès-Lavoisier, Paris.

L. Danlos, B. Gaiffe and L. Roussarie. 2003. Document Structuring à la SDRT. *International Workshop on Generation, ACL*, Toulouse.

M. F. McTear. 2002. Spoken Language Technology: Enabling the Conversational User Interface. *ACM Computer Surveys*, 34(1).

V. Popescu, J. Caelen and C. Burileanu. 2007. Logic-Based Rhetorical Structuring Component in Natural Language Generation for Human-Computer Dialogue. *Text, Speech and Dialog 2007*, Pilsen, Czech Republic, LNCS, Springer.

A. Stent. 2002. A Conversation Acts Model for Generating Spoken Dialogue Contributions. *Computer Speech and Language*, 16.

D. Vanderveken. 1990-1991. *Meaning and Speech Acts*. Cambridge University Press.

Dialogue management for automatic troubleshooting and other problem-solving applications

Johan Boye
TeliaSonera R&D
Vitsandsgatan 9
12386 Farsta, Sweden
johan.boye@teliasonera.com

Abstract

This paper describes a dialogue management method suitable for automatic troubleshooting and other problem-solving applications. The method has a theorem-proving flavor, in that it recursively decomposes tasks into sequences of sub-tasks and atomic actions. An explicit objective when designing the method was that it should be usable by other people than the designers themselves, notably IVR application developers. Therefore the method has a transparent execution model, and is configurable using a simple scripting language.

1 Introduction

In what follows, we will consider **problem-solving dialogues** with the following characteristics:

- The dialogue participants are a novice and an expert.
- The novice has a problem he cannot solve, but is able to make observations and perform actions.
- The expert has the required domain knowledge to solve the problem, but has a limited capacity to make observations and perform actions.
- Because of this, the novice and expert need to communicate (using natural language) to jointly solve the problem.

Such dialogues appear, for instance, in the context of over-the-phone technical support and troubleshooting. Consider the situation where a service agent is helping to restore a customer's Internet connection. The agent may perform some tests remotely (pinging the customer's computer, checking for network failures, etc), but for the most part the agent tries to nail down the problem by asking the customer to perform a number of actions: restarting the modem, restarting the computer, disconnecting routers and hubs, checking and changing network settings in the computer, etc. The customer mostly acts as an answer supplier and the executor of the actions proposed by the agent.

In this paper, we will consider the challenge of automating the expert by means of a spoken dialogue system. Several issues need to be addressed. First, because the system cannot perform all actions or make all necessary observations, grounding and avoiding misunderstandings become very important. The system must make the user understand what action to perform next, and then itself understand the outcome of that action.

Second, the system must be able to adapt to different users with different levels of domain knowledge. This is particularly important in tech-support domains. While some users are perfectly comfortable with terms like "modem", "command window", "IP number", etc, many others don't know the technical terms, and indeed have very vague conceptions of computers in general. Therefore the system needs to adapt its explanations to the needs of the specific user.

Third, the system must be readily configurable, maintainable, and possible to port to new domains by application developers who do not (need to) know exactly how the system is implemented. To this end, it is important that the system offers a **scripting language** in which applications can be coded. This scripting language must have a transparent execution model, so that developers can

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 247–255,
Antwerp, September 2007.

foresee all possible situations that can arise during interaction with a user. (This last point is crucial for achieving "VUI completeness" in the sense of Pieraccini and Huerta (2005), and thus a prerequisite for a dialogue system to be useful in an industrial setting).

This paper describes a configurable dialogue manager for problem-solving dialogue applications. It is currently being used in a prototype for providing automated broadband support to the customers of TeliaSonera[1], and we will use examples from this domain throughout the article. An earlier version of the model (not as easily configurable) was used in the "Nice" fairy-tale computer game prototype (Boye and Gustafson 2005, Boye et al. 2006) as a means to control the behavior of virtual game characters (see Sect 8).

2 Problem-solving tasks and dialogues

Consider the Internet connection problem again. The service agent knows that in order for the customer's connection to work, several conditions need to be satisfied: the network must be functioning, the user must have paid his bill, and the user's equipment must be functioning and set up appropriately. Put equivalently, if the Internet connection does **not** work, one of the conditions just mentioned must be unfulfilled. This latter formulation suggests a procedure for finding the source of the connection error; just check the prerequisite conditions one at a time, until the error is uncovered. If we find that one condition is fulfilled (e.g. the network is working properly), we can cross it off our check list and proceed to examining the other conditions, as the problem surely must lie with one of them.

Some of the conditions are complex, and can be decomposed further into sub-conditions. For instance, that "the user's equipment is functioning" involves checking that the modem, router, computer, wires, etc. are all ok, and some of these sub-conditions can be further broken down into sub-sub-conditions, and so on. That is, the troubleshooting process takes the form of a **hierarchical task decomposition** process.

The way propositions are expressed in terms of sub-propositions as above is reminiscent of Prolog (Sterling and Shapiro, 1994). The problem could also be viewed as one of finding a sequence of actions that transform the current state of affairs (in which the Internet connection is not working) into a desired goal state (in which the connection works). On this viewpoint, the problem seems amenable to AI planning approaches like STRIPS (Fikes and Nilsson, 1971).

However, both Prolog and STRIPS-like approaches assume that full information is available from the very beginning, and that problem solving amounts to searching through known facts about the domain. This is not the case in the problem-solving domains we are considering. A further complication is the fact that the agent cannot carry out actions directly, and cannot foresee which instructions will be understood or not. This is evident from the following fragment (taken from a longer dialogue):

1. **Agent:** We will need to restart your modem. Do you know where your modem is?
2. **User:** Uh, no.
3. **A:** I'll help you find it. Can you locate the telephone plug in the wall?
4. **U:** Uh, yes.
5. **A:** One of the cables going from the telephone plug should lead to a little box that probably has some lights on it.
6. **U:** Ok.
7. **A:** That is your modem.
8. **U:** Ok, I see it.
9. **A:** Good! Now, the modem has a power cord as well. I want you to unplug that power cord in order to restart the modem. Can you do that now?

The user cannot directly restart his modem since he cannot locate it, so the agent needs to embark on a longer explanation (utterances 3–8).

A striking feature of such troubleshooting dialogues is the amount of effort the agent spends on grounding. In fact, the agent needs to have almost every instruction acknowledged by the user (usually a brief "yes" or "OK" is sufficient). If the user does not know how to carry out an instruction (as in utterance 2 above), or rejects it for some other reason, the agent will either explain further, or abandon the current strategy altogether and try an alternative way to proceed.

[1] TeliaSonera is the leading telecommunications provider in the Nordic-Baltic region in Europe.

Smith and Hipp (1994) proposed the "missing axiom theory" as the driving force in problem-solving dialogue management. In this view, completion of actions is represented by theorems, and making sure that an action has been completed involves constructing a proof for the corresponding theorem. If the proof can not be carried out because some needed axiom is missing, the theorem proving process is suspended, and the user is asked to provide the missing axiom (this amounts to a request to the user to perform an action needed to complete the overall task).

Since Smith's system, several other researchers have applied hierarchical task decomposition to dialogue, notably Rich and Sidner (1996), Lemon et al (2002), and Bohus and Rudnicky (2003). The approach presented in this paper differs from aforementioned approaches primarily by featuring a much simpler way of scripting dialogue applications. Automated troubleshooting dialogue has recently been addressed by Acomb et al (2007), and by Williams (2007), who uses a statistical dialogue management approach rather than hierarchical task decomposition.

3 Encoding the domain

3.1 Speech acts

An analysis of a corpus of dialogues between human service agents and customers revealed that the vast majority of the agent's utterances can be described using only six speech acts. These are "**request action**" (e.g. "Locate the telephone plug in the wall"), "**request info**" ("What operating system is your computer running?"), "**request info yes/no**" ("Is your router wireless?"), "**ground status**" ("Now a window should appear"), "**inform**" ("There may be a problem with your router"), and "**acknowledge**" ("Good!").

After having performed an "**inform**" speech act, the agent is not really expecting any reply from the customer; making an "**inform**" is just granting extra information concerning the state of the troubleshooting process (often used when a topic is introduced ("We will need to disconnect your router") or when it's closed ("Now we've disconnected your router.")). In contrast, the "**request info**" speech act requires a reply from the customer, and the agent cannot proceed without it. A **"ground status"** is used when the agent wants to confirm a certain result, for instance that the user can see the "Start" menu appearing on his screen after having clicked the "Start" button. The main purpose of a "ground status" speech act, from the agent's point of view, is to make sure that the user has indeed carried out and understood the effects of the latest action, and is ready to receive the next instruction.

Similarly, the customers' utterances can be classified using speech acts such as "**inform"** (typically as a reply to a "request info"), "**inform-yes"** and "**inform-no"** (in response to a "request info yes/no"), "**acknowledge"** (typically signaling understanding in response to a "request action" or "ground status"), "**please clarify"** (signaling non-understanding), and "**please wait"** (when the user needs more time to carry out some action). Additionally, the customer usually states the problem at the very beginning of the dialogue. We will not consider this heterogeneous group of utterances in this article, as they are dealt with using statistical classification methods (see Boye and Wirén 2007 for a description of that system), quite unlike those presented here.

We will also consider instantiations of the basic speech acts. For instance, "locate the telephone plug in the wall" is an instantiation of a "**request action"**, which we will represent as **requestAction(locate_telephone_plug).** In general, we will represent (the semantic value) of an utterance[2] by such an instantiated speech act $\mathbf{f(a_1, a_2, \ldots, a_n)}$, where **f** is the basic speech act, and the arguments $\mathbf{a_1 \ldots a_n}$ are terms representing the additional information. As another example, "the IP number is 131.1.15.23" would be represented as **inform(ipNumber, 131.1.15.23)** whereas "no" would be represented simply as **inform-no.**

We can now encode the entire dialogue example of section 2, as follows:

1. **Agent:** inform(restart_modem); requestAction(locate-modem)
2. **User:** pleaseClarify
3. **A:** requestAction(locate-telephone-plug)
4. **U:** acknowledge
5. **A:** requestAction(follow-cable-from-telephone-plug)
6. **U:** acknowledge
7. **A:** groundStatus(locate-modem)

[2] In general, an utterance may be represented by a sequence of speech acts (and not necessarily a single speech act).

8. **U:** acknowledge
9. **A:** acknowledge
   requestAction(unplug-power-cord-from-modem)

3.2 Information state

Relevant information about the domain is stored as attribute-value pairs. For instance, we may conceive of an attribute **ipNumber** whose value is **131.1.15.23**. A "proposition" is any statement of the domain that can be either true or false. In particular, the expression **valueOf(x,y)** denotes the proposition that the attribute x has the value y. Some attributes can only take the values **true**, **false,** or **don't know**. If x is such an attribute, we will take the expression **x** to mean the same thing as **valueOf(x,true)**. For instance, **modem-restarted** means the same thing as **valueOf(modem-restarted, true).** We will refer to the ensemble of attribute-value pairs as the "information state"**.**

A proposition is considered to be true, and stored in the information state, as soon it is accepted by the user. For instance, the proposition **locate-telephone-plug** is added after the user's acknowledgement in utterance 4, and **follow-cable-from-telephone-plug** is added after utterance 6.The proposition **locate-modem** is *not* added after utterance 2 since the user does not acknowledge, but is added after utterance 8. Thus, the presence of a proposition like **locate-modem** in the information state in this case means that the user has confirmed that he has performed the action "locate modem". (One may argue that the user having located his modem is an observation rather than an action. However, the distinction between verified executed actions and verified observations is intentionally blurred.)

Non-Boolean values of attributes are added after an **inform** reply from the user (as, for instance, in the exchange: "What operating system is your computer running?", "Windows"). The presence of the proposition **valueOf(operating-system, windows)** in the information state means that the system has already performed a speech act **request-info(operating-system)**, or obtained the information by some other means. In any case, the question needs not be asked again.

4 Deciding system actions

4.1 Dialogue rules: syntax and informal interpretation

In what follows, we will use a rule-based approach of representing the problem decomposition process outlined previously. A rule for making the user restart his modem might look like this:

```
satisfy(restart-modem) {
  satisfy locate-modem;
  perform requestAction(unplug-power-
cord-from-modem);
  perform requestAction(plug-power-
cord-into-modem);
  perform groundStatus(restart-
modem);
}
```

Informally, such a rule is to be interpreted: "In order to have the modem restarted, first make sure that the modem is located (by the user), then ask the user to unplug the power cord, and then ask the user to plug the power cord back in again. Finally, ask the user to verify that the modem actually has been restarted". (We will return to the formal interpretation of the rule shortly.)

That is, the process of satisfying a certain goal can be broken down into a sequence of steps, each of which is either a sub-goal to be satisfied, an action to be executed, or a condition that should be true. The general form of a rule is

satisfy(G) { B_1; B_2; …; B_n; }

where G is a proposition to be satisfied ("the goal"), and each B_i is an expression of one of the following forms:

- **satisfy** P (where P is a proposition)
- **perform** A (where A is an action, i.e. either a speech act or a request for a non-verbal action, such as pinging the user's computer)
- **holds** P (where P is a proposition)

(We will explain the **holds** construct in end of this section.)

Continuing the example, there are two rules for the sub-goal `locate-modem`, corresponding to two alternative strategies for how the agent can proceed. The simple way of making sure the user has located his modem is simply to ask him:

```
satisfy(locate-modem) {
  perform requestAction(locate-
modem);
}
```

The speech act `requestAction(locate-modem)` could for instance be verbalized as "Do you know where your modem is?", as in the second sentence of utterance 1 in the example of section 2. If the user okays this request, the system will draw the conclusion that the goal `locate-modem` is fulfilled (i.e. add that proposition to its information state). Another strategy to fulfill the goal `locate-modem` is to give a step-by-step explanation:

```
satisfy(locate-modem) {
    perform requestAction(locate-
telephone-plug);
    perform requestAction(follow-
cable-from-telephone-plug);
    perform groundStatus(locate-
modem)
}
```

This is what the agent does in utterances 3-8 in the example of section 2.

The informal interpretation of the construct "**holds** P" is that the proposition P must be true at that point in order for the rule to be applicable. Usually, it is used as a pre-condition, as in the rule:

```
satisfy(check-network-settings) {
  holds valueOf(operating-system,
windows);
  . . . more . . .
}
```

Unless the system already knows that the user's operating system is Windows, this rule is not applicable.

We will also allow variables in rules, as in the following rule (variables are prefixed with a "$"):

```
satisfy(valueOf(radio-button($x),$y){
  perform requestAction(tick(radio-
button($x, $y));
}
```

This rule states that one way of ensuring that the alternative $y is ticked in the radio button $x is to ask the user to tick it (whatever the values of $x and $y). The use of variables is a notational convenience that reduces the number of rules by increasing their applicability.

Rules such as these constitute a *static* specification of how the automated agent can go about diagnosing and correcting the error (by "static" we mean that the rules will not change during the course of a dialogue).

4.2 The agenda and the formal interpretation of dialogue rules

During the course of the dialogue, the system makes use of the rules to construct and traverse a **dynamic** tree-structure, the **agenda**, which at any point in time represent current and future goals and actions. The agenda is a tree-structure since goals are represented as parent nodes of the sub-goals and actions needed to fulfill them.

Agenda trees can be defined inductively as follows:

- if P is a proposition, then a single node labeled with "**satisfy** P" is an agenda;
- if A_1 is an agenda, then A_2 is an agenda if A_2 can be constructed from A_1 by means of the following **expansion** operation:
  (1) choose a leaf node L which is labeled "**satisfy** X"
  (2) choose a matching dialogue rule "**satisfy** Y { B_1; … B_n }", where σ is a binding of the variables in Y, such that σ(Y)=X. Add n children to L, labeled σ(B_1), …, σ(B_n).

As an example, the agendas in figures 1c and 1d (found at the end of the article) are both obtained by expansion (using two different rules) of the agenda in figure 1b, which in its turn is an expansion of the agenda 1a.

Note that it is also possible to transform agenda 1c into 1d by selecting the node labeled "**satisfy** locate-modem", pruning all children below that node (we will refer to this operation as performing a "cut-off" at that node), and then expanding that same node using another rule.

Whenever the system needs to decide what to do next, it searches, expands and transforms the agenda in order to find the *next action node*. The next action node is always labeled "perform A", where A is taken to be the action to be carried out next.

In order to find the next action node, the agenda is searched depth-first, left-to-right, starting from the top node, ignoring already satisfied goals and

executed actions, until the first non-executed action is encountered. More precisely, for each visited node **n**, the following decisions are made:

1. If **n** is labeled "**perform** A":
    a. If A has already been performed (this is determined as described in section 3.2), then proceed to the next sibling node.
    b. If A has not been performed, then **n** is the next action node, and A is the action to carry out next.
2. If **n** is labeled "**satisfy** P":
    a. If P is a true proposition then proceed to the next sibling node.
    b. If P is not true, then proceed to the leftmost child of **n**. If **n** is a leaf node, then expand (using the expansion operation above), and then proceed to the leftmost child of **n**.
3. If **n** is labeled "**holds** P":
    a. If P is a true proposition, then proceed to the next sibling node.
    b. If P is not true, remove **n** and all of **n**'s siblings. Then expand **n**'s parent node, using another rule than before, and proceed to the leftmost child of **n**.

In cases 2b and 3b, the system currently uses the Prolog-like strategy of using the rules in the order they are listed. That is, in case 2b the first matching rule is selected, and in case 3b the first *unused* matching rule is selected.

To illustrate how the system uses the agenda, suppose figure 1a is the starting point. The system would expand the agenda twice, leading to figure 1c. The next action node is thus labeled "**perform** requestAction(locate-modem)", which is what the system will say (verbalized as utterance 1 of the dialogue example of section 2).

Since the user does not acknowledge but rather asks the system to clarify (in utterance 2), the system considers the chosen strategy to be no good. As a reaction, the agenda is rebuilt into figure 1d.

5 Interpreting user input

Each speech act has an associated system utterance, and most of them have an associated grammar. Furthermore, all speech acts have an associated set of **expectations** that tells the system how to interpret the user's input. When a particular speech act is chosen by the system as the next action, the associated utterance is played, and then speech recognition is performed using the associated grammar. If there is no associated grammar, the system assumes that it is its turn to speak again.

After **request action** and **ground status** speech acts, a grammar is used which is capable of recognizing the user speech acts **acknowledge, please clarify,** and **please wait** (speech recognition grammars with semantic attachment rules are used, so there is no need for a separate parsing step). As explained in section 3.2, an **acknowledge**ment from the user makes the system consider the proposition under discussion to be true (and add it to the information state). This is what happens in the utterances 3-8 in the dialogue example. Using the algorithm described in section 4.2, the system traverses the agenda (in figure 1d), and visits the nodes marked A, B, and C, in that order.

On the other hand, if the user asks the system to clarify, the system will abandon its current strategy, and rebuild the agenda. That is what happens after utterance 2, when agenda 1c is rebuilt into agenda 1d. This is done by removing the current action node and all its siblings, and re-expanding the parent node (in this case labeled "**satisfy** locate-modem") using the next applicable rule.

Some speech acts have specially developed associated grammars. For instance, the speech act **requestInfo(ipNumber)** has a grammar recognizing IP numbers, and so on. The recognized utterance will be interpreted as a value for the attribute (**ipNumber**, in this case), unless the user makes a **please clarify** or **please wait** speech act (these are always among the user's options).

6 Associating utterances with tree events

In the algorithm of section 4.2, the agenda is traversed, expanded and transformed in order to find the next action. During this process, a number of **events** are generated, notably

- When a **satisfy** node is expanded (a "**topic intro**" event).
- When a cut-off is performed at a **satisfy** node, and the node is expanded using the next applicable rule (a "**new strategy**" event).

- When a proposition P is first found to be true, after it has previously been found to be false (a "**topic outro**" event).
- When the system attempts to rebuild the tree, but there are no more unused matching rules (a "**cannot solve**" event).

Note that the first two events correspond (roughly) to the "call" and "redo" entry points in the Prolog "procedure box" control flow model (Byrd 1980), whereas the two latter events correspond, respectively, to the "exit" and "fail" points in the same model.

A useful feature in the dialogue manager is that it allows the dialogue designer to associate system utterances to such events. If there is no associated utterance, an event will just pass unnoticed, otherwise the associated system utterance will be generated.

For example, the event **topicIntro(restart-modem)** is generated when the agenda in figure 1a is expanded into figure 1b, and the event **topicIntro(locate-modem)** is generated in the transition from 1b to 1c. Suppose we associate the utterance "We will need to restart your modem" with the former event (and no utterance with the latter event); then this utterance is generated just before the **requestAction(locate-modem)** utterance ("Do you know where your modem is?"). Together, these two make up the system's first utterance in the dialogue example of section 2.

In the same vein, we may associate the utterance "Good!" with the event **topicOutro(locate-modem)**. When the user has finally located his modem (in utterance 8), the proposition **locate-modem** is added to the information state. At that point in time, the agenda looks like figure 1d. When the system traverses it and reaches the "**satisfy**(locate-modem)" node, the **topicOutro(locate-modem)** event is generated just before the system moves to the next node and generates the **requestAction(unplug-power-cord-from-modem)** utterance. Together, these two make up utterance 9 in the dialogue example.

7 Putting it all together

This is a summary of the execution model of the dialogue manager:

1. The agenda is traversed (and possibly expanded or transformed) using the algorithm of section 4.2. All utterances associated with the ensuing tree events are generated.
2. The result of step 1 is an action or a speech act (if there is no result, the dialogue is finished). Perform this action (in the case of a speech act, generate the associated utterance).
3. If the speech act has an associated grammar, perform speech recognition. Then interpret the resulting speech act based on the expectations associated with the system's latest speech act.
4. Go to 1.

8 Other kinds of problem-solving applications

We began the paper by considering dialogues featuring an expert and a novice, trying jointly to solve a problem. The endeavor here has been aiming at automating the expert side of such a dialogue.

Other configurations are also possible. In spoken natural language robot control interfaces, such as considered e.g. in Rayner et al. (2000), the human takes the role of the expert, having the responsibility for long-term planning, whereas the robot is the novice, responsible for executing actions and making observations. If the robot or device has some planning capabilities of its own, the expert-novice distinction is not clear-cut, and plans may be constructed jointly (see Rich and Sidner 1996, Lemon et al 2002).

An interesting situation is when both the expert and the novice are automated. This might be the case in interactive entertainment (Cavazza et al 2002), or in computer games such "Nice" (Boye and Gustafson 2005, Boye et al 2006). The Nice game features two animated characters with whom the user can talk; however they can also communicate with each other and interfere in each other's plans.

The Nice game used the same dialogue management kernel as the one described in this paper. However, free input was allowed (using a stochastic language model for speech recognition, and a separate robust parsing step), and the system was also capable of performing some reference resolution. Another difference is that the tech-support

application described here has a fixed overall goal with the dialogue (the top node of the agenda), which is kept throughout. By contrast, the game characters in the Nice game added new goals to the agenda during the dialogue, as a result of the user's requests and questions.

