Decalog 2007

Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue (SemDial 11)

May 30 – June 1, 2007
Rovereto, Italy

edited by
Ron Artstein
Laure Vieu

Preface

We are happy to present Decalog 2007, the 11th workshop on the Semantics and Pragmatics of Dialogue, ten years after the inception of the series in 1997. This year's workshop continues the tradition of presenting high-quality talks and posters on dialogue from a variety of perspectives such as formal semantics and pragmatics, artifical intelligence, computational linguistics, and psycholinguistics. The appeal of the SemDial series is growing – this year we have seen interest from researchers in fields as diverse as language pedagogy, field linguistics, and sociology (unfortunately these people did not submit papers, mostly for technical reasons).

We received 32 submissions to the main session, and each was reviewed by two or three experts. We selected 19 talks for oral presentation (of which 17 will be presented); the poster session hosts many of the remaining submissions, together with additional submissions that came in response to a call for late-breaking posters and demos.

We are lucky to have four first-rate researchers on discourse and dialogue as invited speakers – Bruno G. Bara, Renato De Mori, Paul Piwek and Ipke Wachsmuth. They represent a broad range of perspectives and disciplines, and together with the accepted talks and posters we hope to have a productive and lively workshop.

We are grateful to our reviewers, who invested a lot of time giving very useful feedback, both to the program chairs and to the authors: Jan Alexandersson, Maria Aloni, Nicholas Asher, Anton Benz, Raffaella Bernardi, Patrick Blackburn, Johan Bos, Monica Bucciarelli, Craig Chambers, Marco Colombetti, Paul Dekker, Raquel Fernández, Ruth Filik, Simon Garrod, Jonathan Ginzburg, Joris Hulstijn, Elsi Kaiser, Alistair Knott, Staffan Larsson, Alex Lascarides, Colin Matheson, Nicolas Maudet, Philippe Muller, Fabio Pianesi, Martin Pickering, Manfred Pinkal, Matthew Purver, Hannes Rieser, Laurent Roussarie, David Schlangen, Amanda Stent, Matthew Stone, Enric Vallduvi, and Henk Zeevat.

This workshop would not have been possible without the generous support of CIMeC – the Center For Mind/Brain Sciences at the University of Trento; LUNA – the EU-funded project on Spoken Language Understanding in Multilingual Communication Systems; and ILIKS – the Interdisciplinary Laboratory on Interacting Knowledge Systems.

Many thanks to the local organization team in Rovereto, headed by Massimo Poesio and Alessia La Micela, who have invested an enormous amount of work and preparations to have the workshop run smoothly.

Ron Artstein and Laure Vieu

Colchester and Trento, May 2007

Contents

Wednesday talks

Thursday talks

Friday talks

Posters and demos

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, page 7.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Neuropragmatics: Mind/brain evidence for communicative intentions.

Bruno G. Bara
Center for Cognitive Science
University and Polytechnic of Turin
bruno.bara@psych.unito.it

Keyword: communication; pragmatics; intention; social brain.

Human beings are genetically designed in order to maximize their capacity for social interaction. At birth they already possess complex primitives (like sharedness) which allow them to master communication far beyond other animals' ability.
The most important primitive for communication is *communicative intention*, which may be formally defined (Bara, 2007) as follows:

$$\mathrm{CINT}_{A,B}\, p = \mathrm{INT}_A\ \mathrm{Shared}_{B,A}\ (\mathrm{p} \wedge \mathrm{CINT}_{A,B}\, p)$$

A has the communicative intention that *p* towards B ($\mathrm{CINT}_{AB}\ p$) when A intends (INT_A) that the following two facts be shared by B and herself (Shared_{BA}): that *p*, and that she intended to communicate to B that *p* ($\mathrm{CINT}_{AB}\ p$).
The developmental evidence of communicative intention as primitive is that 9-months-old children perform communication acts like declarative pointing. I.e., they are able to express the intention of sharing an action/object between the self and the other (Tomasello *et al*., 2005).
The neuroimaging evidence consists in a series of fMRI experiments, where we demonstrated that the anterior paracingulate cortex is not necessarily involved in the understanding of other people's intentions per se, but primarily in the understanding of the intentions of people involved in social interaction (Walter *et al.*, 2004). Moreover, this brain region showed activation when a represented intention implies social interaction and therefore had not yet actually occurred. This result suggests that the anterior paracingulate cortex is also involved in our ability to predict future intentional social interaction, based on an isolated agent's behaviour. We conclude that distinct areas of the neural system underlying theory of mind are specialized in processing distinct classes of intentions (Ciaramidaro *et al.*, 2007), among which there is communicative intention with its distinctive features.