9 Concluding remarks

In the introduction, we stated three important issues: (1) grounding and avoidance of misunderstandings, (2) on-the-fly adaptation to different kinds of users, and (3) ease-of-use for application developers.

Misunderstandings are avoided, or at least made less probable, by not updating the information state without a confirmation from the user. Rules that encode action chains in several steps are best concluded with a **ground status** speech act, which the user has to confirm ("Now you've restarted your modem.", "Ok!").

The system adapts to the user by rejecting the current strategy and replacing it with an alternative strategy (an alternative dialogue rule) as soon as the user indicates that he does not understand. This may amount to no more than replacing a direct request ("Can you restart your modem?") with a more elaborate step-by-step description to achieve the same thing. But it may also mean trying an alternative way to proceed. For instance, if the user is unable to detect the "Start" button on the screen of his Windows computer, the system may instead ask him to press the "Windows" button on his keyboard.

Finally, as concerns ease-of-use for application developers, our initial experiences are positive, though the broadband tech-support prototype is still under development. It is planned to be deployed by the end of 2007.

Acknowledgement: The author would like to thank Mats Wirén and the anonymous reviewers for valuable comments. This work was supported by EU's 6th framework project "COMPANIONS".

References

Acomb, K., Bloom, J., Dayanidhi, K., Hunter, P., Krogh, P., Levin, E. and Pieraccini, R. (2007) Technical support dialog systems: Issues, problems and solutions. *Proc. Naacl'07 Workshop on Bridging the gap: Academic and industrial research in dialog technologies,* Rochester, NY.

Bohus, D., and Rudnicky A. (2003) RavenClaw: Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda, *Proc. Eurospeech*, Geneva, Switzerland.

Boye, J., and Gustafson, J. (2005) How to do dialogue in a fairy-tale world. *Proc. SIGDIAL.*

Boye, J., Gustafson, J. and Wirén, M. (2006) Robust spoken language understanding in a computer game. *Speech Communication*, 48, pp. 335-353.

Boye J. and Wirén, M. (2007) Multi-slot semantics for natural-language call routing systems. *Proc. Naacl'07 Workshop on Bridging the gap: Academic and industrial research in dialog technologies,* Rochester, NY.

Byrd, L. (1980) Understanding the control flow of Prolog programs, *Proc. Logic Programming Workshop*, Debrecen, Hungary

Cavazza, M., Charles, F. and Mead S. J. (2002) Character-based interactive storytelling. *IEEE Intelligent Systems*, Special issue on AI in Interactive Entertainment, pp. 17-24.

Fikes, R. E., and Nilsson, N. (1971) STRIPS: a new approach to the application of theorem proving to problem solving, *Artificial Intelligence*, 2 (3-4), pp.189-208

Lemon, O., Gruenstein, A. and Peters, S. (2002) Collaborative activities and multi-tasking in dialogue systems. *Traitement Automatique des Langues (TAL)*, special issue on dialogue, 43(2), pp. 131-154

Pieraccini, R., and Huerta, J. (2005) Where do we go from here? Research and commercial spoken dialog systems, *Proc. SIGDIAL*

Rayner M., Hockey B.A. and James, F. (2000) A compact architecture for dialogue management based on scripts and meta-outputs, *Proc. Applied Natural Language Processing (ANLP).*

Rich, C., and Sidner, C. (1996) When agents collaborate with people, *Proc. AGENTS'97, 1st international conference on autonomous agents.*

Smith, R. and Hipp, R. (1994) *Spoken natural language dialog systems: A practical approach*, Oxford University Press.

Sterling, L., and Shapiro, E. (1994) *The art of Prolog,* 2nd edition, The MIT Press.

Williams, J. (2007) Applying POMDPs to dialog systems in the troubleshooting domain. *Proc. Naacl'07 Workshop on Bridging the gap: Academic and industrial research in dialog technologies,* Rochester, NY.

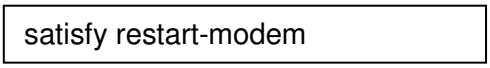


Figure 1(a): An agenda consisting of one node

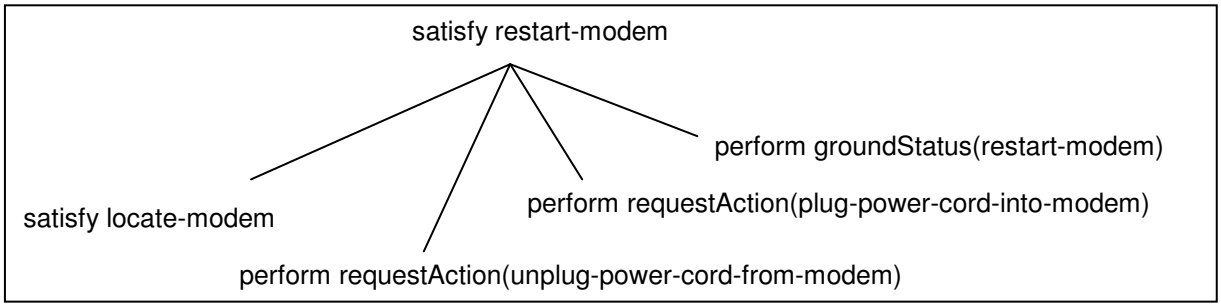


Figure 1(b): An agenda which is an expansion of 1(a)

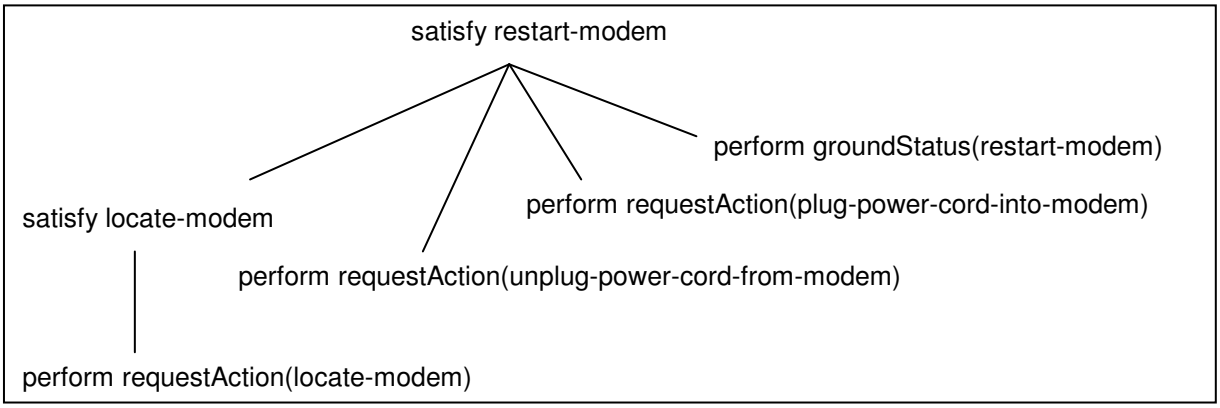


Figure 1(c): An agenda which is an expansion of 1(b)

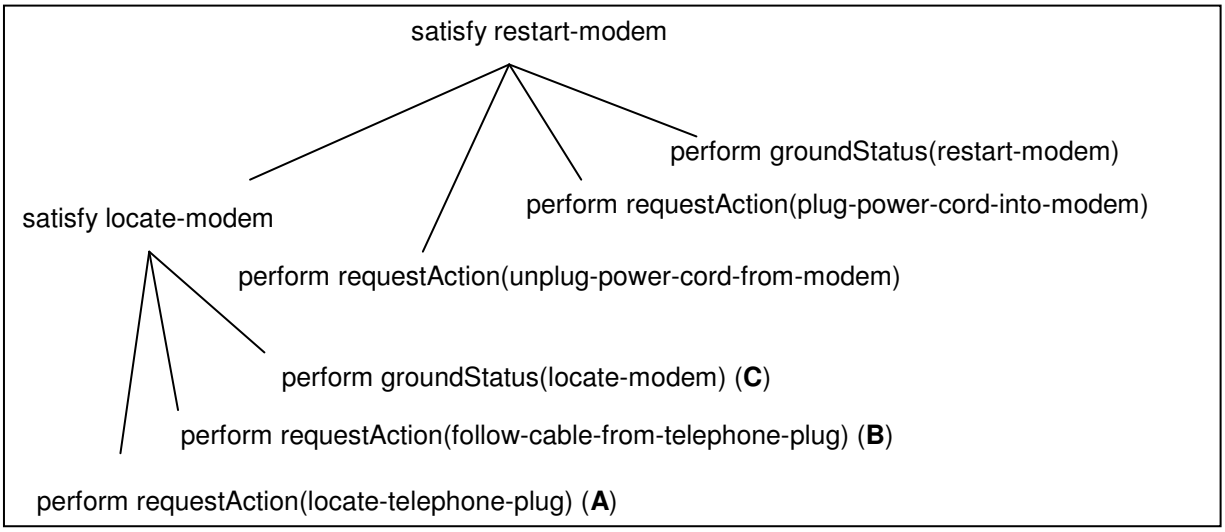


Figure 1(d): Another agenda which is an expansion of 1(b)

Implicitly-supervised learning in spoken language interfaces: an application to the confidence annotation problem

Dan Bohus[*]
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, 15217
dbohus@cs.cmu.edu

Alexander I. Rudnicky
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, 15217
air@cs.cmu.edu

Abstract

In this paper we propose the use of a novel learning paradigm in spoken language interfaces – implicitly-supervised learning. The central idea is to extract a supervision signal online, directly from the user, from certain patterns that occur naturally in the conversation. The approach eliminates the need for developer supervision and facilitates online learning and adaptation. As a first step towards better understanding its properties, advantages and limitations, we have applied the proposed approach to the problem of confidence annotation. Experimental results indicate that we can attain performance similar to that of a fully supervised model, without any manual labeling. In effect, the system learns from its own experiences with the users.

1 Introduction

Spoken language interfaces are complex systems that combine many diverse sources of knowledge. Oftentimes, simple algorithmic approaches are insufficient for solving the difficult problems that arise. Instead, machine learning techniques are used, and one of the most often encountered paradigms is that of supervised learning. In this paradigm, the developer provides a training dataset that contains pairs of inputs and desired outputs, and various learning algorithms can be used to derive a model that captures and generalizes the relationship between the two. At runtime, the system generates the corresponding output based on the current input and on the learned model. Such approaches are used in a variety of tasks in spoken dialog systems: acoustic and language-modeling, confidence annotation, dialog act tagging, emotion detection, user modeling, etc.

Supervised learning approaches have however at least two important limitations. First, they require a pre-existing corpus of labeled data. Unfortunately, such corpora are difficult and expensive to collect, especially in the early stages of system development. Secondly, they generally favor an off-line, or "batch" approach. A corpus is collected, manually labeled, and then model parameters are estimated from this data. The resulting model mirrors the properties of the training set, but does not respond well to changes in the system's environment and the underlying data distribution. Unfortunately, such changes are generally expected. Oftentimes, system developers might alter various aspects of system functionality based on feedback and observations. In addition, the users' behavior changes as they repeatedly interact with the system and familiarize themselves with it. Finally, the very introduction of the newly trained model can lead to changes in the interaction. Conversational spoken language interfaces are interactive systems that operate in dynamic environments, and shifts in the underlying data distribution are inevitable.

In this paper, we propose and evaluate a novel learning paradigm that addresses these drawbacks. The proposed approach, dubbed **implicitly-supervised learning**, builds on a key property of spoken dialog systems: their interactivity. The central idea is to extract the required supervision signal from naturally-occurring patterns in the conversation, for instance from user corrections. No developer supervision is therefore required. Rather, the system learns on-line, throughout its lifetime, by interacting with its users. We believe this new para-

[*] Currently at Microsoft Research, Redmond, WA

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 256–264,
Antwerp, September 2007.

digm can be applied in a number of learning problems, and can pave the way towards building routinely self-improving systems.

Consider for instance the problem of confidence annotation. Spoken dialog systems use confidence scores to guard against potential misunderstandings: for every utterance, a confidence score reflecting the probability that the system correctly understood the user's utterance is computed. Confidence annotation models are traditionally built using supervised learning techniques (Litman et al, 1999; Carpenter et al, 2001; San-Segundo et al, 2001; Hazen et al, 2002; Hirchberg et al, 2004.) A corpus of dialogs (typically thousands of utterances) is manually labeled by a human annotator: each utterance is marked as either correctly-understood or misunderstood by the system. Supervised learning techniques are then used in conjunction with features that characterize the current utterance to train a model that can predict whether or not this utterance was misunderstood by the system. This approach suffers from the shortcomings we have outlined above: it requires a pre-existing corpus of in-domain utterances, a significant amount of human effort and expertise for labeling this corpus, and it produces a static solution.

The alternative implicitly-supervised solution eliminates these drawbacks. The starting point is the observation that the system could obtain the necessary information (i.e. the misunderstanding labels) by leveraging a particular confirmation pattern that occurs naturally in conversation. Consider the example in Figure 1, from Let's Go! Public (Raux et al, 2006), a spoken dialog system that provides bus schedule information in Pittsburgh. In the first turn, the system asked for the departure location. The user responded "the airport", but this was misrecognized as "Liberty and Wood". Next, in turn 2, the system tried to explicitly confirm the departure location it heard. The user corrected the system by answering "no". The immediate reason

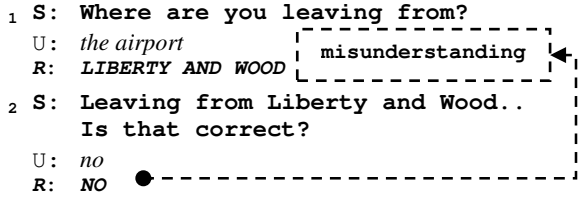


Figure 1. User responses to explicit confirmation questions can provide labels for building a confidence annotation model

for the user response in turn 2 was to allow the conversation to proceed correctly. Notice however that this interaction pattern generates additional useful information: the system now knows that it misunderstood the user in turn 1 and can use this information to refine the confidence annotator.

Spoken dialog systems should be able to successfully elicit and leverage this and other interaction patterns to continuously improve their performance, without developer supervision. For instance, we can envision a system that starts by explicitly confirming all the pieces of information it acquires from the user – many systems do this routinely. As the system collects more labels through interaction and updates its confidence annotation model, its error detection abilities improve and the system can start trusting the confidence annotation model more, and use explicit confirmations only when the confidence score is very low. Several interesting questions arise: (1) can a system make effective use of the information obtained through interaction? (2) How can a system balance its long-term knowledge elicitation goals with the short-term need to efficiently provide information to the user? (3) Could a system discover new interaction patterns that can provide labels for confidence?

We believe that implicitly-supervised learning approaches can be used in a number of other problems in spoken language interfaces (more on this in Section 7.) The work described in this paper constitutes only a starting point for a larger research program aimed at investigating the properties, advantages and limitations of this paradigm. We begin our investigation by applying the proposed approach to the confidence annotation problem. Moreover, we focus for now only on the first one of the three questions we have raised above: can a system make effective use of the information obtained through interaction to build a high quality confidence model? In future work, we plan to address the remaining questions, and to investigate the use of this paradigm in other problems.

2 Implicitly supervised learning for confidence annotation

We have already outlined the basics of using implicitly-supervised learning for building confidence annotation models. The key idea is that the system can obtain the required supervision signal by leveraging a certain pattern that occurs naturally in

conversation: in this case user responses to explicit confirmation questions. This eliminates the need for developer supervision (i.e. for manually labeling data) and in the process creates an opportunity for continuous, on-line learning. The implicitly obtained labels (**implicit labels** in the sequel) can be used in conjunction with a traditional supervised learning methodology to construct or refine a confidence annotation model.

More specifically, the implicit labels are generated automatically as follows: if the system engages in an explicit confirmation and the recognized user response was yes (or equivalent), then the previous user turn is labeled as correctly understood by the system; alternatively, if the recognized user response was no (or equivalent) the previous user turn is considered misunderstood by the system; finally if the recognized user response did not contain a positive or negative marker, no implicit label is generated. Note that the implicit labels are not noise-free. In the example from Figure 1, the user response was a simple "no", which was correctly understood by the system. In general, user responses to explicit confirmation actions extend beyond simple yes and no answers, and can also be subject to recognition errors (Krahmer et al, 2001; Bohus and Rudnicky, 2005.) As a consequence, the labels produced by this interaction pattern will not always be perfect.

The implicit labels can be characterized in terms of **accuracy** and **recall**. In this context, by **accuracy** we will refer to the accuracy of the implicit labels with respect to the reference set of manual labels. By **recall** we refer to the proportion of utterances for which this interaction pattern can generate labels (i.e. the utterances followed by an explicit confirmation and a simple user response.) Finally, there is a third factor that affects the quality of the implicitly labeled data: **the sampling bias**. Even though the proposed interaction pattern provides labels for a certain proportion of the utterances in the corpus, these implicitly labeled utterances do not constitute a random sample of the entire corpus. Rather, these are utterances that are followed by explicit confirmations, which in turn are followed by simple user responses. The underlying distribution of the features in this subset of utterances does not necessarily match the general distribution in the full set of utterances. Similarly, because this implicit labeling scheme relies on recognition of user responses, it might bias the implicit labels towards one of the two classes.

Whether or not these implicit labels are sufficient for training an accurate confidence annotation model remains an open question. In this paper, we empirically investigate this question, using corpora collected with two different spoken dialog systems.

3 Systems

The first system, Room-Line, is a telephone-based, mixed-initiative spoken dialog system that can assist users in making conference room reservations on the CMU campus (Bohus, 2007). The system has access to the live schedules of 13 conference rooms on campus, and to their characteristics, and can engage in a negotiation dialog to identify the room that best matches the user's needs.

The second system, Let's Go! Public (Raux et al, 2006), provides bus route and schedule information in the greater Pittsburgh area. Since March 2005, this system has been connected to the Pittsburgh Port Authority customer service line during non-business hours, and therefore receives a large number of calls from users with real needs.

4 Data

The RoomLine corpus consists of 484 dialogs (8037 user turns) collected in a user study in which 46 participants were asked to perform 10 scenario-based interactions with the system. The Let's Go! Public corpus consists of a subset of 617 dialog sessions (6029 utterances) collected during the first month of public operation for the system. Both corpora were orthographically transcribed, and misunderstandings were manually labeled. Table 1 shows a number of basic corpus statistics.

The RoomLine and Let's Go! Public systems used very different policies for engaging in explicit confirmations. RoomLine made this decision by comparing the confidence score of the recognized utterance against a confirmation threshold. As a result, the total number of explicit confirmations in this corpus is 1412, amounting to 17.6% of the total number of utterances (8037). In contrast, given the more adverse environment, the Let's Go! Public system used a simpler, more conservative confirmation policy: the system always explicitly confirmed every piece of information received from the user. The number of explicit confirmations in the Let's Go! Public corpus is therefore signifi-

Statistics	RoomLine	Let's Go
# of sessions	484	617
# of utterances	8037	6029
# of misunderstandings	1523	1863
% misunderstandings	18.9%	30.9%
# of explicit confirmations	1412	2594
% of explicit confirmations	17.6%	43.0%
# Implicit labels	976	1998
Implicit labels recall	10.8%	33.1%
Implicit labels accuracy	89.9%	82.5%

Table 1. Corpora statistics

cantly larger – 2594, representing 43.0% of the total number of utterances (6029).

Due to the different confirmation policies, the recall and the accuracy of the implicit labeling scheme proposed above was different in these two domains. As expected, given that explicit confirmations were more often engaged in the Let's Go! Public system, the recall of the implicit labeling scheme was significantly larger than in the RoomLine system: 33.1% versus 10.8%. At the same time, given the more adverse noise conditions and worse recognition performance in this domain, the accuracy is lower: 82.5% versus 89.9% in the RoomLine system.

5 Features

To build the confidence annotation model, we considered a large set of features extracted from different knowledge sources in the systems. Below, we give a brief overview of these features. The full feature set is presented in detail in (Bohus, 2007):

- **speech recognition features,** e.g. acoustic and language model scores; # of words and frames; word-level confidence scores generated by the recognizer; signal and noise-levels; speech-rate; etc.
- **prosody features,** e.g. various pitch characteristics such as mean, max, min, standard deviation, min and max slopes, etc.
- **lexical features,** e.g. presence or absence of the top-10 words most correlated with misunderstandings (these are system-specific.)
- **language understanding features,** e.g. number of (new / repeated) semantic slots in the parse; measures of parse-fragmentation;
- **inter-hypotheses features.** features describing differences between the top-most hypothesis from each recognizer (each system used 2 gender-specific parallel recognizers);
- **dialog management,** e.g. match-score between the recognition result and the dialog manager expectation; dialog state; etc.
- **dialog history,** e.g. # of previous consecutive non-understandings; ratio of non-understandings up to the current point in the dialog; tallied averages of the acoustic-, language-model, and parse-scores.

6 Experimental results

We used stepwise logistic regression (Myers et al. 2001) to train confidence annotation models based on the implicitly labeled portions of the RoomLine and Let's Go! Public corpora. The features described in the previous section served as independent variables in the model; the dependent (target) variable was whether or not the utterance was correctly understood by the system. The models were trained and evaluated using a 20-fold cross-validation procedure. The quality of the models was assessed in terms of mean squared error, also known as Brier score. In contrast to classification error metrics, the Brier score is a proper-scoring rule that captures both the refinement (accuracy) as well as the calibration of the confidence annotator (Cohen and Goldszmidt, 2004.)

We begin by describing results in the Let's Go! Public system, because the number of implicitly labeled training points in this corpus is larger and enables a more robust analysis.

6.1 Results in the Let's Go! Public domain

The results are illustrated in Figure 2. The Brier score for the majority baseline (i.e. always predicting the majority class) is 0.2156. The average test-set Brier score for the fully-supervised model, i.e. the model that uses the entire Let's Go! Public corpus with the manually annotated labels, is 0.1200. The proposed implicitly-supervised approach leads to an average test-set Brier score of 0.1443, closing 75% of the gap between the majority baseline and the fully-supervised model, without requiring any manually labeled data.

If a small amount of manually labeled data is available, it can be used to calibrate the implicitly-supervised model. The post-calibration step consists of training the parameters of an additional sigmoid to map the implicitly-supervised model scores into more accurate probabilities, based on the manually labeled data (Platt, 1999.) This pro-

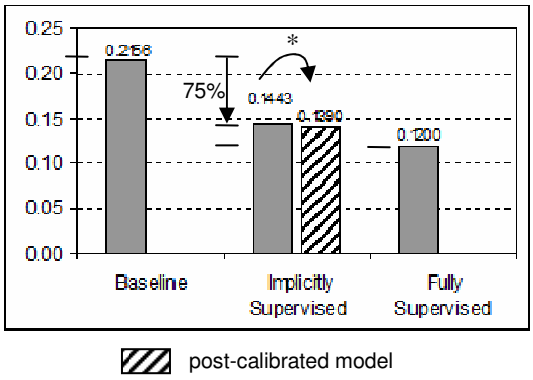


Figure 2. Implicitly- versus fully-supervised learning on Let's Go! Public data

cedure (based in our case on 100 randomly chosen labeled data-points) further increased the model's performance to 0.1390, therefore closing 80% of the gap between the baseline and fully supervised model. The difference between the un-calibrated and calibrated models is statistically significant (paired t-test, p=0.002).

The remaining performance gap between the implicitly and fully-supervised models is explained by the recall, accuracy and sampling bias of the implicit labels To better understand the effect of these factors on model performance, we constructed a number of additional models.

First, to distinguish between the effects of accuracy and recall, we constructed a model, dubbed **full-accuracy/same-recall (FA/SR)**. In training this model we only used the subset of utterances that were implicitly labeled (hence same-recall), but in conjunction with the manually obtained labels for these utterances (hence full-accuracy). The average test-set Brier score for this model was 0.1321, about half-way between the implicitly-supervised and fully-supervised models, with both differences statistically significant ($p<10^{-6}$) – see Figure 3. This result indicates that both the lack of recall and the lack of accuracy in the implicit labels contribute in roughly equal amounts to the observed performance gap.

Next, we constructed two additional models to investigate the effect of sampling bias on performance. (Recall that the subset of implicitly labeled utterances does not constitute a random sample for the entire corpus.) The first one of these models, **full-accuracy/random-same-recall (FA/RR)**, addresses the recall-bias issue and was trained with a randomly selected subset of utterances that has the same recall (size) as the implicitly labeled subset (hence random-same-recall). The second model, **random-same-accuracy/ same-recall (RA/SR)**, addresses the accuracy-bias issue. This model uses the utterances that were implicitly labeled (hence same-recall); the training labels were however constructed by starting from the reference labels and randomly altering them to attain the same accuracy level as the implicit labels have.

The performance of the full-accuracy/random-same-recall model, 0.1239, places it closer to the fully-supervised model (0.1200) than to the full-accuracy/same-recall-model (0.1321) – see Figure 3. Both differences are statistically significant in a paired t-test. The larger difference to the full-accuracy/same-recall model seems to indicate that the recall bias does affect performance in this case. On the other hand, the random-same-accuracy/ same-recall model performs similarly to the implicitly supervised model, in fact slightly worse (0.1475 versus 0.1443, no statistically significant

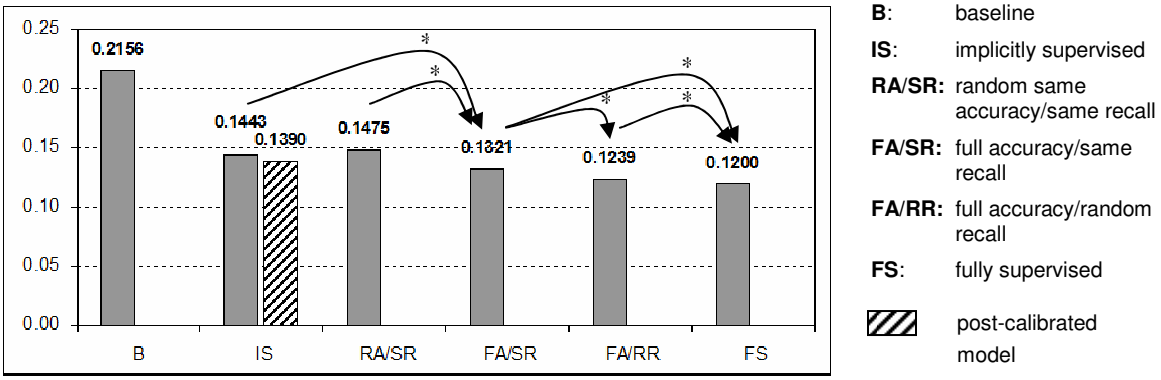


Figure 3. Implicitly- versus fully-supervised learning performance gap decomposition in Let's Go! Public domain (arrows with stars mark statistically significant differences, p<0.001)

difference detected). This result indicates that, at least in the Let's Go! Public system, the proposed implicitly generated labels do not exhibit a detrimental accuracy bias.

On a final note, recall that in Figure 2 we have seen that the implicitly-supervised approach closes 75% of the gap between the majority baseline and a fully-supervised approach (using the whole corpus). A comparison with the full-accuracy/random-same-recall model is more informative, because this model uses the same amounts of labeled data. Correcting for sample bias represents a difficult and interesting research problem (Zhang and Rudnicky, 2006). At the same time, we can easily envision using more data (since we don't need to manually label it.) As more data becomes available, the full-accuracy/random-same-recall model will eventually reach the performance of the fully supervised model. When compared to this model, the proposed implicitly-supervised approach closes 78% of the performance gap; the post-calibrated model closes 84% of this gap.

6.2 Results in the RoomLine domain

We now shift our attention to the RoomLine domain. Here, due to the more optimistic confirmation policy, the recall of the proposed implicit labeling scheme is lower: 10.8%. At the same time, due to better environmental conditions and less recognition errors, the accuracy is higher: 89.9%.

The results in this domain are illustrated in Figure 4. The implicitly-supervised approach again attains a significant improvement over the majority baseline. The relative improvement is smaller than the one attained in the Let's Go! Public domain. On the RoomLine corpus, the implicitly-supervised approach closes only 48% of the gap to the fully-supervised model; the post-calibrated model performs slightly better, but the improvement is not statistically significant. When compared to the full-accuracy/random-same-recall model, the implicitly supervised approach closes 59% of the gap (vs 78% in the Let's Go! Public domain.)

The lower performance on the RoomLine domain was expected due to the more optimistic confirmation policy and the resulting lower recall of the implicit labeling scheme. Overall, the RoomLine corpus contains 977 implicitly labeled training points, while the Let's Go! Public corpus contains more than double that amount. In the ideal case, in order to build a confidence annotation model using the proposed implicitly-supervised approach we would like the system to start with an always-confirm policy, like in the Let's Go! Public system. The full-accuracy/same-recall model (FA/SR in Figure 4), confirms that a significant part of the remaining performance gap is indeed explained by the lower recall. At the same time, part of the remaining performance gap is also explained by the lack of accuracy. This is somewhat surprising, since the accuracy is higher than in the Let's Go! Public domain. A possible explanation is that, when only small amounts of data are available for training, and/or when the class marginals are more skewed, precision plays a more important role.

Finally, the random-same-accuracy/same-recall and full-accuracy/random-same-recall models reveal that there is no detrimental sampling or recall bias in this domain. Like before, as the amount of training data increases, we can expect the gap between the full-accuracy/same-random-recall and fully-supervised model to decrease; further per-

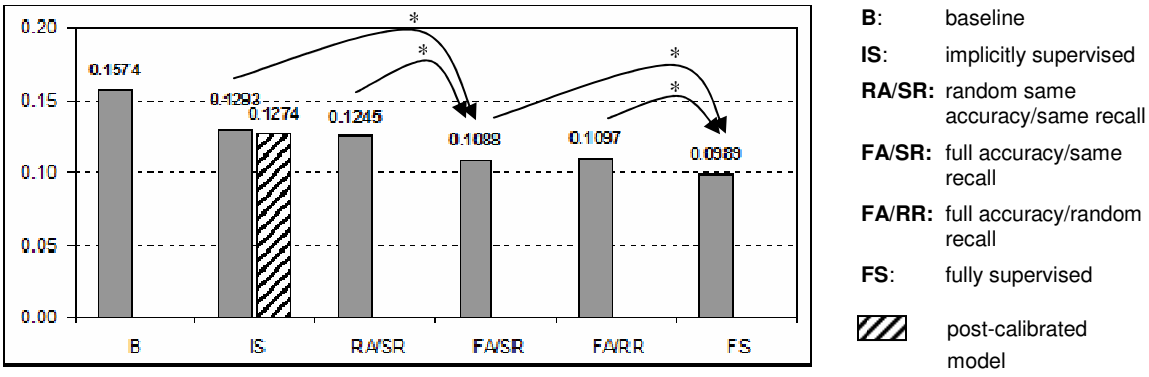


Figure 4. Implicitly- versus fully-supervised learning performance gap decomposition in RoomLine domain (arrows with stars mark statistically significant differences, $p<0.001$)

formance gains for the implicitly-supervised model are therefore expected, as we increase the dataset size. Experiments in which we trained the models using increasingly larger amounts of implicitly-labeled training data corroborate this conjecture (more details are presented in Appendix A.)

7 Discussion and future plans

While the empirical results we described in the previous section are very encouraging, they represent only a first step towards understanding the properties, advantages and limitations of the proposed implicitly-supervised paradigm.

So far, we have only performed a batch mode evaluation. However, apart from eliminating the need for a manually labeled corpus, a second important advantage of the implicitly-supervised approach is that it facilitates online learning and adaptation. The next question therefore is: how can a system engage in explicit confirmations in pursuit of its learning goals but without significantly disrupting the interaction? This is a control problem, where the system must balance the benefits of gaining knowledge via explicit confirmations against the costs potentially incurred by the user.

To some extent, dialog managers already have to solve similar trade-offs when deciding between different confirmation strategies, for instance between explicit or implicit confirmation. Explicit confirmations take an extra dialog turn, but the system has a better chance of understanding the follow-up user response, especially if the information to be confirmed is incorrect (Krahmer et al, 2001; Bohus and Rudnicky, 2005.) Typically, the costs are assumed to be known and are immediate. Solutions to these trade-off problems range from hard-coded heuristics to various offline corpus-based methods. In an online implicitly-supervised approach, the additional learning goals change the nature of the problem in two different ways. First, system actions not only create immediate dialog costs, but also produce knowledge that can be used to improve future performance. To address this new trade-off, the system must be able to assess the long-term benefits of the knowledge that stands to be gained. Secondly, in order to provide an online solution, systems should be able to continuously monitor their current performance and adjust their control policies, as their models improve.

Finally, another interesting question regards the knowledge-producing interaction pattern itself. In the experiments discussed above, the pattern consisted of user responses to system confirmation questions. Intuitively, other informative patterns could be found. For instance, if in a certain segment the dialog advances normally towards its goals, and no non-understandings occur, we might consider all those user turns correctly understood by the system. Alternatively, if a certain concept is corrected by the user at a later point in the dialog, we might mark the utterance from which the system extracted the first value for that concept as incorrect. We believe that an interesting avenue for future research is to develop techniques that allow systems to automatically discover such knowledge-producing interaction patterns.

The central idea in the proposed implicitly-supervised learning paradigm is therefore to acquire knowledge online, by discovering, eliciting and leveraging natural patterns that occur in interaction as a by-product of the collaboration between the system and an invested user. This paradigm can eliminate the need for developer supervision and can enable fast online adaptation and learning. We conjecture that it can supplement and or even provide a strong alternative to existing learning approaches, and enable significant autonomous learning in interactive systems.

The use of implicit feedback and human supervision for labeling, learning or adaptation purposes appears before in a number of other areas, like information retrieval (Brown and Claypool, 2003; Shen et al, 2005), image labeling (von Ahn and Dabbish), meeting segmentation (Banerjee and Rudnicky, 2007). To our knowledge, the work described in this paper is the first effort in learning from implicit supervision in the context of conversational spoken language interfaces. While in this paper we have focused only on one learning problem (i.e. building confidence annotation models), we believe that the proposed implicitly-supervised paradigm can be applied to a number of other problems in conversational spoken language interfaces. In fact, we have already developed and will soon report on an implicitly-supervised approach for learning how to automatically correct non-understanding errors in a spoken dialog system.

8 Conclusion

In this paper, we have proposed the use of an implicitly-supervised approach for learning in spoken language interfaces and have applied it for constructing confidence annotation models. Previous supervised learning solutions (Litman et al, 1999; Carpenter et al, 2001; San-Segundo et al, 2001; Hazen et al, 2002; Hirchberg et al, 2004.) rely on pre-existing, in-domain, manually labeled data and lead to static solutions. In contrast, the proposed approach does not require developer supervision. Instead, the system obtains the supervision signal from follow-up user responses to the system's explicit confirmation questions. In effect, the system learns from its own experiences.

We evaluated the proposed approach in two different dialog domains: RoomLine and Let's Go! Public. Empirical results confirm that a system can indeed successfully leverage interaction patterns to automatically construct a confidence annotation model that performs similarly to a fully-supervised model. The experiments we have reported here represent only a first step towards a fuller understanding of the proposed implicit-learning paradigm. The encouraging results we have obtained on the confidence annotation task point towards what we believe to be a very interesting research avenue. We conjecture that the proposed approach can be applied to address a number of other problems in conversational spoken language interfaces, and in interactive systems in general. Ultimately, we hope that it will enable the development of autonomously self-improving systems.

Acknowledgements

The authors would like to thank Antoine Raux for helpful discussions and suggestions during the early stages of this work, and Maxine Eskenazi and Alan W Black for making the Let's Go! Public corpus available for research purposes.

References

Banerjee, S., and Rudnicky, A. 2007. *Segmenting Meetings into Agenda Items by Extracting Implicit Supervision from Human Note-Taking*, in Proc. of IUI'07, Honolulu, Hawaii, USA.

Bohus, D., and Rudnicky, A. 2005. *Constructing Accurate Beliefs in Spoken Dialog Systems*, in Proc. of ASRU-2005, San Juan, Puerto Rico.

Bohus, D. 2007. *Error Awareness and Recovery in Conversational Spoken Language Interfaces*, Ph.D Thesis, Carnegie Mellon University, CS-07-124

Brown, D., and Claypool, M. 2003. *Curious Browsers: Automated gathering of implicit interest indicators by an instrumented browser*, in Workshop on Implicit Measures of User Interests and Preferences, SIGIR'2003, Toronto, Canada

Carpenter, P., Chun, J., Wilson, D., Zhang, R., Bohus, D., and Rudnicky, A. 2001. *Is this conversation on track?*, in Proc. of Eurospeech'99, Aalborg, Denmark

Cohen, I. and Goldszmidt, M., 2004 - *Properties and benefits of calibrated classifiers*. in Proc. of EMCL/PKDD. Pisa, Italy.

Hazen, T., Seneff, S., and Polifroni, J. 2002. *Recognition confidence scoring and its use in speech understanding systems*, Computer Speech and Language.

Hirchberg, J., Litman, D., and Swerts, M. 2004. *Prosodic and other cues to speech recognition failures*, Speech Communication, 2004.

Krahmer, E., Swerts, M., Theune, M., and Weegels, M., 2001. *Error Detection in Spoken Human-Machine Interaction*. International Journal of Speech Technology. 4(1): p. 19-30.

Litman, D., Walker, M., and Kearns, M. 1999. *Automatic Detection of Poor Speech Recognition at the Dialogue Level*, in Proc. of ACL'99, College Park, MD.

Myers, R. H., Montgomery, D. C., and Vining, G. 2001. *Generalized Linear Models: With Applications in Engineering and the Sciences*. Wiley series in probability and statistics, ed. Wiley-Interscience.

Platt, J. 1999. *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*, Advances in Large Margin Classifiers

Raux, A., Bohus, D., Langner, B., Black, A.W, and Eskenazi, M. 2006. *Doing Research in a Deployed Spoken Dialog Systems: One Year of Let's Go! Public Experience*, in Proc. of Interspeech'06, Pittsburgh, PA.

San-Segundo, R., Pellom, B., Hacioglu, K., and Ward, W. 2001. *Confidence Measures for Spoken Dialogue Systems*, in Proc. of ICASSP'01, Salt Lake City, UT

Shen, X., Tan, B., and Zhai, ChengXiang, 2005. *Context-Sensitive Information Retrieval using Implicit Feedback*, in Proc. of SIGIR'05, Salvador, Brazil

von Ahn, L., and Dabbish, L., 2004. *Labeling images with a computer game*, in CHI'04, New York, NY

Zhang, R. and Rudnicky, A. 2006. *A New Data Selection Approach for Semi-Supervised Acoustic Modeling*, in Proc. of ICASSP'06. Toulouse, France.

Appendix A. Performance as a function of training set size

We investigated the relationship between the performance of the implicitly-supervised confidence annotation models and the overall training set size. The results are shown in Figure 5.A for the RoomLine domain, and Figure 5.B for the Let's Go! Public domain.

In the RoomLine domain, the performance of the implicitly-supervised model does not yet reach an asymptote by the time we have considered the full training set (7537 utterances.) This result corroborates our previous conjecture that, if more data were available, further performance gains would be possible. As more data becomes available, the full-precision/random-same-recall model is guaranteed to reach the same asymptote as the fully supervised model. At the same time, we expect that the gap between the implicitly supervised method and the full-precision/random-same-recall model will stay roughly constant. As a consequence, we expect corresponding gains in the implicitly-supervised model performance.

Another interesting observation is that the random-same-precision/same-recall model closely tracks the implicitly supervised model, and the full-precision/random-same-recall model closely tracks the full-precision/same-recall model. These trends confirm that there is no detrimental sample bias (neither in terms of accuracy nor recall) in the proposed implicit learning scheme in the RoomLine data.

In the Let's Go! Public domain, the implicitly-supervised model seems to have reached a performance asymptote; this is not surprising, given the larger recall of the implicit labeling scheme in this domain. As the amount of data increases, the full-precision/random-same-recall model shows increasingly larger improvements over the full-precision/same-recall model.

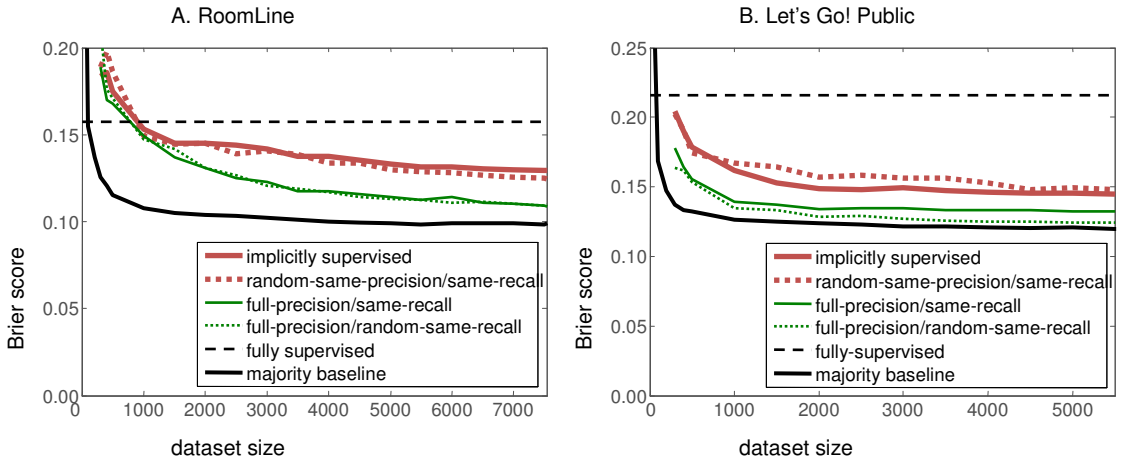


Figure 5. Implicitly supervised confidence annotation model performance as a function of training set size (in the RoomLine and Let's Go! Public domains)

Planning Dialog Actions

Mark Steedman
School of Informatics
University of Edinburgh
Edinburgh EH8 9LW, Scotland, UK
steedman@inf.ed.ac.uk

Ronald P. A. Petrick
School of Informatics
University of Edinburgh
Edinburgh EH8 9LW, Scotland, UK
rpetrick@inf.ed.ac.uk

Abstract

The problem of planning dialog moves can be viewed as an instance of the more general AI problem of planning with incomplete information and sensing. Sensing actions complicate the planning process since such actions engender potentially infinite state spaces. We adapt the Linear Dynamic Event Calculus (LDEC) to the representation of dialog acts using insights from the PKS planner, and show how this formalism can be applied to the problem of planning mixed-initiative collaborative discourse.

1 Introduction

Successful planning in dynamic domains often requires reasoning about sensing acts which, when executed, update the planner's knowledge state without necessarily changing the world state. For instance, reading a piece of paper with a telephone number printed on it may provide the reader with the prerequisite information needed to successfully complete a phone call. Such actions typically have very large, even infinite, sets of possible outcomes in terms of the actual sensed value, and threaten to make search impracticable. There have been several suggestions in the AI literature for how to handle this problem, including Moore (1985); Morgenstern (1988); Etzioni et al. (1992); Stone (1998); and Petrick & Bacchus (2002; 2004).

Stone (2000) points out that the problem of planning effective conversational moves is also a problem of planning with sensing or knowledge-producing actions, a view that is also implicit in early "beliefs, desires and intentions" (BDI) -based approaches (e.g., Litman & Allen (1987); Bratman, Israel & Pollack (1988); Cohen & Levesque (1990); Grosz & Sidner (1990)). Nevertheless, most work on dialog planning has in practice tended to segregate domain planning and discourse planning, treating the former as an AI black box, and capturing the latter in large state-transition machines mediated or controlled via a blackboard or "information state" representing mutual belief, updated by specialized rules more or less directly embodying some form of speech-act theory, dialog game, or theory of textual coherence (e.g., Lambert & Carberry (1991); Traum & Allen (1992); Green & Carberry (1994); Young & Moore (1994); Chu-Carroll & Carberry (1995); Matheson, Poesio & Traum (2000); Beun (2001)); Asher & Lascarides (2003); Maudet (2004)). Such accounts often lend themselves to optimization using statistical models (e.g., Singh et al. (2002)).

One of the ostensible reasons for making this separation is that *indirect* speech acts, i.e., achieving coherence via implicatures, abound in conversation. (For instance, Green and Carberry cite studies showing around 13% of answers to Yes/No questions are indirect.) Nevertheless, that very same ubiquity of the phenomenon suggests it is a manifestation of the same planning apparatus as the domain planner, and that it should not be necessary to construct a completely separate specialized planner for dialog acts.

This paper addresses the problem of dialog planning by applying techniques developed in the AI planning literature for handling sensing and incomplete information. To this end, we work with planning domains axiomatized in the language of the

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 265–272,
Antwerp, September 2007.

Linear Dynamic Event Calculus (LDEC), but extended with constructs inspired by the knowledge-level conditional planner PKS.

2 Linear Dynamic Event Calculus (LDEC)

The Linear Dynamic Event Calculus (LDEC) (Steedman, 1997; Steedman, 2002) is a logical formalism that combines the insights of the Event Calculus of Kowalski & Sergot (1986), itself a descendant of the Situation Calculus (McCarthy and Hayes, 1969), and the STRIPS planner of Fikes & Nilsson (1971), together with the Dynamic and Linear Logics developed by Girard (1987), Harel (1984), and others.