References

Bara B.G. (2007). *Cognitive Pragmatics: Mental processes of communication*. MIT Press, Boston, MA.

Ciaramidaro A., Adenzato M., Enrici I., Erk S., Pia L., Bara B.G., Walter H. (2007). The intentional network: How the brain reads varieties of intentions. *Neuropsychologia*.

Tomasello M., Carpenter M., Call J., Behne T., Moll H. (2005). Understanding and sharing intentions: the origins of cultural cognition. *Behavioral and Brain Sciences*, 28:5, 675-691

Walter H., Adenzato M., Ciaramidaro A., Enrici I., Pia L., Bara B. G. (2004). Understanding intentions in social interaction: the role of anterior paracingulate cortex. *Journal of Cognitive Neuroscience*, 16:10, 1854-1863.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 9–16.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

A dialogue act based model for context updating

Roser Morante Simon Keizer Harry bunt
Department of Communication and Information Sciences
Faculty of Humanities
Tilburg University, The Netherlands
{R.Morante,S.Keizer,H.Bunt}@uvt.nl

Abstract

In this paper we describe a context update model that has been implemented in a dialogue manager. The model is based on the assumptions that utterances in a dialogue can be represented in terms of dialogue acts, and that they provoke several types of effects in the dialogue participant's belief state. In the paper, a step-by-step analysis of the context update during a dialogue will be provided, focusing on the belief states of the dialogue participants.

1 Introduction

In this paper we describe a context update model that has been implemented in a dialogue manager that operates within an interactive question answering system (Keizer and Bunt, 2006), making it possible to develop complex dialogue act generation mechanisms that employ the rich information provided by the beliefs in the context model.

The context update algorithm is built on Dynamic Interpretation Theory (DIT), (Bunt, 2000), in which dialogue utterances are interpreted as having intended context–changing effects that are determined by the dialogue act(s) being performed with the utterance. So, generally speaking, we follow the Information State Update approach in dialogue modelling (Traum and Larsson, 2003), with a strong emphasis on dialogue acts and a complex context model.

The context update is based on the specification of the preconditions of the dialogue acts in the DIT taxonomy, which describe the motivation and assumptions of an agent to perform the dialogue act, and on the representation of several types of effects that utterances have in the belief state of dialogue participants.

This paper is organised as follows. Section 2 presents the theoretical background. In Section 3 we describe the update model, which is then applied to an example dialogue in Section 4. A step-by-step analysis of the context update during a dialogue is provided, showing how the belief states of the dialogue agents evolve, provoking changes in the context model that have a role in the generation of utterances. Section 5 ends the paper with discussion and conclusions.

2 Theoretical background

In Dynamic Interpretation Theory (DIT) (Bunt, 2000), a dialogue is modelled as a sequence of utterances expressing sets of *dialogue acts*. These are semantic units, operating on the information states of the participants. Formally, a dialogue act in DIT consists of a *semantic content* and a *communicative function*, the latter specifying how the information state of the addressee is to be updated with the former upon understanding the corresponding utterance. Communicative functions are organised in a taxonomy[1] consisting of ten *dimensions* (Bunt, 2006): Task-Oriented acts, Auto-Feedback, Allo-Feedback, six dimensions of Interaction Management (IM), such as turn- and time-management, and Social Obligations Management (SOM). Several dialogue acts can be performed in each utterance, at most one from each dimension. Dimensions of communication are different aspects of the communication process that can be addressed independently and simultaneously by means of dialogue acts.

[1] See web page http://ls0143.uvt.nl/dit/.