The particular dynamic logic that we work with here exclusively uses the deterministic "necessity" modality $[\alpha]$. For instance, if a program α computes a function f over the integers, then an expression like "$n \geq 0 \Rightarrow [\alpha](y = f(n))$" indicates that "in any situation in which $n \geq 0$, after every execution of α that terminates, $y = f(n)$." We can think of this modality as defining a logic whose models are Kripke diagrams, where accessibility between situations is represented by events defined in terms of the conditions which must hold before an event can occur (e.g., "$n \geq 0$"), and the consequences of the event that hold as a result (e.g., "$y = f(n)$").

Thus, *actions* (or *events*) in LDEC provide the sole means of change and affect the *fluents* (i.e., properties) of the world being modelled. Like other dynamic logics, LDEC does not use explicit situation terms to denote the state-dependent values of fluents, but instead, chains together finite sequences of actions using a *sequence* operator ";". For instance, $[\alpha_1; \alpha_2; \ldots; \alpha_n]$ denotes a sequence of n actions and $[\alpha_1; \alpha_2; \ldots; \alpha_n]\phi$ means that ϕ must necessarily hold after every execution of this sequence.

One of the novel features of LDEC is that it mixes two types of logical implication. Besides standard (or intuitionistic) implication $\Rightarrow$, LDEC follows Bibel et al. (1989) and others in using *linear* logical implication, denoted by the symbol $\multimap$. Linear implication extends LDEC's representational power and provides a solution to the *frame problem* (McCarthy and Hayes, 1969), as we'll see below.

An LDEC *domain* is formally described by a collection of axioms. For each action α, a domain includes an *action precondition axiom* of the form:

$$L_1 \wedge L_2 \wedge \ldots \wedge L_k \Rightarrow \mathit{affords}(\alpha),$$

where each L_i is a fluent or its negation (we discuss *affords* below), and an *effect axiom* of the form:

$$\{\mathit{affords}(\alpha)\} \wedge \phi \multimap [\alpha]\psi,$$

where ϕ and ψ are conjunctions of fluents or their negations. LDEC domains can also specify a collection of *initial situation axioms* of the form:

$$L_1 \wedge L_2 \wedge \ldots \wedge L_p,$$

where each L_i is a ground fluent literal. Finally, LDEC domains can include a set of background axioms (e.g., for defining the properties of other modal operators), and a set of simple state constraint axioms (e.g., for encoding inter-fluent relationships). We will not discuss the details of these axioms here.

Action precondition axioms specify the applicability conditions of actions using a special *affords* fluent. Effect axioms use linear implication to build certain "update rules" directly into the LDEC representation. In particular, the fluents of ϕ in the antecedent of an effect axiom are treated as consumable resources that are replaced by the fluents of ψ in the consequent when an action α is applied.[1] $\{\mathit{affords}(\alpha)\}$ means that it is not defined whether $\mathit{affords}(\alpha)$ still holds after α. All other fluents are unchanged. Thus, LDEC's use of linear implication builds a STRIPS-style (Fikes and Nilsson, 1971) treatment of action effects into the semantics of the language, which lets us address the frame problem without having to write explicit frame axioms.

Previous work has demonstrated LDEC's versatility as a language for modelling dialog, by introducing notions of speaker/hearer supposition and common ground (Steedman, 2006). This is achieved by defining a new set of modal operators of the form [X], that designate the participants in the dialog and provide a reference point for the shared beliefs that exist between those participants. For instance, [S] and [H] refer to the "speaker" and "hearer", respectively, while $[C_{SH}]$ refers to the common ground between speaker and hearer.[2] Using these modalities

[1] We treat consumed fluents as being made false.

[2] Additional participant modalities can be defined as needed. A set of LDEC background axioms is provided as part of a domain to govern the behaviour of these modalities.

we can write LDEC formulae that capture common propositions that arise in dialog. For instance, [S] p means "the speaker supposes p", [S] [H] p means "the speaker supposes that the hearer supposes p", and [C_{SH}] [X] p means "it is common ground between the speaker and hearer that X supposes p".

In this paper we extend LDEC even further. First, we recognize the need to model *knowledge* in LDEC, which is a necessary prerequisite for planning with sensing actions, including those needed for effective discourse. Second, we require that our extended representation lend itself to tractable reasoning, in order to facilitate a practical implementation. Finally, although LDEC supports classical plan generation through proof (Steedman, 2002), prior work has not addressed the problem of translating LDEC domains into a form that can take advantage of recent planning algorithms for reasoning with incomplete information and sensing. For a solution to these problems we turn to the PKS planner.

3 Planning with Knowledge and Sensing (PKS)

PKS (Planning with Knowledge and Sensing) is a knowledge-level planner that can build conditional plans in the presence of incomplete information and sensing (Petrick and Bacchus, 2002; Petrick and Bacchus, 2004). Unlike traditional approaches that focus on modelling the world state and how actions change that state, PKS works at a much higher level of abstraction: PKS models an agent's knowledge state and how actions affect that knowledge state.

The key idea behind the PKS approach is that the planner's knowledge state is represented using a first-order language. Since reasoning in a general first-order language is impractical, PKS employs a restricted subset of this language and limits the amount of inference it can perform. This approach differs from those approaches that use propositional representations (i.e., without functions and variables) over which complete reasoning is feasible, or works that attempt to represent complete sets of possible worlds (i.e., sets of states compatible with the planner's incomplete knowledge) using BDDs, Graphplan-like structures, clausal representations, or other such techniques.

What makes the PKS approach particularly novel is the level of abstraction at which PKS operates. By reasoning at the knowledge level, PKS can avoid some of the irrelevant distinctions that occur at the world level, which gives rise to efficient inference and plans that are often quite "natural". Although the set of inferences PKS supports is weaker than that of many possible-worlds approaches, PKS can make use of non-propositional features such as functions and variables, allowing it to solve problems that can be difficult for world-level planners.

Like LDEC, PKS is based on a generalization of STRIPS. In STRIPS, the world state is modelled by a single database. In PKS, the planner's knowledge state, rather than the world state, is represented by a set of five databases whose contents have a fixed, formal interpretation in a modal logic of knowledge. To ensure efficient inference, PKS restricts the types of knowledge (especially disjunctions) each database can model. We briefly describe three of these databases (K_f, K_v, and K_w) here.

K_f: This database is like a standard STRIPS database except that both positive and negative facts are stored and the closed world assumption is not applied. K_f can include any ground literal ℓ, where $\ell \in K_f$ means "ℓ is known". K_f can also contain knowledge of function values.

K_v: This database stores information about function values that will become known at execution time, such as the plan-time effects of sensing actions that return numeric values. During planning, PKS can use K_v knowledge of finite-range functions to build multi-way conditional branches into a plan. K_v function terms also act as "run-time variables"—placeholders for function values that will only be available at execution time.

K_w: This database models the plan-time effects of "binary" sensing actions. $\phi \in K_w$ means that at plan time the planner either knows ϕ or knows $\neg\phi$, and that at execution time this disjunction will be resolved. PKS uses such "know-whether" facts to construct binary conditional branches in a plan.

PKS also includes a database (K_x) of known "exclusive-or" disjunctions and a database (*LCW*) for modelling known instances of "local closed world" information (Etzioni et al., 1994).

Actions in PKS are modelled as queries and updates to the databases. *Action preconditions* are specified as a list of *primitive queries* about the state

of the databases: (i) Kp, is p known to be true?, (ii) $K_v t$, is the value of t known?, (iii) $K_w p$, is p known to be true or known to be false (i.e., does the planner know-whether p)?, or (iv) the negation of (i)–(iii). *Action effects* are described by a set of STRIPS-like *database updates* that specify the formulae to be added to and deleted from the databases. These updates capture the changes to the planner's knowledge state that result from executing the action.

Using this representation, PKS constructs plans by applying actions in a simple forward-chaining manner: provided an action's preconditions are satisfied by the planner's knowledge state, an action's effects are applied to form a new knowledge state. Conditional branches can be added to a plan provided the planner has K_w or (particular types of) K_v information. For instance, if the planner has K_w information about a formula p then it can add a binary branch to a plan. Along one branch, p is assumed to be known while along the other branch $\neg p$ is assumed to be known. PKS can also use K_v information to denote certain execution-time quantities in a plan. Planning continues along each branch until all branches satisfy the goal.

4 Planning Speech Acts with LDEC/PKS

Our approach to planning dialog acts aims to introduce certain features of PKS within LDEC, with the goal of generating plans using the PKS framework. In this paper we primarily focus on the representational issues concerning LDEC, and simply sketch our approach for completing the link to PKS.

The most important insight PKS provides is its action representation based on simple *knowledge primitives*: K/K_f "know", K_v "know value", and K_w "know whether". In particular, PKS's tractable treatment of this information—which underlies its databases and queries—is essential to its ability to build plans with incomplete knowledge and sensing.

In order to model similar conditions of incomplete information in LDEC, we introduce a set of PKS-style knowledge primitives into LDEC in the form of *knowledge fluents* (Demolombe and Pozos Parra, 2000). Knowledge fluents are treated as ordinary fluents but are understood to have particular meanings with respect to the knowledge state. For instance, in our earlier example of reading a piece of paper with a telephone number printed on it, we could use a knowledge fluent $KhavePaper$ to indicate that an agent knows it has the required piece of paper, $K_v phoneNumber$ to represent the result of reading the phone number from the paper (i.e., the agent "knows the value of the phone number"), and $K_w connected$ to denote the result of actually dialling the phone number (i.e., the agent "knows whether the call connected successfully").

In a dialog setting, we must also ground all knowledge-level assertions to particular participants in the dialog, or to the common ground. Otherwise, such references will have little meaning in a multi-agent context. Thus, we couple speaker/hearer modalities together with knowledge fluents to write LDEC expressions like $[\mathrm{S}]\,Kp$ — "the speaker knows p", $[\mathrm{H}]\,K_v t$ — "the hearer knows the value of t", or more complex expressions like $[\mathrm{C_{SH}}]\,[\mathrm{H}]\,K_w p$ — "it's common ground between the speaker and hearer that the hearer knows whether p".

Although we treat knowledge fluents as ordinary fluents in LDEC, we retain their knowledge-level meanings with respect to their use in PKS. Thus, knowledge fluents serve a dual purpose in LDEC. First, they act as queries for establishing the truth of particular knowledge-level assertions (e.g., an action precondition axiom like $[\mathrm{X}]\,Kp \Rightarrow \mathit{affords}(\alpha)$ means "if X knows p then this affords action α"). Second, they act as updates that specify how knowledge changes due to action (e.g., an effect axiom like $\{\mathit{affords}(\alpha)\} \multimap [\alpha][X]K_v t$ means "executing α causes X to come to know the value of t"). This correlation between LDEC and PKS is not a coincidence but one, we hope, that will let us use PKS as a target planner for LDEC domains.

We illustrate our LDEC extensions in the following domain axiomatization, which is sufficient to support planning with dialog acts.

4.1 Background Axioms

(1) $[\mathrm{X}]\,p \Rightarrow p$ Supposition Veridicality

(2) $[\mathrm{X}]\,\neg p \Rightarrow \neg\,[\mathrm{X}]\,p$ Supposition Consistency

(3) $\neg\,[\mathrm{X}]\,p \Rightarrow [\mathrm{X}]\,\neg\,[\mathrm{X}]\,p$ Negative Introspection

(4) $[\mathrm{C_{SH}}]\,p \Leftrightarrow ([\mathrm{S}]\,[\mathrm{C_{SH}}]\,p \wedge [\mathrm{H}]\,[\mathrm{C_{SH}}]\,p)$ Common Ground

(5) $[X]\,[C_{XY}]\,p \Rightarrow [X]\,p$

Common Ground Veridicality

4.2 Initial Facts

(6) a. "I suppose Bonnie doesn't know what train I will catch"
b. $[S] \neg [B]\, K_v train$

(7) a. "If I know what time it is, I know what train I will catch."
b. $[S]\, K_v time \Rightarrow [S]\, K_v train$

(8) a. "I don't know what train I will catch."
b. $[S] \neg K_v train$

(9) a. "I suppose you know what time it is."
b. $[S]\,[H]\, K_v time$

(10) a. "I suppose it's not common ground that I don't know what time it is."
b. $[S] \neg [C_{SH}] \neg [S]\, K_v time$

4.3 Rules

(11) a. "If X supposes p, and X supposes p is not common ground, X can tell Y p"
b. $[X]\,p \wedge [X] \neg [C_{XY}]\,p$
$\Rightarrow \mathit{affords}(\mathit{tell}(X, Y, p))$

(12) a. "If X tells Y p, Y stops not knowing it and starts to know it."
b. $\{\mathit{affords}(\mathit{tell}(X, Y, p))\} \wedge \neg [Y]\,p$
$\multimap [\mathit{tell}(X, Y, p)]\,[Y]\,p$

(13) a. "If X doesn't know p and X supposes Y does, X can ask Y about it."
b. $\neg [X]\,p \wedge [X]\,[Y]\,p$
$\Rightarrow \mathit{affords}(\mathit{ask}(X, Y, p))$

(14) a. "If X asks Y about p, it makes it common ground X doesn't know it"
b. $\{\mathit{affords}(\mathit{ask}(X, Y, p))\}$
$\multimap [\mathit{ask}(X, Y, p)]\,[C_{XY}] \neg [X]\,p$

Axioms (1) – (5) capture a set of standard assumptions about speaker/hearer modalities and common ground. In (3), we assume the presence of a negative introspection axiom, however, we do not require its full generality in practice.[3]

Axioms (6) – (10) specify a number of initial facts about speaker/hearer suppositions. In particular, (10) asserts a speaker supposition about common ground that illustrates the types of conclusions we typically require. These facts also include two K_v knowledge fluents, $K_v train$ and $K_v time$. As in PKS, these fluents act as placeholders for the values of known functions that can map to a wide range of possible values, but whose definite values may not be known at plan/reasoning time.

Rules (11) – (14) encode action precondition and effects axioms for two speech acts, *ask* and *tell*.

Using this axiomatization, we consider the task of constructing two dialog-based plans, as a problem of planning through proof.

4.4 Planning a Direct Speech Act

Goal: I need Bonnie to know which train I'll catch.

By speaker supposition, the hearer knows what time it is:

(15) $\Rightarrow [H]\, K_v time$ (*9b*); (*1*)

The speaker doesn't know what time it is:

(16) $\Rightarrow \neg [S]\, K_v time$ (*8b*); (*2*); (*7b*)

By speaker supposition, Bonnie doesn't know what train the speaker will catch:

(17) $\Rightarrow \neg [B]\, K_v train$ (*6b*); (*1*)

The speaker supposes it's not common ground with Bonnie as to what train the speaker will catch:

(18) $\Rightarrow [S] \neg [C_{SB}]\, K_v train$ (*8b*); (*2*); (*5*); (*3*); (*4*)

The situation affords $\mathit{ask}(S, H, K_v time)$:

(19) $\Rightarrow \mathit{affords}(\mathit{ask}(S, H, K_v time))$ (*16*); (*9b*); (*13b*)

After applying $\mathit{ask}(S, H, K_v time)$:

(20) $\Rightarrow [C_{SH}] \neg [S]\, K_v time$ (*19*); (*14b*)

The situation now affords $\mathit{tell}(H, S, K_v time)$:

(21) $\Rightarrow \mathit{affords}(\mathit{tell}(H, S, K_v time))$
(15); (20); (4); (5); (11*b*)

After applying $\mathit{tell}(H, S, K_v time)$:

(22) $\Rightarrow [S]\, K_v time$ (*21*); (*16*); (*12b*)

—which means I know what train I will catch:

(23) $\Rightarrow [S]\, K_v train$ (*22*); (*7b*)

The situation now affords $\mathit{tell}(S, B, K_v train)$

(24) $\Rightarrow \mathit{affords}(\mathit{tell}(S, B, K_v train))$ (*23*); (*18*); (*11b*)

After applying $\mathit{tell}(S, B, K_v train)$:

[3] The weaker property $[X] \neg p \Rightarrow [X] \neg [C_{XY}]\, p$ (which also follows from negative introspection) will typically suffice.

(25) $\Rightarrow$ [B] K_v*train* (*24*); (*17*); (*12b*)

4.5 Planning an Indirect Speech Act

The original situation also affords telling the hearer that I don't know the time:

(26) $\Rightarrow$ [S] $\neg$ [S] K_v*time* (*8b*); (*2*); (*7*); (*3*)

(27) $\Rightarrow$ [S] $\neg$ [C_{SH}] $\neg$ [S] K_v*time* (*10*)

(28) $\Rightarrow$ *affords*(*tell*(S, H, $\neg$ [S] K_v*time*)) (26); (27); (11*b*)

After saying "I don't know what time it is"—that is, applying the action *tell*(S, H, $\neg$ [S] K_v*time*),

(29) $\Rightarrow$ [C_{SH}] $\neg$ [S] K_v*time* (*14b*)

Since (29) is identical to (20), the situation again affords *tell*(H, S, K_v*time*), and the rest of the plan can continue as before.

Asking the time by saying "I don't know what time it is" would usually be regarded as an indirect speech act. Under the present account, both "direct" and "indirect" speech acts have effects that change the same set of facts about the knowledge states of the participants. Both involve inference. In some sense, there is no such thing as a "direct" speech act. In that sense, it is not surprising that indirect speech acts are so widespread: *all* speech acts are indirect in the sense of involving inference. Crucially, the plan does not depend upon the hearer identifying the fact that the speaker's utterance "I don't know what time it is" had the illocutionary force of a request or question such as "What time is it?".

From an axiomatic point of view, the above examples illustrate that the reasoning required to achieve the desired conclusions is straightforward—in most cases only direct applications of the domain axioms are used. Most importantly, we do not need to resolve knowledge-level conclusions like K_v*train* at this level of reasoning and, thus, do not require standard axioms of knowledge to reason about the formulae *within* the scope of $K/K_v/K_w$.

Direct manipulation of fluents like K_v*train* means that we can manage knowledge and sensing actions in a PKS-style manner in our account. For instance, the above plans result in the conclusion [S] K_v*time* as a consequence of the *ask* and *tell* actions. The particular effect of "coming to know the value" of *time* means that we should treat these actions as sensing actions. At the knowledge-level of abstraction, the effects of *ask* and *tell* are no different than the effect produced by reading a piece of paper to come to know a telephone number in our earlier example. This PKS-style use of knowledge fluents also opens up the possibility of constructing conditional plans and, ultimately, planning with PKS itself.

4.6 On So-called Conversational Implicature

The fact that we distinguish speaker suppositions about common ground from the hearer suppositions themselves means that we can include the following rules parallel to (11) and (12) without inconsistency:

(30) a. "X can always say p to Y"
b. $\Rightarrow$ *affords*(*say*(X, Y, p))

(31) a. "If X says p to Y, and Y supposes $\neg p$, then Y continues to suppose $\neg p$, and supposes that $\neg p$ is not common ground."
b. {*affords*(*say*(X, Y, p))} $\wedge$ [Y] $\neg p$
$\multimap$ [*say*(X, Y, p)][Y]$\neg p \wedge$ [Y] $\neg$ [C] $\neg p$

Speakers' calculations about what will follow from making claims about hearers' knowledge states extend to what will follow from making *false* utterances. To take a famous example from Grice, suppose that we both know that you have done me an unfriendly turn:

(32) a. "I know that you are not a good friend"
b. [S] $\neg$*friendship*(h) = *good*

(33) a. "You know that you are not a good friend"
b. [H] $\neg$*friendship*(h) = *good*

After applying *say*(S, H, *friendship*(h) = *good*), say by uttering the following:

(34) You're a fine friend!

the following holds:

(35) $\Rightarrow$ [H] $\neg$*friendship*(h) = *good*
$\wedge$ [H] $\neg$ [C] $\neg$*friendship*(h) = *good* (*32*); (*33*); (*31b*)

One might not think that getting the hearer to infer something they already know is very useful. However, if we assume a mechanism of attention, whereby things that are inferred become salient, then we have drawn their attention to their trespass. Moreover, the information state that we have brought them to is one that would normally suggest,

via rules like (11) and (12), that the hearer should tell the original speaker that they are not a fine friend. Of course, further reflection (via similar rules we pass over here) is likely to make the hearer unwilling to do so, leaving them few conversational gambits other than to slink silently and guiltily away. This of course is what the original speaker really intended.

4.7 A Prediction of the Theory

This theory explains, as Grice did not, why this trope is asymmetrical: the following is predicted to be an ineffectual way to make a hearer pleasantly aware that they have acted as a good friend:

(36) #You're a lousy friend!

It is counterproductive to make the hearer think of the key fact for themselves. Moreover, there is no reason for them not to respond to the contradiction. Unlike (34), this utterance is likely to evoke a vociferous correction to the common ground, rather than smug acquiescence to the contrary, parallel to the sheepish response evoked by (34).

5 Discussion

We have presented a number of toy examples in this paper for purposes of exposition: scaling to realistic domains will raise all the usual problems of knowledge representation that AI is heir to. However, the update effects (and side-effects) of discourse planning that we describe are general-purpose. They are entirely driven by the knowledge state, without recourse to specifically conversational rules, other than some very general rules of consistency maintenance in common ground. There is therefore some hope that conversational planning itself is of low complexity, and that any domain we can actually plan in, we can also plan conversations about.

According to this theory, illocutionary acts such as questioning and requesting are discourse subplans that are emergent from the general rules for maintaining consistency in the common ground and for manipulating knowledge-level information, such as the K_v formulae in our examples. Of course, for practical applications that require efficient execution, we can always memoize the proofs of such frequently-used sub-plans in the way that is standard in Explanation-Based Learning (EBL). For instance, by treating action sequences as "compound" actions in the planning process, we would be in effect compiling them into a model of dialog state-change of the kind that is common in practical dialog management. More importantly, the present work offers a way to derive such models automatically from first principles, rather than laboriously constructing them by hand.

In contrast to approaches that reject the planning model on complexity grounds, e.g., (Beun, 2001), our choice of a planner with limited reasoning capabilities and knowledge resources—conditions often cited as underlying human planning and dialog—aims to address such concerns directly. Furthermore, the specialized rules governing speech act selection in alternate approaches can always be adopted as planning heuristics guiding action choice, if existing planning algorithms fail to produce sufficient plans.

We have also argued that LDEC, extended with PKS-style knowledge primitives, is sufficient for planning dialog actions. Although we have motivated a correspondence between LDEC and PKS, we have not described how PKS planning domains can be formed from LDEC axioms. While some of the mechanisms needed to support a translation already exist—the compilation of LDEC rules into PKS queries and database updates is straightforward and syntactic—we have yet to extend PKS's inference rules to encompass speaker/hearer modalities, and formally prove the soundness of our translation. We are also exploring the use of PKS's *LCW* database to manage common ground as a form of closed world information. (For example, if a participant X cannot establish p as common ground then X should assume p is not common ground.) Finally, we require a comprehensive evaluation of our approach to assess its feasibility and scalability to more complex dialog scenarios. Overall, we are optimistic about our prospects for adapting PKS to the problem of planning dialog acts.

Acknowledgements

The work reported in this paper was partially funded by the European Commission as part of the PACO-PLUS project (FP6-2004-IST-4-27657), and by the NSF under grant number NSF-IIS-0416128.

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press, Cambridge.

Robbert-Jan Beun. 2001. On the generation of coherent dialogue. *Pragmatics and Cognition*, 9:37–68.

Wolfgang Bibel, Luis Farinas del Cerro, B. Fronhfer, and A. Herzig. 1989. Plan generation by linear proofs: on semantics. In *German Workshop on Artificial Intelligence - GWAI'89*, volume 216 of *Informatik-Fachberichte*, Berlin. Springer Verlag.

Michael Bratman, David Israel, and Martha Pollack. 1988. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355.

Jennifer Chu-Carroll and Sandy Carberry. 1995. Response generation in collaborative negotiation. In *Proceedings of ACL-95*, pages 136–143. ACL.

Philip Cohen and Hector Levesque. 1990. Rational interaction as the basis for communication. In Philip Cohen, Jerry Morgan, and Martha Pollack, editors, *Intentions in Communication*, pages 221–255. MIT Press, Cambridge, MA.

Robert Demolombe and Maria del Pilar Pozos Parra. 2000. A simple and tractable extension of situation calculus to epistemic logic. In *Proceedings of ISMIS-2000*, pages 515–524.

Oren Etzioni, Steve Hanks, Daniel Weld, Denise Draper, Neal Lesh, and Mike Williamson. 1992. An approach to planning with incomplete information. In *Proceedings of KR-92*, pages 115–125.

Oren Etzioni, Keith Golden, and Daniel Weld. 1994. Tractable closed world reasoning with updates. In *Proceedings of KR-94*, pages 178–189. Morgan Kaufmann Publishers.

Richard Fikes and Nils Nilsson. 1971. Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

Jean-Yves Girard. 1987. Linear logic. *Theoretical Computer Science*, 50:1–102.

Nancy Green and Sandra Carberry. 1994. A hybrid reasoning model for indirect answers. In *Proceedings of ACL-94*, pages 58–65. ACL.

Barbara Grosz and Candace Sidner. 1990. Plans for discourse. In Philip Cohen, Jerry Morgan, and Martha Pollack, editors, *Intentions in Communication*, pages 417–444. MIT Press, Cambridge, MA.

David Harel. 1984. Dynamic logic. In Dov Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604. Reidel, Dordrecht.

Robert Kowalski and Maurice Sergot. 1986. A logic-based calculus of events. *New Generation Computing*, 4:67–95.

Lynn Lambert and Sandra Carberry. 1991. A tripartite plan-based model of dialogue. In *Proceedings of ACL-91*, pages 47–54. ACL.

Diane Litman and James Allen. 1987. A plan recognition model for subdialogues in conversation. *Cognitive Science*, 11:163–200.

Colin Matheson, Massimo Poesio, and David Traum. 2000. Modeling grounding and discourse obligations using update rules. In *Proceedings of NAACL 2000, Seattle*.

Nicolas Maudet. 2004. Negotiating language games. *Autonomous Agents and Multi-Agent Systems*, 7:229–233.

John McCarthy and Patrick Hayes. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Bernard Meltzer and Donald Michie, editors, *Machine Intelligence*, volume 4, pages 473–502. Edinburgh University Press, Edinburgh.

Robert Moore. 1985. A formal theory of knowledge and action. In Jerry Hobbs and Robert Moore, editors, *Formal Theories of the Commonsense World*, pages 319–358. Ablex, Norwood, NJ. Reprinted as Ch. 3 of (Moore, 1995).

Robert Moore. 1995. *Logic and Representation*, volume 39 of *CSLI Lecture Notes*. CSLI/Cambridge University Press, Stanford CA.

Leora Morgenstern. 1988. *Foundations of a Logic of Knowledge, Action, and Communication*. Ph.D. thesis, NYU, Courant Institute of Mathematical Sciences.

Ronald P. A. Petrick and Fahiem Bacchus. 2002. A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of AIPS-02*, pages 212–221.

Ronald P. A. Petrick and Fahiem Bacchus. 2004. Extending the knowledge-based approach to planning with incomplete information and sensing. In *Proc. of ICAPS-04*, pages 2–11.

Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artifial Intelligence Research*, 16:105–133.

Mark Steedman. 1997. Temporality. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 895–938. North Holland/Elsevier, Amsterdam.

Mark Steedman. 2002. Plans, affordances, and combinatory grammar. *Linguistics and Philosophy*, 25:723–753.

Mark Steedman. 2006. Surface compositional semantics of intonation. *In submission*.

Matthew Stone. 1998. Abductive planning with sensing. In *Proceedings of AAAI-98*, pages 631–636, Menlo Park CA. AAAI.

Matthew Stone. 2000. Towards a computational account of knowledge, action and inference in instructions. *Journal of Language and Computation*, 1:231–246.

David Traum and James Allen. 1992. A speech acts approach to grounding in conversation. In *Proceedings of ICSLP-92*, pages 137–140.

R. Michael Young and Johanna D. Moore. 1994. DPOCL: a principled approach to discourse planning. In *Proceedings of the 7th International Workshop on Natural Language Generation*, pages 13–20.

Statistical User Simulation with a Hidden Agenda

Jost Schatzmann and Blaise Thomson and Steve Young
Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, United Kingdom
{js532,brmt2,sjy}@eng.cam.ac.uk

Abstract

Recent work in the area of probabilistic user simulation for training statistical dialogue managers has investigated a new *agenda*-based user model and presented preliminary experiments with a handcrafted model parameter set. Training the model on dialogue data is an important next step, but non-trivial since the user agenda states are not observable in data and the space of possible states and state transitions is intractably large. This paper presents a summary-space mapping which greatly reduces the number of state transitions and introduces a tree-based method for representing the space of possible agenda state sequences. Treating the user agenda as a hidden variable, the forward/backward algorithm can then be successfully applied to iteratively estimate the model parameters on dialogue data.

1 Introduction

1.1 Statistical user simulation

A key advantage of taking a statistical approach to dialogue manager (DM) design is the ability to formalise design criteria as objective reward functions and to learn an optimal dialogue policy from human-computer dialogue data (Young, 2002). The amount of suitably annotated in-domain data required for training a statistical system, however, typically exceeds the size of available dialogue corpora by several orders of magnitude and it is thus common practise to use a two-phased simulation-based approach. First, a statistical model of user behaviour is trained on the limited amount of available data. The trained model is then used to simulate any number of dialogues with the interactively learning dialogue manager (Levin et al., 2000; Scheffler and Young, 2002; Pietquin, 2004; Georgila et al., 2005; Lemon et al., 2006; Rieser and Lemon, 2006; Schatzmann et al., 2006).

1.2 Agenda-based user modelling

Recent work by Schatzmann et al. (2007) has presented a new technique for user simulation based on explicit representations of the *user goal* and the *user agenda*, which provide compact models of the dialogue context and the user's "state of mind" and are dynamically updated during the dialogue. Experimental results with the statistical POMDP-based Hidden Information State dialogue system (Young et al., 2007; Thomson et al., 2007) show that a competitive dialogue policy can be learnt even with handcrafted user model parameters.

1.3 Training on real data

While this result is useful for bootstrapping a prototype DM when no access to dialogue data is available, training the agenda-model on real human-computer dialogue data is an important next step. Training avoids the effort and expertise needed to manually set the model parameters and ensures that the learned system policy is optimized for human dialogue behaviour rather than the handcrafted simulator. The implementation of a suitable training algorithm for the agenda-based user model, however, is non-trivial since the user agenda and goal states are not observable in data. Moreover, the space of possible states and state transitions is intractably large.

1.4 Paper overview

This paper reviews the agenda-based user model (Section 2) and presents an Expectation-Maximization (EM)-based training method (Section 3) which models the observable dialogue data in terms of a sequence of hidden user states. Section 4 discusses the tractability problems associated with the vast state space and suggests a summary-space mapping for state transitions. Using an efficient tree-based method for generating state sequences on-the-fly, the forward/backward algorithm can then be applied to iteratively estimate the model parameters on data. Section 5 concludes with a brief evalution.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 273–282,
Antwerp, September 2007.

2 Agenda-based user simulation

2.1 User simulation at a semantic level

The agenda-based model introduced by Schatzmann et al. (2007) formalises human-machine dialogue at a semantic level as a sequence of states and dialogue acts[1]. At any time t, the user is in a state S, takes action a_u, transitions into the intermediate state S', receives machine action a_m, and transitions into the next state S'' where the cycle restarts.

$$S \rightarrow a_u \rightarrow S' \rightarrow a_m \rightarrow S'' \rightarrow \cdots \quad (1)$$

Assuming a Markovian state representation, user behaviour can be decomposed into three models: $P(a_u|S)$ for action selection, $P(S'|a_u, S)$ for the state transition into S', and $P(S''|a_m, S')$ for the transition into S''. Dialogue acts are assumed to be of the form *act(a=x, b=y,...)*, where *act* denotes the type of action (such as *hello*, *inform* or *request*) and act items *a=x* and *b=y* denote slot-value pairs, such as *food=Chinese* or *stars=5* as described in (Young et al., 2005).

2.2 State decomposition into goal and agenda

Inspired by agenda-based approaches to dialogue management (Wei and Rudnicky, 1999; Lemon et al., 2001; Bohus and Rudnicky, 2003) the user state is factored into an agenda A and a goal G.

$$S = (A, G) \qquad \text{and} \qquad G = (C, R) \quad (2)$$

During the course of the dialogue, the goal G ensures that the user behaves in a consistent, goal-directed manner. G consists of constraints C which specify the required venue, eg. "a centrally located bar serving beer", and requests R which specify the desired pieces of information, eg. "the name, address and phone number of the venue".

The user agenda A is a stack-like structure containing the pending user dialogue acts that are needed to elicit the information specified in the goal. At the start of the dialogue a new goal is randomly generated using the system database and the agenda is populated by converting all goal constraints into *inform* acts and all goal requests into *request* acts. A *bye* act is added at the bottom of the agenda to close the dialogue (cf. Fig. 5 in the Appendix.).

As the dialogue progresses the agenda is dynamically updated and acts are selected from the top of the agenda to form user acts a_u. In response to incoming machine acts a_m, new user acts are pushed onto the agenda and no longer relevant ones are removed. The agenda thus serves as a convenient way of tracking the progress of the dialogue as well as encoding the relevant dialogue history. Dialogue acts can also be temporarily stored when actions of higher priority need to be issued first, hence providing the simulator with a simple model of user memory (see Fig. 5 for an illustration). When using an n-gram based approach, by comparison, such long-distance dependencies between dialogue turns are neglected unless n is set to a large value, which in turn often leads to poor model parameters estimates.

Another, perhaps less obvious, advantage of the agenda-based approach is that it enables the simulated user to take the initiative when the dialogue is corrupted by recognition errors or when the incoming system action is not relevant to the current task. The latter point is critical for training statistical dialogue managers because policies are typically learned from a random start. The "dialogue history" during the early training phase is thus often a sequence of random dialogue acts or dialogue states that has never been seen in the training data. The stack of dialogue acts on the agenda enables the user model to take the initiative in such cases and behave in a goal-directed manner even if the system is not.

2.3 Action selection and state transition models

As explained in detail in (Schatzmann et al., 2007), the decomposition of the user state S into a goal G and an agenda A simplifies the models for action selection and state transition. Since the agenda (of length N) is ordered according to priority, with $A[N]$ denoting the top and $A[1]$ denoting the bottom item, forming a user response is equivalent to popping n items of the top of the stack. Using $A[N-n+1..N]$ as a Matlab-like shorthand notation for the top n items on A, the action selection model can be expressed as

$$P(a_u|S) = \delta(a_u, A[N- n + 1..N])P(n|A, G) \quad (3)$$

where $\delta(p, q)$ is 1 iff $p = q$ and zero otherwise.

The state transition models $P(S'|a_u, S)$ and $P(S''|a_m, S')$ are rewritten as follows. Letting A' denote the agenda after popping off a_u and using $N' = N - n$ to denote the size of A', we have

$$A'[i] \quad := \quad A[i] \quad \forall i \in [1..N']. \quad (4)$$

Using this definition of A' and assuming that the goal remains constant when the user executes a_u, the first state transition depending on a_u is entirely deterministic:

$$\begin{aligned} P(S'|a_u, S) &= P(A', G'|a_u, A, G) \\ &= \delta(A', A[1..N'])\delta(G', G). \end{aligned} \quad (5)$$

The second state transition based on a_m can be decomposed into *goal update* and *agenda update* modules:

$$\begin{aligned} &P(S''|a_m, S') \\ &= \underbrace{P(A''|a_m, A', G'')}_{\text{agenda update}} \underbrace{P(G''|a_m, G')}_{\text{goal update}}. \end{aligned} \quad (6)$$

[1]The terms *dialogue act* and *dialogue action* are used interchangeably here.

3 Model Parameter Estimation

3.1 The user state as a hidden variable

Estimating the parameters of the action selection and state transition models is non-trivial, since the goal and agenda states are not observable in training data.

Previous work on the state-based approach to statistical user simulation (Georgila et al., 2005; Lemon et al., 2006; Rieser and Lemon, 2006) has circumvented this problem by annotating training data with dialogue state information and conditioning user output on the observable dialogue state rather than the unobservable user state. While this simplifies the training process, providing the necessary annotation requires a considerable effort. If done manually, the process is often expensive and it can be difficult to ensure inter-annotator agreement. Using an automatic tool for dialogue state annotation (Georgila et al., 2005) can improve efficiency, but the development of the tool itself is a time-consuming process.

The parameter estimation approach presented here avoids the need for dialogue state annotation by modelling the observable user and machine dialogue acts in terms of a *hidden* sequence of agendas and user goal states. More formally, the dialogue data $\mathcal{D}$ containing dialogue turns 1 to T

$$\mathcal{D} = \{\mathbf{a_u}, \mathbf{a_m}\} = \{a_{m,1}, a_{u,1}..., a_{m,T}, a_{u,T}\} \quad (7)$$

is modelled in terms of latent variables

$$X = \{\mathbf{A}, \mathbf{G}\} \quad (8)$$

where

$$\mathbf{A} = \{A_1, A'_1, ..., A_T, A'_T\} \quad (9)$$

$$\mathbf{G} = \{G_1, G'_1, ..., G_T, G'_T\}. \quad (10)$$

Collecting the results from Section 2, and noting that from (5) the choice of n deterministically fixes A', the joint probability can hence be expressed as

$$P(X, D) = P(\mathbf{A}, \mathbf{G}, \mathbf{a_u}, \mathbf{a_m}) =$$

$$\prod_{t=1}^{T} P(n_t|A_t, G_t)P(A''_t|a_{m,t}, A'_t, G''_t)P(G''_t|a_{m,t}, G'_t). \quad (11)$$

The goal is to learn maximum likelihood (ML) values for the model parameter set θ such that the log likelihood

$$\mathcal{L}(\theta) = \log P(\mathcal{D}|\theta) = \log \sum_X P(X, \mathcal{D}|\theta) \quad (12)$$

is maximized

$$\theta_{ML} = \arg\max_\theta \mathcal{L}(\theta). \quad (13)$$

3.2 An EM-based approach

The direct optimization of $\mathcal{L}(\theta)$ is not possible, however, an iterative Expectation-Maximization (EM)-based approach (Dempster et al., 1977) can be used to find a (local) maximum of the latent variable model likelihood. Using Jensen's inequality, any distribution $q(X)$ can be used to obtain a lower bound on $\mathcal{L}(\theta)$

$$\mathcal{L}(\theta) =$$

$$\log \sum_X q(X) \frac{P(X, \mathcal{D}|\theta)}{q(X)} \geq \sum_X q(X) \log \frac{P(X, \mathcal{D}|\theta)}{q(X)}$$

$$\stackrel{\text{def}}{=} \mathcal{F}(q(X), \theta). \quad (14)$$

Since $\mathcal{L}(\theta)$ is always greater or equal to the "negative free energy" $\mathcal{F}(q(X), \theta)$ the problem of maximizing $\mathcal{L}(\theta)$ is equivalent to maximizing $\mathcal{F}(q(X), \theta)$. Starting from arbitrarily selected model parameters, EM iterates by alternating an E-step and an M-step.

During the E-step, the distribution $q^{(k)}(X)$ over the latent variables is estimated for fixed model parameters $\theta^{(k-1)}$

$$q^{(k)}(X) := \arg\max_{q(X)} \mathcal{F}(q(X), \theta^{(k-1)}). \quad (15)$$

It can be shown that this is achieved by setting

$$q^{(k)}(X) = P(X|\mathcal{D}, \theta^{(k-1)}). \quad (16)$$

Using Bayes rule and the law of total probability the RHS of Eq. 16 can be expressed as

$$P(X|\mathcal{D}, \theta^{(k-1)})$$

$$= \frac{P(\mathcal{D}|X, \theta^{(k-1)})P(X|\theta^{(k-1)})}{\sum_X P(\mathcal{D}|X, \theta^{(k-1)})P(X|\theta^{(k-1)})}. \quad (17)$$

Resubstituting (7) and (8) into (17) completes the E-step:

$$q^{(k)}(\mathbf{A}, \mathbf{G})$$

$$= \frac{P(\mathbf{a_u}, \mathbf{a_m}|\mathbf{A}, \mathbf{G}, \theta^{(k-1)})P(\mathbf{A}, \mathbf{G}|\theta^{(k-1)})}{\sum_{\mathbf{A},\mathbf{G}} P(\mathbf{a_u}, \mathbf{a_m}|\mathbf{A}, \mathbf{G}, \theta^{(k-1)})P(\mathbf{A}, \mathbf{G}|\theta^{(k-1)})}. \quad (18)$$

The M-step now optimizes $\mathcal{F}(q(X), \theta)$ with respect to θ whilst holding $q^{(k)}(X)$ fixed

$$\theta^{(t)} := \arg\max_\theta \mathcal{F}(q^{(k)}(X), \theta). \quad (19)$$

This is achieved by maximizing the auxiliary function

$$Q(\theta, \theta^{(k-1)}) = \sum_X P(X, \mathcal{D}|\theta^{(k-1)}) \log P(X, \mathcal{D}|\theta). \quad (20)$$

Substituting Eq. 11 into the above, differentiating with respect to θ and setting the result to zero, one arrives at the parameter reestimation formulae shown in Eqs. 21-23 in Fig. 1.

$$\hat{P}(n|A,G) = \frac{\sum_t P(A_t = A, G_t = G|\mathbf{a_u}, \mathbf{a_m}, \theta^{(k-1)})\delta(n_t, n)}{\sum_t P(A_t = A, G_t = G|\mathbf{a_u}, \mathbf{a_m}, \theta^{(k-1)})} \quad (21)$$

$$\hat{P}(A''|a_m, A', G'') = \frac{\sum_t P(A''_t = A'', A'_t = A', G''_t = G''|\mathbf{a_u}, \mathbf{a_m}, \theta^{(k-1)})\delta(a_{m,t}, a_m)}{\sum_t P(A'_t = A', G''_t = G''|\mathbf{a_u}, \mathbf{a_m}, \theta^{(k-1)})\delta(a_{m,t}, a_m)} \quad (22)$$

$$\hat{P}(G''|a_m, G') = \frac{\sum_t P(G''_t = G'', G'_t = G'|\mathbf{a_u}, \mathbf{a_m}, \theta^{(k-1)})\delta(a_{m,t}, a_m)}{\sum_t P(G'_t = G'|\mathbf{a_u}, \mathbf{a_m}, \theta^{(k-1)})\delta(a_{m,t}, a_m)} \quad (23)$$

Figure 1: Model parameter update equations for the action selection and agenda and goal state transition models. Note that $\delta(n_t, n)$ is one iff $n_t = n$ and zero otherwise. Similarly, $\delta(a_{m,t}, a_m)$ is one iff $a_{m,t} = a_m$ and zero otherwise.

4 Implementation

4.1 Tractability considerations

In the Hidden Information State (HIS) Dialogue System (Young et al., 2007) used for the experiments presented in this paper, the size of the user and machine dialogue action sets $\mathcal{U}$ and $\mathcal{M}$ is

$$|\mathcal{U}| \approx 10^3 \qquad \text{and} \qquad |\mathcal{M}| \approx 10^3. \quad (24)$$

Goals are composed of N_C constraints taken from the set of constraints $\mathcal{C}$, and N_R requests taken from the set of requests $\mathcal{R}$. Note that the ordering of constraints and requests does not matter, and there are no duplicate constraints or requests. Using typical values for goal specifications during previous HIS Dialogue System user trials (Thomson et al., 2007) the size of the goal state space can be estimated as

$$|\mathcal{G}| = \binom{|\mathcal{C}|}{N_C}\binom{|\mathcal{R}|}{N_R} = \binom{50}{4}\binom{8}{3} \approx 10^7. \quad (25)$$

The size of the agenda state space $\mathcal{A}$ depends on the number of unique user dialogue acts $|\mathcal{U}|$ as defined above and the maximum number N_A of user dialogue acts on the agenda. The maximum length of the agenda is a design choice, but it is difficult to simulate realistic dialogues unless it is set to at least $N_A = 8$. If fully populated, $\mathcal{A}$ therefore comprises the vast number of

$$|\mathcal{A}| = \frac{|\mathcal{U}|!}{(|\mathcal{U}| - N_A)!} \approx 10^{20}. \quad (26)$$

potential agenda states[2] and the number of parameters needed to model $P(A''|a_m, A', G'')$ is of the order

$$|\mathcal{A} \times \mathcal{M} \times \mathcal{A} \times \mathcal{G}| \approx 10^{50}. \quad (27)$$

[2]Note that the order of agenda items matters and that there are no duplicate items.

4.2 Agenda updates as a sequence of push actions

The estimates show that when no restrictions are placed on A'', the space of possible state transitions is vast. It can however be assumed that A'' is derived from A' and that each transition entails only a limited number of well-defined atomic operations (Schatzmann et al., 2007).