$$\left[\begin{array}{ll}
\mathit{LingContext} : & \left[\begin{array}{l}
\mathit{user\_utts} : \langle \mathit{last\_user\_dial\_act} = \mathit{uda_0}, \mathit{uda_{-1}}, \mathit{uda_{-2}}, \ldots \rangle \\
\mathit{system\_utts} : \langle \mathit{last\_system\_dial\_act} = \mathit{sda_0}, \mathit{sda_{-1}}, \mathit{sda_{-2}}, \ldots \rangle \\
\mathit{topic\_struct} : \langle \mathit{referents} \rangle \\
\mathit{conv\_state} : \mathit{opening} | \mathit{body} | \mathit{closing} \\
\mathit{candidate\_dial\_acts} : \ldots \\
\mathit{dial\_acts\_pres} : \ldots
\end{array}\right] \\
\mathit{SemContext} : & \left[\begin{array}{l}
\mathit{task\_progress} : \mathit{comp\_quest} | \mathit{quest\_qa} | \mathit{answ\_eval} | \mathit{user\_sat} \\
\mathit{user\_model} : \langle \mathit{beliefs} \rangle
\end{array}\right] \\
\mathit{CogContext} : & \left[\begin{array}{l}
\mathit{own\_proc\_state} : \left[\begin{array}{l}
\mathit{proc\_problem} : \mathit{perc} | \mathit{int} | \mathit{eval} | \mathit{exec} | \mathit{none} \\
\mathit{user\_model} : \langle \mathit{beliefs} \rangle
\end{array}\right] \\
\mathit{partner\_proc\_state} : \left[\begin{array}{l}
\mathit{proc\_problem} : \mathit{perc} | \mathit{int} | \mathit{eval} | \mathit{exec} | \mathit{none} \\
\mathit{user\_model} : \langle \mathit{beliefs} \rangle
\end{array}\right] \\
\mathit{belief\_model} : \langle \mathit{beliefs} \rangle \\
\mathit{common\_ground} : \langle \mathit{mutual\_beliefs} \rangle
\end{array}\right] \\
\mathit{SocContext} : & \left[\begin{array}{l}
\mathit{comm\_pressure} : \mathit{none} | \mathit{grt} | \mathit{apo} | \mathit{thk} | \mathit{valed}
\end{array}\right]
\end{array}\right]$$

Figure 1: Feature structure representation of the context model used.

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 in the dialogue so far (a kind of 'extended dialogue history'); information about planned system dialogue acts (a '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 Section 3.3);

4. *Physical and Perceptual Context*: the perceptible aspects of the communication process and the task/domain;

5. *Social Context*: current communicative pressures.

In Figure 1, a feature structure representation is given of our context model. The context model is extensively described in (Keizer and Morante, 2007). Currently, information about the physical and perceptual context is not considered relevant for the types of dialogue and underlying tasks that we will consider in Section 4.

In updating the context model on the basis of dialogue acts, their preconditions form the basis for changing the system's belief model. There is a correspondence between the dimension of a dialogue act and the components of the context model it particularly operates on. For example, dialogue acts in the task/domain dimension typically provoke changes in the *Semantic Context* and SOM acts typically create or release communicative pressures as recorded in the *Social Context*. The meta-information for user utterances typically results in the recording of processing problems in the own processing state of the *Cognitive Context*. Feedback acts also provoke changes in the *Cognitive Context*, but may cause beliefs in any part of the context model to be cancelled. The system providing domain information to the user will result in a belief in the *Semantic Context* about the user now having this information, but that belief will have to be cancelled when the user then produces a negative auto-feedback act, indicating he did not hear or understand the system's utterance.

In the next section we describe the part of the update model related to updating the beliefs of dialogue participants.

3 The context update model

Regarding the context update, DIT follows the same basic idea as the information state update approach (Traum and Larsson, 2003): the context model is updated during a dialogue under the influence of the participants' utterances, depending particularly on the *dialogue acts* performed. The context update starts from an abstract representation of the utterances in terms of dialogue acts. Dialogue acts have preconditions, which represent the motivation and assumptions required for the agent to perform a dialogue act. This approach is similar to the BDI paradigm (Allen and Perrault, 1980).

In order to explain the epistemic aspects of the context update, DIT defines mechanisms for context update, as well as several types of effects that utterances provoke in the context model. This section is devoted to present both the mechanisms and the types of effects, whereas the next section will present the analysis of a dialogue.

3.1 Mechanisms for context update

The four mechanisms for context update are creation, adoption, strengthening, and cancellation of beliefs.

Creation: Belief creation is the effect of assigning an interpretation to what has been said. When an utterance is understood by an addressee A as a dialogue act of a certain type, then if *c* is a precondition of that dialogue act, A will believe that *c* holds unless *c* contradicts with other beliefs that A entertains. If *b* is a belief of S resulting from processing a previous utterance, A will believe that *b*, unless *b* contradicts with other beliefs of A.