More specifically, the agenda transition from A' to A'' can be viewed as a sequence of push-operations in which dialogue acts are added to the top of the agenda. In a second "clean-up" step, duplicate dialogue acts, "empty" acts, and unnecessary *request()* acts for already filled goal request slots must be removed but this is a deterministic procedure so that it can be excluded in the following derivation for simplicity. Considering only the push-operations, the items 1 to N' at the bottom of the agenda remain fixed and the update model is rewritten as follows:

$$\begin{aligned} & P(A''|a_m, A', G'') \\ &= P(A''[1..N'], A''[N'+1..N'']|a_m, A'[1..N'], G'') \\ &= \delta(A''[1..N'], A'[1..N']) \\ &\quad \cdot \; P(A''[N'+1..N'']|a_m, G''). \end{aligned} \quad (28)$$

The second term on the RHS of Eq. 28 can now be further simplified by assuming that every dialogue act item (slot-value pair) in a_m triggers one push-operation. This assumption can be made without loss of generality, because it is possible to push an "empty" act (which is later removed) or to push an act with more than one item. The advantage of this assumption is that the known number M of items in a_m now determines the number of push-operations. Hence $N'' = N' + M$ and

$$\begin{aligned} & P(A''[N'+1..N'']|a_m, G'') \\ &= P(A''[N'+1..N'+M]|a_m[1..M], G'') \quad (29) \\ &= \prod_{i=1}^{M} P(\underbrace{A''[N'+i]}_{a_{push}} | \underbrace{a_m[i]}_{a_{cond}}, G'') \quad (30) \end{aligned}$$

The expression in Eq. 30 shows that each item $a_m[i]$ in the system act triggers one push operation, and that this

operation is conditioned on the goal. For example, given that the item *x=y* in $a_m[i]$ violates the constraints in G'', one of the following might be pushed onto A'': *negate()*, *inform(x=z)*, *deny(x=y, x=z)*, etc.

Let $a_{push} \in \mathcal{U}$ denote the pushed act $A''[N'+i]$ and $a_{cond} \in \mathcal{M}$ denote the conditioning dialogue act containing the single dialogue act item $a_m[i]$. Omitting the Dirac delta function in Eq. 28, the agenda update step then reduces to the repeated application of a *push transition model* $P(a_{push}|a_{cond}, G'')$. The number of parameters needed to model $P(a_{push}|a_{cond}, G'')$ is of the order

$$|\mathcal{U} \times \mathcal{M} \times \mathcal{G}| \approx 10^{13}. \quad (31)$$

While still large, this number is significantly smaller then the number of parameters needed to model unrestricted transitions from A' to A'' (cf. Eq. 27).

4.3 A summary space model for push transitions

To further reduce the size of the model parameter set and make the estimation of $P(a_{push}|a_{cond}, G'')$ tractable, it is useful to introduce the concept of a "summary space", as has been previously done in the context of dialogue management (Williams and Young, 2005). First, a function ϕ is defined for mapping the machine dialogue act $a_{cond} \in \mathcal{M}$ and the goal state $G'' \in \mathcal{G}$ from the space of machine acts $\mathcal{M}$ and goal states $\mathcal{G}$ to a smaller summary space Z_{cond} of "summary conditions"

$$\phi : \mathcal{M} \times \mathcal{G} \mapsto Z_{cond} \quad \text{with} \quad |\mathcal{M} \times \mathcal{G}| \gg |Z_{cond}|. \quad (32)$$

Secondly, a "summary push action" space Z_{push} is defined, which groups real user dialogue acts into a smaller set of equivalence classes. Using a function ω, summary push actions are mapped back to "real" dialogue acts

$$\omega : Z_{push} \mapsto \mathcal{U} \quad \text{with} \quad |Z_{push}| \ll |\mathcal{U}|. \quad (33)$$

Agenda state transitions can now be modelled in summary space using

$$P(a_{push}|a_{cond}, G'') \approx P(z_{push}|z_{cond}) \quad (34)$$

where $z_{push} \in Z_{push}$ and $z_{cond} \in Z_{cond}$ and

$$z_{cond} = \phi(a_{cond}, G'') \quad (35)$$
$$a_{push} = \omega(z_{push}). \quad (36)$$

For the experiments presented in this paper, 20 summary conditions and 20 summary push actions were defined, with examples shown in Fig 6. The total number of parameters needed to model $P(z_{push}|z_{cond})$ is therefore

$$|Z_{cond} \times Z_{push}| = 400. \quad (37)$$

The parameter set needed to model agenda transitions is now small enough to be estimated on real dialogue data.

4.4 Representing agenda state sequences

Given our estimate of $|\mathcal{A}| \approx 10^{20}$ for the size of the agenda state space, the direct enumeration of all states in advance is clearly intractable. The actual number of states needed to model a particular dialogue act sequence, however, is much smaller, since agenda transitions are restricted to push/pop operations and conditioned on dialogue context. The training algorithm can exploit this by generating state-sequences on-the-fly, and discarding any state sequence X for which $P(X, \mathcal{D}|\theta) = 0$.

A suitable implementation for this is found in the form of a dynamically growing agenda-tree, which allows agenda-states to be represented as tree-nodes and state transitions as branches. The tree is initialised by creating a root node containing an empty agenda and then populating the agenda according to the goal specification as explained in Sect. 2. However, since the initial ordering of dialogue acts on the agenda is unknown, all possible permutations of constraints and requests must be created, resulting in a row of $N_C! \cdot N_R!$ initial agendas (cf. Fig. 2).

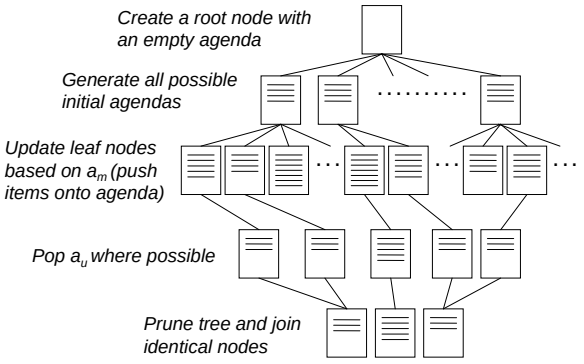


Figure 2: Tree-based method for representing state sequences.

4.4.1 Updating the tree based on a_m

The dialogue is now "parsed" by growing the tree and creating branches for all possible state sequences. Updates based on a machine dialogue act a_m involve mapping each item in a_m to its corresponding summary condition z_{cond} using the function ϕ. For each z_{cond} a list of summary push actions z_{push} is generated, discarding cases where $P(z_{push}|z_{cond}) = 0$. The summary push actions are then mapped back to real push actions using ω and used to create new agendas which are attached to the tree as new branches. The probability of the transition/branch is computed as the product of the probabilities of the real push actions. (See Fig. 6 in the appendix for a detailed illustration.)

The leaf nodes are now cleaned up in a deterministic procedure to remove empty and duplicate dialogue acts,

to delete all dialogue acts below a *bye()* act, and to remove all requests for items that have already been filled in the user goal. (An exception to the latter is made for requests that have just been added to the agenda, such that the simulated user can re-request filled items.)

4.4.2 Updating the tree based on a_u

In the next step, the tree is updated based on the observed user act a_u. This part simplifies to popping a_u from the top of the agenda wherever this is possible. Agendas which do not allow a_u to be popped off represent states with zero probability and can be discarded. In all other cases, a new node with the updated agenda is attached to the tree. The branch is marked as a pop-transition and its probability is computed based on the number of items popped.

4.4.3 Pruning the tree and joining identical nodes

Once the update based on a_u is completed, the tree is pruned to reduce the number of nodes and branches. First, all branches which were not extended during the dialogue turn, i.e. branches where a_u could not be popped off the leaf node agenda, are removed. All remaining branches represent possible sequences of agenda states with non-zero probability for the dialogue acts seen so far. In a second step, a more aggressive type of pruning can be carried out by removing all branches which do not have a given minimum leaf node probability. After pruning, the size of the tree is further reduced by joining nodes with identical agendas.

4.5 Action selection and goal update model

The action selection and goal update models experience similar tractability problems as the agenda update model, but in both cases a straightforward solution was found to produce satisfactory results. To simplify the action selection model $P(n|A, G)$, the random variable n can be assumed independent of A and G. The probability distribution $P(n)$ over small integer values for n (typically in the range from 0 to 6) can then be estimated directly from dialogue data by obtaining frequency counts of the number of dialogue act items in every user act.

The goal update model $P(G''|a_m, G')$ is decomposed into separate update steps for the constraints and requests. Assuming that R'' is conditionally independent of C' given C'' it is easy to show that

$$\begin{aligned} & P(G''|a_m, G') \\ &= P(R''|a_m, R', C'')P(C''|a_m, R', C'). \end{aligned} \quad (38)$$

The two update steps can be treated separately and implemented deterministically using two rules: 1) If R' contains an empty slot *u* and a_m is a dialogue act of the form *inform(u=v,r=s,...)*, then R'' is derived from R' by setting *u=v* given that no other information in a_m violates any constraints in C''. 2) If a_m contains a request for the slot *x*, a new constraint *x=y* is added to C' to form C''. The latter does not imply that the user necessarily responds to a system request for any slot *x*, since the agenda update model does not enforce a corresponding user dialogue act to be issued.

4.6 Applying the forward/backward algorithm

Using the summary space mapping for agenda transitions and simplifying assumptions for the goal update and action selection model, the parameter update equation set reduces to a single equation:

$$\hat{P}(z_{push}|z_{cond}) = \frac{\sum_k P(z_{push,k} = z_{push}, z_{cond,k} = z_{cond}|\mathbf{a_u}, \mathbf{a_m}, \theta)}{\sum_k P(z_{cond,k} = z_{cond}|\mathbf{a_u}, \mathbf{a_m}, \theta)} \quad (39)$$

Note that k is used here rather than t, since every dialogue turn t involves two state transitions, and there are hence $K = 2T$ observations and update steps.

The parameter update equation can now be efficiently implemented by applying the forward/backward algorithm. Let $\alpha_i(k)$ denote the forward probability of being in state i after seeing the observations from 1 to k, and let $\beta_i(k)$ denote the backward probability of seeing the observations from $k+1$ to K, given that we are in state i after update step k:

$$\alpha_i(k) = P(o_1, o_2, \ldots, o_k, x_k = i|\theta) \quad (40)$$

$$\beta_i(k) = P(o_{k+1}, o_{k+2}, \ldots, o_K|x_k = i, \theta) \quad (41)$$

Based on the observations, a tree of agendas is constructed as described in Section 4.4. After the last observation K, all agenda items have been popped, so that the leaf node agendas are empty and can be merged to form a single end node. The forward/backward probabilities are now initialised using

$$\alpha_i(1) = \frac{1}{N_C! N_R!} \quad , \quad 1 \leq i \leq N_C! N_R! \quad (42)$$

$$\beta_{end}(K) = 1 \quad (43)$$

and then recursively defined for the update steps from $k = 2$ to $k = K - 1$ using

$$\alpha_j(k) = \sum_i \alpha_i(k-1) a_{ij} \quad (44)$$

$$\beta_i(k) = \sum_j a_{ij} \beta_j(k+1) \quad (45)$$

where the transition probability a_{ij} of transitioning from state i to j depends on whether it is a push or a pop transition. When the transition involves popping n items off the agenda, a_{ij} equals $P(n)$. If the transition involves a

sequence of push actions, then a_{ij} is defined as the product of the probability of the associated real push actions (see Fig. 6 in the appendix for an illustration).

Using the forward/backward probabilities, one can now compute the probability $\tau_k(i,j)$ of transitioning from state i to state j at update step k as

$$\tau_k(i,j) = \frac{\alpha_i(k) a_{ij} \beta_j(k+1)}{\alpha_{end}(K)}. \quad (46)$$

Finally, the push transition model parameters are updated using

$$\hat{P}(z_{push}|z_{cond}) = \frac{\sum_{\{k,i,j|SPA=z_{push},SC=z_{cond}\}} \tau_k(i,j)}{\sum_{\{k,i,j|SC=z_{cond}\}} \tau_t(i,j)} \quad (47)$$

where the summation subscripts indicate if the summary push action (SPA) z_{push} and summary condition (SC) z_{cond} were used to transition from i to j at step k.

5 Evaluation

5.1 Dialogue training data

The parameter estimation approach presented in this paper was tested using a small corpus collected with the HIS Dialogue System (Young et al., 2007; Thomson et al., 2007; Schatzmann et al., 2007). The dataset consists of 160 dialogues from the tourist information domain, recorded with 40 different speakers, each of whom completed 4 dialogues. In total, the corpus contains 6452 dialogue turns and 21667 words. All utterances were manually transcribed and annotated using the set of dialogue act definitions described in Section 2.1. No dialogue state or user state annotation was needed.

5.2 Training results

The user model was trained on the dialogue corpus described above and Fig. 3 shows the number of agenda tree leaf nodes during a typical training episode on a sample dialogue. For each machine dialogue act, the tree is extended and 1 or more new nodes are attached to each tree branch, so that the number of leaf nodes stays constant or increases. Pop operations are then performed where possible, the tree is pruned and identical nodes are joined so that the number stays constant or decreases. At the end of the dialogue, only a single leaf node with an empty agenda remains.

When plotting the log probability of the data (Fig. 4), it can be seen that the EM-based algorithm produces a monotonically increasing curve (as expected). The algorithm quickly converges to a (local) optimum, so that in practise only a few iterations are needed. For illustration purposes, the training run in Fig. 4 was performed on two dialogues. As can be seen the log prob of the individual dialogues increases (top two lines), just as the log prob of the complete dataset (bottom line).

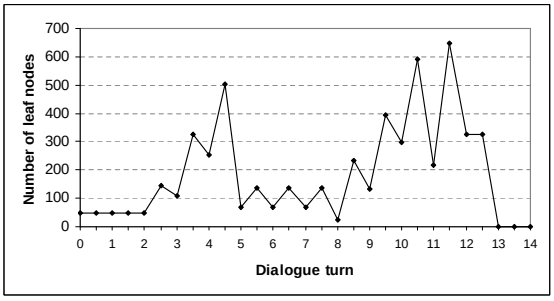


Figure 3: Graph showing the number of agenda tree leaf nodes after each observation during a training run performed on a single dialogue.

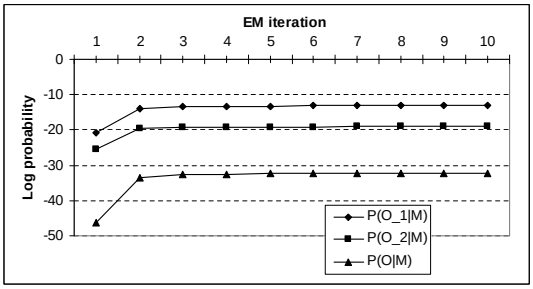


Figure 4: Graph showing a monotonous increase in log probability $\mathcal{L}(\theta)$ after each iteration of the EM algorithm.

5.3 Comparison of real and simulated data

An initial evaluation of the simulation quality has been performed by testing the similarity between real and simulated data. Table 1 shows basic statistical properties of dialogues collected with 1) real users, 2) the trained agenda model and 3) the handcrafted baseline simulator used by Schatzmann et al. (2007). All results were obtained with the same trained dialogue manager and the same set of user goal specifications. Since the model aims to reproduce user behaviour but not recognition errors, only the subset of 84 dialogues with a semantic accuracy above 90% was used from the real dialogue corpus[3]. The results show that the trained simulator performs better than the handcrafted baseline. The difference between the statistical properties of dialogues generated with the trained user model and those collected with real users is not statistically significant with confidence of more than 95%. Hence, based on these metrics, the trained agenda model appears to more closely match real human dialogue behaviour. One may expect that a dialogue system trained on this model is likely to perform better on real users than a system trained with the handcrafted simulator, but this is still an open research question.

[3]Semantic accuracy was measured in terms of substitution, insertion and deletion errors as defined by Boros et al. (1996).

	Real Users	Tr. Sim	Hdc. Sim
Sample size	84	1000	1000
Dial. length	3.30±0.53	3.38±0.07	4.04±0.19
Compl. rate	0.98±0.03	0.94±0.02	0.93±0.02
Performance	16.23±1.01	15.32±0.34	14.65±0.50

Table 1: Comparison of basic statistical properties of real and simulated dialogue data (mean±95% confidence thresholds). Dialogue length is measured in turns, task completion rate is based on the recommendation of a correct venue, and dialogue performance is computed by assigning a 20 point reward for a successful recommendation (0 otherwise) and subtracting 1 point for every turn.

6 Summary

This paper has extended recent work on an agenda-based user model for training statistical dialogue managers and presented a method for estimating the model parameters on human-computer dialogue data. The approach models the observable dialogue acts in terms of a sequence of hidden user states and uses an EM-based algorithm to iteratively estimate (locally) optimal parameter values.

In order to make estimation tractable, the training algorithm is implemented using a summary-space mapping for state transitions. Agenda state sequences are represented using tree structures, which are generated on-the-fly for each dialogue in the training corpus. Experimental results show that the forward/backward algorithm can be successfully applied to recompute the model parameters.

A comparison of real and simulated dialogue data has shown that the trained user model outperforms a handcrafted simulator and produces dialogues that closely match statistical properties of real data. While these initial results are promising, further work is needed to refine the summary state mapping and to fully evaluate the trained model. We look forward to reporting these results in a future paper.

References

D. Bohus and A. Rudnicky. 2003. Ravenclaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proc. of Eurospeech*. Geneva, Switzerland.

M. Boros, W. Eckert, F. Gallwitz, G. Gorz, G. Hanrieder, and H. Niemann. 1996. Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. In *Proc. of ICSLP*. Philadelphia, PA.

A. Dempster, N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

K. Georgila, J. Henderson, and O. Lemon. 2005. Learning user simulations for information state update dialog systems. In *Proc. of Eurospeech*. Lisbon, Portugal.

O. Lemon, A. Bracy, A. Gruenstein, and S. Peters. 2001. The WITAS multi-modal dialogue system I. In *Proc. of Eurospeech*. Aalborg, Denmark.

O. Lemon, K. Georgila, and J. Henderson. 2006. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Eval. In *Proc. of SLT*, Palm Beach, Aruba.

E. Levin, R. Pieraccini, and W. Eckert. 2000. A Stochastic Model of Human-Machine Interaction for Learning Dialog Strategies. *IEEE Trans. on Speech and Audio Processing*, 8(1):11–23.

O. Pietquin. 2004. *A Framework for Unsupervised Learning of Dialogue Strategies*. Ph.D. thesis, Faculte Polytechnique de Mons.

V. Rieser and O. Lemon. 2006. Cluster-based User Simulations for Learning Dialogue Strategies. In *Proc. of ICSLP*, Pittsburgh, PA.

J. Schatzmann, K. Weilhammer, M.N. Stuttle, and S. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *KER*, 21(2):97–126.

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Proc. of HLT/NAACL*. Rochester, NY.

K. Scheffler and S. Young. 2002. Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning. In *Proc. of HLT*. San Diego, CA.

B. Thomson, J. Schatzmann, K. Weilhammer, H. Ye, , and S. Young. 2007. Training a real-world POMDP dialogue system. In *Proc. of HLT/NAACL Workshop: Bridging the Gap*. Rochester, NY.

X. Wei and A.I. Rudnicky. 1999. An agenda-based dialog management architecture for spoken language systems. In *Proc. of IEEE ASRU*. Seattle, WA.

J. D. Williams and S. Young. 2005. Scaling Up POMDPs for Dialog Management: The "Summary POMDP" Method. In *Proc. of ASRU*. San Juan, Puerto Rico.

S. Young, J. Williams, J. Schatzmann, M. Stuttle, and K. Weilhammer. 2005. The hidden information state approach to dialogue management. Technical Report CUED/F-INFENG/TR.544, Cambridge University.

S. Young, J. Schatzmann, K. Weilhammer, and H. Ye. 2007. The Hidden Information State Approach to Dialog Management. In *Proc. of ICASSP*, Honolulu, HI.

S. Young. 2002. Talking to machines (statistically speaking). In *Proc. of ICSLP*. Denver, CO.

7 Appendix

7.1 Sample dialogue and user state sequence

Initialisation *(Generate goal constraints and requests and populate the agenda)*

$$C_0 = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \end{bmatrix} \quad R_0 = \begin{bmatrix} name = \\ addr = \\ phone = \end{bmatrix} \quad A_0 = \begin{bmatrix} inform(type = bar) \\ inform(drinks = beer) \\ inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Sys 0 Hello, how may I help you? *(Push 0 items onto the agenda)*
Usr 1 I'm looking for a nice bar serving beer. *(Pop 2 items off the agenda)*

$$C_1' = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \end{bmatrix} \quad R_1' = \begin{bmatrix} name = \\ addr = \\ phone = \end{bmatrix} \quad A_1' = \begin{bmatrix} inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Sys 1 Ok, a wine bar. What price range? *(Add 1 constraint, push 2 items onto the agenda)*

$$C_2 = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \\ prange = cheap \end{bmatrix} \quad R_2 = \begin{bmatrix} name = \\ addr = \\ phone = \end{bmatrix} \quad A_2 = \begin{bmatrix} negate(drinks = beer) \\ inform(prange = cheap) \\ inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Usr 2 No, beer please! *(Pop 1 item off the agenda)*
Sys 2 A bar serving beer, correct? *(Push 1 item onto the agenda)*

$$C_3 = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \\ prange = cheap \end{bmatrix} \quad R_3 = \begin{bmatrix} name = \\ addr = \\ phone = \end{bmatrix} \quad A_3 = \begin{bmatrix} affirm() \\ inform(prange = cheap) \\ inform(area = central) \\ request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Usr 3 Yeah something cheap in the town centre. *(Pop 3 items off the agenda)*

$$C_3' = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \\ prange = cheap \end{bmatrix} \quad R_3' = \begin{bmatrix} name = \\ addr = \\ phone = \end{bmatrix} \quad A_3' = \begin{bmatrix} request(name) \\ request(addr) \\ request(phone) \\ bye() \end{bmatrix}$$

Sys 3 Murphy's on Main Sq. serves cheap beer. *(Fill 2 requests, delete agenda requests for filled slots)*

$$C_4 = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \\ prange = cheap \end{bmatrix} \quad R_4 = \begin{bmatrix} name = Murphy's \\ addr = Main\ Sq \\ phone = \end{bmatrix} \quad A_4 = \begin{bmatrix} request(phone) \\ bye() \end{bmatrix}$$

Usr 4 And what's the phone number? *(Pop 1 item off the agenda)*
Sys 4 The number is 796 69 94. *(Fill 1 request)*

$$C_5 = \begin{bmatrix} type = bar \\ drinks = beer \\ area = central \\ prange = cheap \end{bmatrix} \quad R_5 = \begin{bmatrix} name = Murphy's \\ addr = Main\ Sq \\ phone = 7966994 \end{bmatrix} \quad A_5 = \begin{bmatrix} bye() \end{bmatrix}$$

Usr 5 Thanks, goodbye! *(Pop 1 item off the agenda)*

Figure 5: Sample dialogue showing the state of the user goal and agenda. Note that system turn 1 *"What price range?"* triggers the user act *inform(prange=cheap)* to be pushed onto the agenda but it is not executed until turn 3 because *negate(drinks=beer)* is issued first.

7.2 Sample agenda update transition using the summary-space mapping

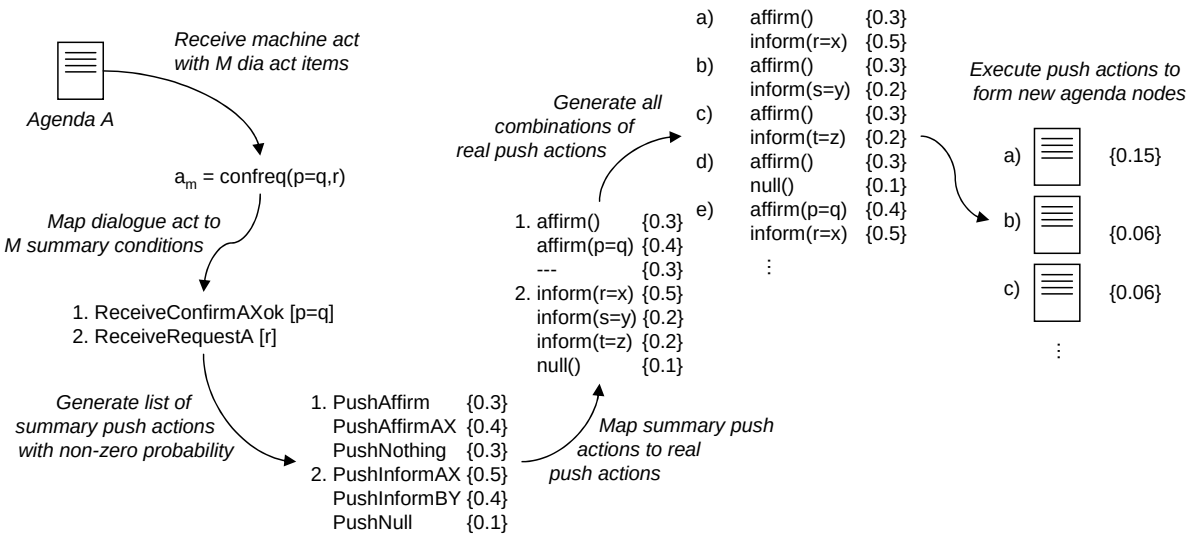


Figure 6: Simplified example illustrating the summary space technique for agenda updates.

The incoming machine act in this example is assumed to be a_m =*confreq(p=q,r)*, i.e. an implicit confirmation of the slot-value pair *p=q* and a request for the slot *r*. The update step proceeds as follows:

1. Based on the current state of the goal (not shown here), the first step is to map each dialogue act item (slot-value pair) to a summary condition z_{cond}. Given that the confirmation *p=q* in the example does not violate any of the constraints in the user goal, it is mapped to *ReceiveConfirmAXok[p=q]*. The request for *r* is mapped to *ReceiveRequestA[r]*.

2. A list of summary push actions z_{push}, each with probability $P(z_{push}|z_{cond})$, is now generated for each summary condition z_{cond}. A (shortened) list of examples is shown in the figure. The summary push action *PushInformAX*, for instance, implies that an *inform* act with the requested slot (in this case *r*) is pushed onto the agenda. Note that summary push actions with zero probability can be discarded at this point.

3. The summary push actions are now mapped to real push actions. This is a 1-to-1 mapping for most summary push actions, but some summary push actions can map to several real push actions. This is illustrated in the figure by the summary push action *PushInformBY*, which implies that the corresponding real push action is an *inform* dialogue act containing some slot-value pair *B=Y* other than the requested slot, in this case *s=y* or *t=z*. In such cases, the probability mass is split evenly between the real push actions for a summary push action, as shown in the figure.

4. Using one real push action from each summary condition, a list of all possible combinations of push actions is now generated. Each combination represents a series of dialogue acts to be pushed onto the agenda. As shown in the figure, each combination is used to create a new agenda. The transition probability is computed as the product of the real push actions that were used to make the transition.

Note that the set of summary conditions and summary actions is independent of the number of concepts and database entries, allowing the method to scale to more complex problem domains and larger databases.

An empirically based computational model of grounding in dialogue

Harry Bunt, Roser Morante and Simon Keizer
Department of Communication and Information Sciences
Tilburg University
{r.morante, harry.bunt,s.keizer@uvt.nl}

Abstract

In this paper we present a simple, empirically grounded computational model of grounding in dialogue. Grounding is shown to occur as a result of the dynamics of the information states of dialogue participants. A step-by-step analysis and representation of how information states develop through dialogue utterance processing illustrates exactly how this works.

1 Introduction

In an information-state update (ISU) approach, a dialogue is viewed as a sequential structure consisting of communicative acts that the participants perform in order to change each other's information state. For example, consider the following dialogue at a railway station between traveler A and employee B of the railway company:

(1) 1. A: Excuse me, can you tell me what time the next train to Amsterdam leaves?
2. B: Yes, that's at 9:17.
3. A: And at which platform is that?
4. B: That's at platform 5.
5. A: Thanks a lot.
6. B: You're welcome.

The second utterance tells A, among other things, that B believes that the next train to Amsterdam leaves at 9:17. Let us call this information p. Assuming that employees of the railway company provide correct information about train departure times, A will adopt the belief that p. So both participants now believe that p, and A also believes that B believes that p. After utterance 3, B will moreover believe that A has come to believe that p, although nothing is said about that. The dialogue continues on the topic of departure platform, which would seem not to influence A's and B's beliefs relating to p. So at the end of the dialogue we have the following situation with respect to the information p:

(2) a. A believes that p; B believes that p;
b. A believes that B believes that p; B believes that A believes that p.

In a shallow sense, p has become a shared belief: both participants have this belief and they both believe that the other has that belief. But studies of the logical foundations of communication tell us that participants in a dialogue should establish a *common ground* in a deeper sense. In their groundbreaking studies of common ground, Stalnaker and Lewis, among others, have suggested to define common ground in terms of *mutual beliefs*, explained as follows:

(3) p is a mutual belief of A and B iff:
- A and B believe that p;
- A and B believe that A and B believe that p;
- A and B believe that A and B believe that A and B believe that p;
and so on *ad infinitum*.

Clearly, the situation represented in (2) is a very poor approximation of this notion of common ground. Yet, intuitively, at the end of dialogue (1) the information that the next train to Amsterdam leaves at 9:17 seems to be *grounded*, i.e. to have been added to the common ground of A and B.

A technical problem presents itself here: the communicative acts expressed by the dialogue utterances create only finite iterations of belief of one dialogue participant about the beliefs of the other participant, as illustrated by (2); the full recursive nature of mutual beliefs cannot be achieved in this way in a dialogue of finite length.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 283–290,
Antwerp, September 2007.

In this paper we will describe a computational model of grounding where the establishment of common ground comes out as a consequence of successful communication, def ned as the recognition of each other's intentions, plus two pragmatic principles, one concerning the way in which dialogue participants deal with expectations of being understood and believed; and one about the cumulative effects of feedback. The model, which does not require any specif c grounding acts, is backed up by empirical observations from corpora of information-seeking and assistance dialogues.

This paper is organised as follows. Section 2 summarizes some existing views on grounding. Section 3 presents the conceptual model of grounding, based on dialogue analysis according to the framework of Dynamic Interpretation Theory (DIT, (Bunt, 2000)); section 4 presents our computational model of grounding, and Section 5 ends with concluding remarks.

2 Common Ground and Grounding

In Clark and Schaefer's model of grounding (Clark and Schaefer, 1989), participants in a dialogue try to establish for each utterance the mutual belief that the addressees have understood what the speaker meant. This is accomplished by the use of units called *contributions*. Contributions are divided into an acceptance and a presentation phase, so that every contribution, except for those that express negative evidence, has the role of accepting the previous contribution. A diff culty with this model is that its grounding criterion says that *"the contributor and the partners mutually believe that the partners have understood what the contributor meant"*. So the grounding *process* is conceived in terms of mutual beliefs. However, the central problem of grounding is precisely how mutual beliefs are established. Work based on this model includes its extension to human–computer interaction by Brennan and collaborators (Brennan, 1998; Cahn and S. E. Brennan, 1999), Li et al.'s model for multimodal grounding (Li et al., 2006), and Paek and Horvitz's formal theory of grounding (Paek and Horvitz, 2000).

In his inf uential computational model of grounding, Traum (1994) has introduced separate *grounding acts* which are used to provide communicative feedback and thereby create mutual beliefs. For this approach to work, Traum assumes that feedback acts are always correctly perceived and understood, therefore a dialogue participant does not need feedback about his feedback acts. This is an unwarranted assumption, however. Like any dialogue utterance, an utterance which expresses feedback can suffer from the addressee temporarily being disturbed by the phone, or by an aircraft f ying over, or by noise on a communication channel; hence a speaker who performs a grounding act can never be sure that his act was performed successfully until he has received some form of feedback. A limitation and somewhat confusing aspect of this model is that it discusses the grounding of *utterances*, rather than the grounding of information conveyed by utterances through their semantic content.

(Matheson et al., 2000) use elements of Traum's model in their treatment of grounding from the Information State Update perspective. They represent grounded and ungrounded discourse units in the information state, and change their status from ungrounded to grounded through grounding acts. The dialogue act `Acknowledgement` is the only grounding act implemented; its main effect is to merge the information in the acknowledged discourse unit into the grounded information. They do not deal with cases of misunderstandings or cases where the user asks for acknowledgement. The model keeps only the last two utterances in the information state, so it is not clear what would happen if the utterance to be grounded is more than two utterances back – which we will argue to be the rule rather than the exception.

3 Grounding and Belief Strengthening

The addition of something to a common ground relies on evidence that the belief in question is mutually believed. The nature of such evidence depends on the communicative situation, for instance on whether the participants can see each other, and on whether they are talking about something they (both know that they) can both see. We restrict ourselves here to situations where grounding is achieved through verbal communication only, as in the case of telephone conversations, email chats, or spoken human-computer dialogue.

In the DIT framework, information can pass from one dialogue participant to another through mech-

anisms linked to understanding and believing each other. The first of these consists of the information state of the addressee of a dialogue act undergoing certain changes when he understands the corresponding dialogue behaviour. Understanding communicative behaviour is modeled as the addressee coming to believe that the preconditions hold which are characteristic for the dialogue acts that are expressed by that behaviour. For example, if A asks B a Yes/No-Question about a proposition p, then as a result of understanding this, B will know that A wants to know whether p, and that A thinks that B knows whether p. The second mechanism is that of belief adoption (a.k.a. 'belief transfer', Allen and Perrault, 1980). When A has asked B whether p, and B answers "Yes", then upon understanding this A will assume that B believes that p. In such a situation, A may be expected to believe B, so A also believes that p: he has *adopted* p.

To be sure that information is indeed transferred through the mechanisms of understanding and/or adoption, a speaker needs evidence of correct understanding of his communicative behaviour and of being believed. Feedback, positive or negative, provides information about an addressee's understanding and adoption of information.

Let us consider the transfer of information through understanding and adoption in some more detail, to see its contribution to grounding processes. In the following dialogue fragment, A initially contributes utterance du_1 which expresses an `Inform` act; let c_1 be the precondition that A *believes that* p, with p the propositional content of the act (the information that the next train is at 11:02). Successful communication should lead to c_1 as well as p at some point being in A's and B's common ground.

(4) du_1. A: *The next train is at 11:02.*
du_2. B: *At 11:02.*
du_3. A: *That's correct.*
du_4. B: *Okay thanks.*

How could A for example come to believe that p is mutually believed? First, he should have evidence that B understands his utterance du_1 and believes its content p. B's utterance du_2 can be taken to provide such evidence. So after du_2, A believes that B believes that p, and that B believes that A believes that p. However, A cannot be certain that B indeed believes that p, since in du_2 he also seems to offer that belief for confirmation. A's response du_3 gives that confirmation. At this point A does not yet know whether his utterance has reached B and was well understood. B's next contribution du_4 provides evidence for that; upon understanding du_4, A has accumulated the following beliefs:

(5) A believes that p
A believes that B believes that p
A believes that B believes that A believes that p
A believes that B believes that A believes that B believes that p
A believes that B believes that A believes that B believes that A believes that p

Although we see nested beliefs of some depth emerging, A is still a long way from believing that p is mutually believed – an infinitely long way, in fact. Clearly, continuing along this line could not lead to mutual beliefs in a finite amount of time. We therefore want to suggest a different explanation.

In natural face-to-face dialogue, speakers receive feedback while they are speaking as the participants give explicit and implicit feedback about their understanding of what is being said by means of facial expressions, head movements, direction of gaze, and verbal elements. In situations without visual contact, such as telephone dialogues or computer-mediated chatting, or in human-computer dialogue, a speaker often receives no feedback while speaking (or typing). This has the effect that, when a speaker has finished a turn, he does not know whether his contribution has been perceived, understood, and accepted. In a situation where "normal input-output" conditions hold (Searle, 1969), i.e. where participants speak the same language, have no hearing or speaking impairments, use communication channels without severe distortions, and so on, a speaker normally expects that the addressee perceives, understands and believes what is being said. We model this by the speaker having a doxastic attitude that we call *weak belief* that the addressee of his dialogue acts believes the preconditions and the content of the dialogue act to be true.[1] So after contributing an utterance that expresses a dialogue act with precondition c_1, the speaker A has the weak belief that B be-

[1] A weak belief is characteristically distinguished from a firm belief in that it is not inconsistent to weakly believe that p while at the same time having the goal to know whether p. In fact, the combination of such a goal and weak belief forms the preconditions of a CHECKQUESTION.

lieves that c_1. And similarly, in information-seeking dialogues, assistance dialogues, and other types of cooperative dialogue where the participants are expected to only provide correct information about the task at hand, if the utterance offers the information p about the task, then the speaker A also has the weak belief that B believes that c_1.

Of course, the assumptions of being understood and believed are not idiosyncratic for a particular speaker, but are commonly made by dialogue participants in cooperative dialogue in normal input-output conditions. B will therefore believe that A makes this assumption, so:

(6) B believes that A weakly believes that B believes that c_1. B believes that A weakly believes that B believes that p.

By the same token, A believes this to happen, hence:

(7) A believes that B believes that A weakly believes that B believes that c_1 and that p.

This line of reasoning can in principle be continued *ad inf nitum*, leading to the conclusion that:

(8) Both A and B believe that it is mutually believed that A weakly believes that B believes that c_1 and that p.

In the example dialogue, this means in particular that, after contributing utterance du_1, A will among other things believe the following 'weak mutual beliefs' to have been established, 'weak' in the sense that the mutual belief contains a weak belief link:

(9) a. A believes that it is mutually believed that A weakly believes that B believes that c_1.

b. A believes that it is mutually believed that A weakly believes that B believes that p.

The f rst of these weak mutual beliefs comes from the expected understanding of du_1, the second from the expected adoption of the information that du_1 offered.

More generally, what we see happening with respect to grounding, is that for an agent to ground a belief, what he has to do is not so much extend a f nite set of nested beliefs like (5) to an inf nite set of nested beliefs of any depth, but to replace the weak belief link in believed mutual beliefs of the form

(10) *A believes that it is mutually believed that A* **weakly** *believes that B believes that q*

by an ordinary belief link, turning it into

(11) *A believes that it is mutually believed that A believes that B believes that q*

which is equivalent[2] to:

(12) *A believes that it is mutually believed that q*

So the question is **what evidence is necessary and suff cient to strengthen the weakest link** in certain 'weak mutual beliefs'.

We have suggested above that the evidence behind nested beliefs of the complexity of (5) is *necessary* but not suff cient. That it is indeed necessary can be seen from the following example.

(13) 1. A: Where should I insert the paper?
2. B: In the paper feeder.
3. A: The paper to be faxed.
4. B: What did you say?

This example illustrates the above remark that utterances which provide feedback on a previous utterance are themselves also in need of feedback in order to make sure that they contribute to the grounding process. With utterance 3, A explains what he meant by *the paper* in his previous utterance, thereby indicating that he's not sure that his question was correctly understood. In other words, utterance 2 apparently did not provide A with positive feedback relating to being understood. A would certainly not be allowed to ground, having insuff cient evidence about the feedback that B has received up to this point in the dialogue. Hence at this point the process does not move into the direction of establishing a mutual belief about the preconditions of the question, let alone of the answer.

The issue of evidence being necessary and/or suff cient for strengthening the weakest link in a weak mutual belief is an empirical one. The case of (13) represents empirical evidence for the necessity of the evidence behind (5). Contrary to what we suggested above, empirical evidence in fact seems to

[2] This equivalence depends on the assumption that is known in epistemic logic as the Introspection axiom. According to this assumption, an agent believes his own beliefs, and in this case an agent also believes that he has a certain goal when he in fact has that goal. A precondition q of a dialogue act performed by a speaker A is always a property of A's state of beliefs and goals, hence *A believes that q* is equivalent to q. Moreover, all dialogue participants may be assumed to operate according to this assumption, hence *B believes that A believes that q* is equivalent to *B believes that q*.

show that the evidence of correct understanding that supports the beliefs represented in (5) is also *sufficient* for strengthening the weak mutual belief in (8). We express this observation as a pragmatic principle for the strengthening of the weakest link in a 'weak mutual belief'. The principle says that:

(14) a. A dialogue participant strengthens the weak belief link in a 'weak mutual mutual belief' concerning a precondition of a dialogue act that he has performed, when (1) he believes that the corresponding utterance was correctly understood; (2) he has evidence that: (2a) the other dialogue partner also believes that; and (2b) they both have evidence that they both have evidence that (1) and (2a) are the casse.

b. Like clause a., replacing "precondition of" by "task-related information, offered by", and replacing "correctly understood" by "believed".

We call (14) the *Strengthening Principle (SP)*. The SP may not seem very transparent at first; we will show its effect below, where we will see that it in fact comes down to a dialogue participant being able to ground preconditions or contents of a dialogue act when he has twice received positive feedback, namely positive feedback (possibly implicitly only) on the original utterance and positive feedback (again, possibly implicitly) on his response to that feedback act. In Morante (2007) and (Morante, forthcoming 2007) we provide ample empirical evidence for this principle, using corpora of both human-human and spoken human-computer dialogues; here we give just one example.

In dialogue (15), the SP predicts that B grounds the content of the first utterance when he successfully processes utterance 5 (second case of positive feedback). Indeed, it seems impossible for B to continue with utterance 6, expressing doubts about the grounded belief. By contrast, B could very well express such doubts in his previous turn, as (16) illustrates.

(15) 1. A: The next train is at 11:02.
2. B: At 11:02.
3. A: That's correct.
4. B: Okay thanks.
5. A: You're welcome.
6. B: *I thought it would be at 11:08.

(16) 1. A: The next train is at 11:02.
2. B: At 11:02.
3. A: That's correct.
4. B: I thought it would be at 11:08.

Since the only difference between (15) and (16) is the feedback that has been given by utterances 4 and 5, it must be the case that the evidence of correct understanding provided by these utterances makes the difference for grounding.

Limitations of space prevent us from going into the ways in which the various types of dialogue acts facilitate, speed up, or delay grounding in dialogue. See (Morante, forthcoming 2007) for a systematic discussion.

4 The DIT computational model of grounding

Our computational modeling of grounding, based on the strengthening of weak belief links in mutual beliefs, exploits the DIT structured context model and detailed analysis of feedback. The context model consists of several components, each representing a different type of information. The most relevant components to consider here are the *Linguistic Context*, the *Cognitive Context*, and the *Semantic Context*, which are defined as follows:

- Linguistic Context: a record of the dialogue up to this point, including verbatim representations of utterances as well as aspects of their syntactic, semantic and pragmatic analysis;
- Cognitive Context: information about the processing of utterances, notably about any problems in their interpretation or application;
- Semantic Context: information about the task, including nested beliefs about the dialogue partner's semantic context.

Evidence of correct understanding and of being believed, which triggers the application of the Strengthening Principle, is represented in the Cognitive Context. In order to see how the context updates, corresponding to understanding and believing each other, lead to the grounding of information, consider how the content of utterance 2 in the dialogue (17), *In the feeder*, is grounded.

(17) du_1. U: Where should I insert the paper?
du_2. S: In the feeder.
du_3. U: Should I put it in the bottom front tray?
du_4. S: No, in the open tray on top.
du_5. U: OK thanks .
du_6. S: You're welcome.
du_7. U: Goodbye.

We will represent the information that an utterance u was successfully processed (at all levels[3]) by agent Y as $Y^+(u)$, and the fact that agent X has evidence that agent Y succesfully processed that utterance as: $X : Y^+(u)$.[4]

Utterance du_3 in (17) shows a problem in understanding du_2 (represented by $U^-(du_2)$) in the form of a clarif cation question. As a result of recognizing this, S cancels the beliefs which ref ected his expectation that du_2 would be understood without problems (the beliefs labeled ssc4 and ssc5 in Table 1).

Utterance du_5 provides evidence for U's understanding the answer du_4 as well as believing it, so successful processing of du_5 introduces the element $S : U^+(du_4)$ into S's cognitive context. Utterance du_6 likewise can be taken to provide evidence that the preceding utterance was well understood, so that leads to U's cognitive context containing the element $U : S^+(du_5)$. And similarly du_7 leads to S's cognitive context containing $S : U^+(du_6)$.

Due to the local nature that feedback usually has, especially positive feedback (and even more strongly *implicit* positive feedback), this process however does not build up the nested evidence of understanding and believing du_2 that we need for its content to be grounded via the Strenghtening Principle. The key to solving this problem can be found in the observation that, *when you get positive feedback on your last contribution to the dialogue, then that is evidence for you that the speaker thinks that you successfully processed his preceding contribution.*

[3]DIT distinguishes several levels of feedback, namely those of paying attention, perception, understanding, evaluation, and application. The Feedback Chaining principle presented below is a simplif cation; in full it takes the various levels of feedback into account.

[4]Everywhere in this paper when we speak of 'feedback' we mean what in DIT is called *auto-feedback*, as opposed to *allo-feedback*. The former is concerned with information about the speaker's processing of dialogue utterances; the latter with the speaker's beliefs about the addressee's processing. For allo-feedback a similar chaining principle applies as the one described below for auto-feedback.

For example, when you have been asked a question, then positive feedback on the answer that you give constitutes evidence that you had understood the question well. We call this phenomenon **Feedback Chaining**. It can be represented formally as:

(18) $S^+(du_i) \Rightarrow S : A^+(du_{i-1})$

(with S indicating Speaker and A Addressee). Negative feedback is of course a different story: understanding of a negative feedback act means for the addressee that he has to address the utterance that caused the negative feedback. In the example of (17) we see that S recognizes that du_3 signaled a problem with du_2 (item $S^+(du_3^{-2})$ in S's Cognitive Context).

Note that Feedback Chaining is something that all participants in a dialogue do and assume all participants to do. Utterance du_5 in the example dialogue therefore not only leads to the element $S : U^+(du_4)$ in S's cognitive context, saying that S has evidence that U successfully processed utterance du_4, but from applying Feedback Chaining to the new element in his cognitive context also to inferring that U has evidence that S successfully processed the utterance preceding du_4, hence that $S : U : S^+(du_3)$.

Table 1 shows some of the information in the linguistic context of the participant who has the speaker turn, and of the effects of what is said on the participants' cognitive and semantic contexts. Of the linguistic context it shows: (1) the verbatim form of each turn; (2) the speaker of that turn: (3) the chronological location of the turn; (4) the communicative functions of the dialogue acts performed in that turn, where for simplicity we only show the communicative functions that are relevant to the present discussion.

Feedback Chaining has the effect that dialogue acts that provide feedback, either explicitly or implicitly, have a non-local effect and allow dialogue participants to build up evidence about each other's evidence concerning the processing of utterances earlier in the dialogue, and at some stage this nested evidence meets the requirements of the Strengthening Principle. In the example dialogue, U can ground the preconditions of his question du_3 after utterance du_6 since he has evidence that du_3 was well understood (element ucc3 of his cognitive context), and that S has evidence that this is the case (el-

Table 1: Linguistic, Cognitive and Semantic contexts (slightly simplif ed) for dialogue (17)

LC = Linguistic Context; CC = Cognitive Context; SC = Semantic Context. c_{ki} stands for the preconditions of du_k; c_k for the semantic content of du_k. 'und' = understanding of previous utterance; 'exp' = expected; 'ad' = adoption; FC = Feedback Chaining; SP = Strengthening Principle. 'bel' = belief; 'wbel' = weak belief; 'mbel' - mutual belief.

	num	*source*	*S's context*	*num*	*source*	*U's context*
SC				usc1	prec	c_{1i}
LC				du_1	U	*Where should I insert the paper?* WH–QUESTION
CC	scc1	und	$S^+(du_1)$			
SC	ssc1 ssc2 ssc3	und exp und prec	bel(S, c_{1i}) bel(S, mbel(S, U, wbel(U, bel(S, c_{1i})))) bel(S, c_2)	usc2	exp und	bel(U, mbel(S, U, wbel(U, bel(S, c_{1i}))))
LC	du_2	S	*In the feeder.* WH–ANSWER(du_1)			
CC				ucc1	und	$U^-(du_2)$
SC	ssc4 ssc5	exp und exp ad	bel(S, mbel(S, U, wbel(S, bel(U, c_{2i})))) bel(S, mbel(S, U, wbel(S, bel(U, c_2))))	usc2 usc3	exp und exp ad	bel(U, mbel(S, U, wbel(S, bel(U, c_{2i})))) bel(U, mbel(S, U, wbel(S, bel(U, c_2))))
LC				du_3	U	*Should I put it in the bottom front tray?* NEG. FEEDBACK YN-QUESTION(du_2)
CC	scc2 scc3	und FC	$S^+(du_3^{-2})$ $S : U^-(du_2)$			
SC	 ssc6 ssc7 ssc8	 und exp und prec	cancellation of ssc4, ssc5 bel(S, c_{3i}) bel(S, mbel(S, U, wbel(U, bel(S, c_{3i})))) bel(S, c_4)	usc4	exp und	bel(U, mbel(S, U, wbel(U, bel(S, c_{3i}))))
LC	du_4	S	*No, in the open tray on top.* YN–ANSWER(du_3)			
CC				ucc2 ucc3	und FC	$U^+(du_4)$ $U : S^+(du_3)$
SC	 ssc9 ssc10	 exp und exp ad	 bel(S, mbel(S, U, wbel(S, bel(U, c_{4i})))) bel(S, mbel(S, U, wbel(S, bel(U, c_4))))	usc5 usc6 usc7	ad exp und exp ad	cancellation of usc2, usc3 bel(U, c_4) bel(U, mbel(S, U, wbel(S, bel(U, c_{4i})))) bel(U, mbel(S, U, wbel(S, bel(U, c_4))))
LC				du_5	U	*OK thanks.* POSITIVE FEEDBACK(du_4)
CC	scc4 scc5 scc6	und FC FC	$S^+(du_5)$ $S : U^+(du_4)$ $S : U : S^+(du_3)$			
LC	du_6	S	*You'e welcome* POSITIVE FEEDBACK(du_5)			
CC				ucc4 ucc5 ucc6 ucc7	und FC FC FC	$U^+(du_6)$ $U : S^+(du_5)$ $U : S : U^+(du_4)$ $U : S : U : S^+(du_3)$
SC				usc6	SP	bel(S, mbel(S, U, c_{3i}))
LC				du_7	U	*Goodbye* POSITIVE FEEDBACK(du_6)
CC	scc7 scc8 scc9 scc10 scc11	und FC FC FC FC	$S^+(du_7)$ $S : U^+(du_6)$ $S : U : S^+(du_5)$ $S : U : S : U^+(du_4)$ $S : U : S : U : S^+(du_3)$			
SC	ssc7	SP	bel(S, mbel(S, U, c_4))			

ement ucc7).[5] This is what we may call the *grounding of the utterance* by U.

From an intuitive point of view, S should perhaps also be able to ground utterance du_4. But does he in fact have evidence that U correctly understood that utterance? All that S has to go by is U's thanking and goodbye acts, taken to also signal that U believes to have understood S's answer du_4 successfully, but of course U may be wrong; U's belief cannot constitute solid evidence for S. If indeed we want utterances to be grounded in such situations, then we need an additional pragmatic principle saying that, when a dialogue participant expresses that he has successfully processed a dialogue utterance, then this will be believed unless there is evidence to the contrary. Since utterance du_7 provides no such counter-evidence, S may at this point indeed assume that U processed du_4 successfully.

Note that our model of grounding says that the content of du_4 is *not* grounded for U at the end of this dialogue. Doesn't that make it unsatisfactory for U to end the dialogue? We believe not: we have here an information-seeking dialogue, with U as the information seeking participant. As far as U is concerned, the dialogue may end as soon as he believes that his question (du_3), replacing his original question du_1 was well understood and has received an answer (du_4) that he believes. What more could an information-seeking agent want?

5 Concluding Remarks

We have presented a simple, empirically based computational model of grounding in dialogue as the result of the strengthening of weak mutual beliefs. These weak beliefs are created through the assumptions that participants in dialogue make about the understanding and acceptance of what they say when normal input-ouput conditions hold. A crucial role in this model is played by the Strengthening Principle, which says that a dialogue participant can strengthen a weak mutual belief when he has sufficient evidence about both participants' belief that the utterance, which caused the weak belief was understood and accepted by the other participant.

A proof of concept implementation of the grounding model, outlined here, has been integrated as part of the Dialogue Manager module in a speech-based information-extraction system (see (Keizer and Morante, 2007)). This implementation proves the technical validity of the grounding model, and forms a platform for experimenting for example with different forms of the Strenghtening Principle for different types of dialogue.

References

J. F. Allen and C. R. Perrault. 1980. Analyzing intention in dialogues. *Artif cial Intelligence*, 15(3):143–178.

S. Brennan. 1998. The grounding problem in conversations with and through computers. In S.R. Fussell and R.J. Kreuz, editors, *Social and cognitive psychological approaches to interpersonal communication*, pages 201–225. Lawrence Erlbaum, Hillsdale, NJ.

H. Bunt. 2000. Dialogue pragmatics and context specif cation. In H. Bunt and W. Black, editors, *Abduction, Belief and Context in Dialogue*, pages 81–150. Benjamins, Amsterdam.

J. Cahn and S. Brennan. 1999. A psychological model of grounding and repair in dialog. In *Proc. AAAI FAll Symposium on Psychological Models of Communication in Collaborative Systems*, pages 25–33, North Falmouth, MA. AAAI.

H. Clark and E. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294.

S. Keizer and R. Morante. 2007. Dialogue simulation and context dynamics for dialogue management. In *Proc. NODALIDA Conference*, Tartu, Estonia.

S. Li, B. Wrede, and G. Sagerer. 2006. A computational model of multi–modal grounding for human robot interaction. In *Proc. 7th SIGdial Workshop on Discourse and Dialogue*, pages 153–160. ACL, Sydney.

C. Matheson, M. Poesio, and D. R. Traum. 2000. Modelling grounding and discourse obligations using update rules. In *Proceedings NAACL*.

R. Morante. 2007. Computing meaning in interaction. PhD Thesis, Tilburg University.

T. Paek and Eric Horvitz. 2000. Toward a formal theory of grounding. Technical report MSR–TR–2000–40, Microsoft Research, Redmond, WA.

J. Searle. 1969. *Speech acts*. Cambridge University Press, Cambridge, UK.

D. Traum. 1994. *A Computational Theory of Grounding in Natural Language Conversation*. PhD Thesis. Dep. of Computer Science, University of Rochester.

[5] The other SP conditions are also satisf ed since we assume that the attitude 'has evidence that', like the other doxastic attitudes that we use, is logically introspective (cf. footnote 2). Therefore $S^+(du_3) \Rightarrow S : S^+(du_3)$.

Pragmatic Usage of Linear Regression Models for the Prediction of User Judgments

Klaus-Peter Engelbrecht
Quality and Usability Lab
Deutsche Telekom Laboratories
Berlin University of Technology
Germany
Klaus-Peter.Engelbrecht@telekom.de

Sebastian Möller
Quality and Usability Lab
Deutsche Telekom Laboratories
Berlin University of Technology
Germany
Sebastian.Moeller@telekom.de

Abstract

Automatic evaluation of spoken dialog systems has gained interest among researchers in the past years. In the PARADISE framework (Walker et al. 1997), a linear regression function is trained on a dialog corpus to predict user ratings of satisfaction from interaction parameters. The accuracy of such predictions is generally measured with R^2, which usually is rather low. In this paper, it is shown that predictions according to PARADISE can lead to accurate test results despite the low R^2.

1 Introduction

Automatic usability testing of spoken dialog systems (SDSs) has gained interest among researchers in the past years. According to ISO 9241-11 (1998), usability of a system is compound of its effectiveness in doing typical tasks, the efficiency with which the task can be done and the satisfaction of the user with the system. Because user satisfaction is a subjective issue, usability testing involves humans who conduct typical tasks with the system and state their satisfaction with it afterwards. A key issue in automatic usability testing is the estimation of the expected user satisfaction without human involvement.

In the PARADISE framework (Walker et al. 1997) it is proposed to predict user satisfaction on the basis of interaction parameters captured in system log files. A linear regression (LR) model is trained on the parameters as predictors of user judgments of the corresponding dialogs as target. The percentage of the variance of the target that can be explained by the model is measured with R^2, which is based on the comparison of predicted and measured values for each dialog. When applying PARADISE, R^2 usually is below 0.6 for the prediction of the training data themselves (e.g. Walker et al. 2000), while the prediction of independent data is a stronger criterion and typically results in an even lower R^2.

Various steps have been taken to improve the predictive power of such equations. On the one hand, more and better predictor parameters have been searched for (Möller, 2005; Oulasvirta et al. 2006, Hastie et al. 2002), on the other hand, other prediction algorithms, e.g. classification and regression trees (CARTs) have been explored (Compagnoni 2006). However, R^2 values obtained remain unsatisfactory low.

While the standard accuracy measure for LR models, R^2, is based on a comparison of pairs of predicted and measured values for each dialog, in subjective measurement usually single ratings are not looked at. Instead, the researcher examines the overall distribution of all judgments for each questionnaire item. In fact, the very nature of subjective measurement involves joining ratings by multiple test subjects in order to minimize effects of inter-subject rating differences and by this maximizing the reproducibility of the findings. In other words, single ratings are tainted with different kinds of measurement errors (Annett, 2002). Consequently, an accurate prediction of single judgments is a Sisyphus task: it involves the difficult task to estimate the measurement errors, while at the same time the level of detail achieved by this is undesirable for the test result.

Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue, pages 291–294,
Antwerp, September 2007.

In the best case, the detail lost in LR predictions would be congruent with the detail deliberately eliminated during test evaluation. If this was true, the pragmatic value of PARADISE models would be higher than the R^2 values suggest. This paper discusses the application of PARADISE predictions in a pragmatic context, in order to estimate the severity of loss of detail in PARADISE predictions for their practical application.

Corpora of two different experiments serving the evaluation of spoken dialog systems have been analyzed with respect to how well test results can be reproduced by predictions with LR equations. In the following section, the databases used will be described. In Section 3, the application of the PARADISE approach to the data is explained, and in Section 4 examples are given to illustrate how prediction results can be used to reconstruct specific test results.

2 Data

Experiment 1 has been carried out during the EU-funded INSPIRE project (IST 2001-32746). The SDS tested in the experiment is capable of controlling domestic devices such as lamps and a video recorder, leading a mixed-initiative dialog with the user. For the experiment, the speech recognition (ASR) was replaced by a Wizard-of-Oz, transcribing the users' utterances. The aim of the experiment was to test the impact of ASR accuracy on user satisfaction by adding different degrees of word substitutions, deletions and insertions to the wizard's transcription. 28 users took part in the experiment. Test participants were required to carry out three scenarios, each with 9-11 tasks and covering all devices which can be operated with the system. This results in 84 dialogs in this database. Further details can be found in (Möller et al. 2007).

In experiment 2, the BoRIS restaurant information system (Möller 2005; see this also for a detailed description of the experiment) was tested, which allows the user to search for a restaurant in Bochum, Germany, by specifying constraints for type of food, restaurant location etc. In the experiment, ten system configurations have been compared which differed with respect to the prompt quality (TTS or recorded natural language), the confirmation strategy (explicit or implicit) and the ASR performance, modeled in a similar way as in experiment 1. Each of the 40 participants did five telephone calls to the system, following instructive scenarios. 197 dialogs are available in this database.

Both experiments were executed in test labs. From the system log files, a vast number of interaction parameters was computed, including efficiency measures (such as dialog duration), qualitative measures (such as contextual appropriateness) and a classification of user errors (Oulasvirta et al. 2006). A complete list of the qualitative and efficiency measures can be found in (Möller 2005).

After each interaction, the participants filled out a questionnaire designed according to ITU-T Rec. P.851 (2003). The first rating of the questionnaire is on the systems overall quality (OQ), which was collected on a continuous scale with five equidistant and labeled points. The scale margins were extended to encourage the use of the full scale.

3 Prediction of subjective ratings

LR models were calculated with the interaction parameters as predictor variables and OQ as target variable. From the equations found, predictions of the respective ratings were made and compared to the true ratings. Two methods have been applied for the prediction: in the `useall` method, the whole database is used for training and prediction, while in the `leave-one-out` (`l1o`) method, successively each user is predicted from the function trained on the other users. While the `useall` method indicates how well the data can be described with such a function, the `l1o` method gives a more reliable estimation of the predictive power of the model.

Exp.	R^2 `useall`	R^2 `l1o`
1	0,580	0,202
2	0,466	0,235

Table 1. R^2 for predictions of Overall Quality ratings in exp. 1 and 2.

In both cases, in opposition to what is foreseen in the PARADISE approach, the variables have not been z-transformed before the training. In PARADISE, standardization of predictors and targets allows to read the importance of the predictors for the prediction from the coefficients of the equation, which, however, is not relevant for this study. Instead, the mean and STD values should be preserved here to allow an estimation of how well they can be predicted with the function obtained from the training.

Table 1 shows the R^2 values of the prediction models for the two databases. While the values are generally low, R^2 is considerably lower for the `l1o` predictions than for the `useall` predictions. The numbers reported here for the `useall` method lie in the range of those observed by other researchers for other systems, while Walker et al. (2000) achieved better results for tests on independent test data than those reported here for the `l1o` method.

4 Predicting test results

As stated above, we suspected that the R^2 values are not a good indicator of the usefulness of the predictions in a practical context. We therefore applied the same type of analysis to the predictions as has been applied to the real data in the studies the data stem from. In the following, four examples of this are given.

Exp. 1 aimed at detecting the level of ASR performance necessary for system acceptance by simulating four different target word accuracy (WA) rates (60, 73, 86, 100%). The means for each configuration were plotted and connected by straight lines. Then, the threshold of the positive user judgment was located.

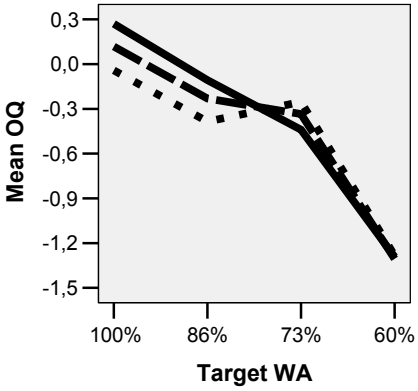


Figure 1. Overall Quality ratings for different WA in exp. 1; the solid line represents true ratings, the dashed line the `useall` predictions and the dotted one the `l1o` predictions.

Figure 1 shows how the results can be reproduced with data gained from predictions made with the LR equation. Displayed are the mean values of measured and predicted ratings for the four WA rates. While the predicted and measured means do not exactly agree with respect to the minimum WA leading to a positive judgment, the relation between WA and ratings is well reproduced by the prediction. The common conclusion that could be drawn from the predicted results as well as the true ratings would be that the WA should not fall below 73%, because from there on judgments decrease rapidly. Above 73%, the effect of WA is less drastic than below this value.

Similarly, results from experiment 2 can be predicted with the `l1o` and the `useall` method. In this experiment, again the users' judgment of the system for different target WA rates was tested. Figure 2 shows the means of measured and predicted values. Although the predicted values are slightly higher than the measured ones, the overall picture looks similar for the prediction and the actual measurement. The conclusion that can be drawn from both is the same: for recognition rates above 80 percent, an improvement of the target recognition rate is not reflected in improved ratings anymore, while for less than 80 percent, ratings drop to a lower range quickly.

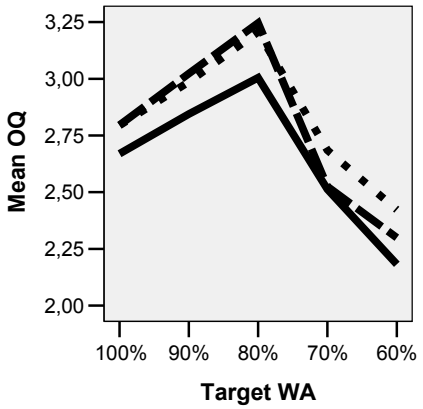


Figure 2. Overall Quality ratings for different target recognition rates in exp. 2; the solid line represents true ratings, the dashed line the `useall` predictions and the dotted one the `l1o` predictions.

In this experiment, also the impact of different system voices on the user judgment was tested. Figure 3 shows that both prediction methods reproduce the dramatic fall of ratings for the synthesized voice as compared to prerecorded human voices, however, the difference among the human speakers would not be detected with the prediction. Remarkably, `useall` and `l1o` method are comparably accurate despite the difference in their R^2*'s*.

Finally, the impact of the confirmation strategy on the judgments was tested with an ANOVA analysis (Table 2). While there is a bigger difference between the two confirmation strategies predicted with the LR equations than was actually

measured in the experiment, in all cases the difference is not significant (p>0.05, although F increases for the predicted values). Thus, the prediction leads to the same conclusion as the subjective ratings, namely that the confirmation strategy does not matter for the users' satisfaction with the BoRIS system.

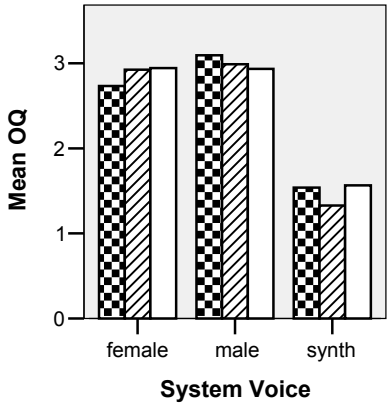


Figure 3. Overall Quality ratings for different Voices in exp. 2; the checkered bar represents true ratings, the shaded bar the `useall` predictions and the white one the `llo` predictions.

Confirmation Strategy (explicit/implicit)			
	measured	**`Useall`**	**`llo`**
F	(1,193) 0.02	(1,195) 2.88	(1,195) 3.10
p	0.89	0.09	0.08

Table 2. ANOVA results for explicit and implicit confirmation strategies. Differences in ratings for both strategies are not significant (p>0.05), neither in the measurement nor in the predictions.

5 Conclusion

In this paper, it was proposed to compare the mean values of predicted and true ratings rather than values for single dialogs. It was shown how LR models can be utilized for the automatic prediction of experimental results based on the observation of mean values. Although the predictions still lack some accuracy, the prediction models are more valuable in practical applications than their R^2 values suggest. In particular, the prediction of unseen data does not cause a dramatic drop of the model accuracy, as was indicated by the R^2 values. This is a particularly valuable finding since most applications intended for PARADISE involve the prediction of unseen data.

A further implication of the findings is that the improvement of usability prediction models on the basis of LR should not be based on changes in R^2 alone. While better methods for the evaluation of the models still have to be found, they might lead to significant progress in the models' development. This includes selection of appropriate modeling techniques (CARTs, Neural Networks etc.) and training methods for the algorithm, as well as the estimation of the usefulness of interaction or system parameters as usability predictors.

References

John Annett. 2002. Subjective Rating Scales. Science or Art? *Ergonomics*, 45(14):966-987

Bernardo Compagnoni. 2006. *Development of Prediction Models for the Quality of Spoken Dialog Systems*. Diploma thesis, Deutsche Telekom Laboratories/Institute for Telecommunications Technology, TU Braunschweig, Germany.

International Organization for Standardization. 1998. *ISO 9241-11, Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability*, Geneva, Switzerland.

Sebastian Möller, Paula Smeele, H. Boland, and Jan Krebber. 2007. Evaluating Spoken Dialog Systems According to De-facto Standards: A Case Study. *Computer Speech and Language* 21:26-53.

Sebastian Möller. 2005. *Quality of Telephone-based Spoken Dialog Systems*, Springer Science+Business Media, Inc., New York, NY.

Antti Oulasvirta, Klaus-Peter Engelbrecht, Anthony Jameson, and Sebastian Möller. 2006. The Relationship Between User Errors and Perceived Usability of a Spoken Dialog System. *ISCA/DEGA Tutorial and Research Workshop on Perceptual Quality of Systems*, Berlin, Germany:61-67.

Marilyn Walker, Candace Kamm, and Diane Litman. 2000. Towards developing general models of usability. *Natural Language Engineering* 6(3-4):363-377.

Marilyn Walker, Diane Litman, Candace Kamm, and Alicia Abella. 1997. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. *Proc. of the ACL/EACL 35th Ann. Meeting of the Assoc. for Computational Linguistics*, Madrid, Spain:271–280.

Helen Wright Hastie, Rashmi Prasad, and Marilyn Walker. 2002. Automatic Evaluation: Using a Dialogue Act Tagger for User Satisfaction and Task Completion Prediction. *Proc. 3rd Int. Conf. on Language Resources and Evaluation (LREC)*, 2:641–648, Las Palmas, Spain.