Adoption: The adoption mechanism specifies when a dialogue participant incorporates beliefs or goals of other dialogue participants as beliefs or goals of his own. For example, when an utterance is understood by an addressee A as an information–providing dialogue act, making the information I available to A, then if A does not hold beliefs that contradict I, A adopts this information as a belief of his own. This rule is reminiscent of the *Belief Transfer Rule* defined by (Perrault, 1990), who states the effects of speech acts in terms of Default Logic. The *Belief Transfer* rule says that if one agent believes that another agent believes something the first agent will come to believe it too, unless he has evidence to the contrary.

Strengthening: Strengthening a belief means converting it from a weak belief into a strong belief. A speaker's weak beliefs, expressing his expectations concerning the understanding and acceptance of an utterance that he has contributed are strengthened to become strong beliefs when the addressee provides explicit or implicit positive feedback about his processing of the utterance. A participant's believed mutual beliefs about a weak belief, are strengthened to become believed mutual beliefs about a strong belief when (1) he believes that both partners believe that the utterance was well understood by the addressee and accepted without evaluation problems; (2) he has evidence that both dialogue partners have evidence that they both have evidence that (1) is the case. An extended explanation about how strengthening applies can be found in (Bunt and Morante, 2007; Morante, 2007).

In short, from the moment that a dialogue participant creates a mutual belief about a weak belief, two non-problematic turns by the other dialogue participant are necessary. This is due to the fact that certain beliefs have to be in the context model before strengthening can take place. For example, if participant A has a mutual belief about a weak belief that p as a result of his interaction with participant B, the following beliefs have to be in the context for the strengthening of the weak belief to take place:

(1) (i) A believes that p
(ii) A believes that B believes that p
(iii) A believes that B believes that A believes that p

In the model, the creation of a mutual belief about a strengthened belief indicates that the information in the strengthened belief is grounded by the holder of the mutual belief.

Cancellation: Cancellation of a belief or goal means removing it from the context model. A goal is cancelled when it has been satisfied or proved to be unsatisfiable.

3.2 Effects of utterances in the context model

The types of effects that utterances provoke in the context model are related to understanding and adopting information.

Understanding effects: If the Addressee understands the Speaker's utterance, beliefs will be created in the Addressee's context model about the fact that he believes that the preconditions of the Speaker's utterance hold. Additionally, if the Speaker's utterance provides implicit positive feedback, beliefs will be created in the Addressee's context model about the Speaker having understood the previous Addressee's utterance(s).

Expected understanding effects: The Speaker will expect that, unless there are reasons to think the contrary (like interferences in the communication), the Addressee understands correctly what the Speaker said, and that the Addressee understands the implicit positive feedback effects of the current utterance with respect to previous utterances. The Speaker cannot be certain about this, however, as long as he does not receive any feedback from the Addressee. This is why these beliefs are modelled as 'weak beliefs': the Speaker weakly believes that the Addressee understood the utterance and that the Addressee understood the implicit positive feedback effects of the utterance.

The Speaker will believe that the Addressee will also believe that the Speaker weakly believes that the Addressee understood the Speaker's utterance and its implicit positive feedback effects. More in general, the idea that speakers expect to be correctly understood is assumed to be shared by all speakers and addressees. That is, both Speaker and Addressee believe that it is *mutually believed* that the Speaker weakly believes that the Addressee understood the Speaker's utterance and its implicit positive feedback effects.

Mutual beliefs about weak beliefs can be converted into mutual beliefs about strong beliefs by applying the *strengthening* mechanism.

Adoption effects: If the Addressee correctly understands the Speaker's utterance, and if the Speaker's utterance contains information that the Addressee considers as trustworthy, then the Addressee will adopt this information.

Expected adoption effects: These are of the same type as the expected understanding effects, with the difference that they apply to effects of adoption instead of effects of understanding. For example, if as a result of an adoption effect Addressee *B believes that p*, the mutual beliefs about expectations of adoption on the side of Speaker A is *A believes that it is mutually believed that A weakly believes that B believes that p*. On the side of Addressee B, if processing has been correct, the same mutual belief arises: *B believes that it is mutually believed that A weakly believes that B believes that p*.

3.3 The role of feedback

DIT establishes four levels of feedback, that reflect how an utterance has been understood and how the speaker is able to react to that utterance depending on his current state of information, either positively or negatively. Negative feedback on one level implies positive feedback of the previous levels.

- *Perception* In terms of utterance processing in a dialogue system, this level is associated with successful speech recognition.
- *Interpretation* corresponds to being able to recognise the dialogue act(s) performed with the utterance.
- *Evaluation* indicates that the beliefs that result from the (preconditions of the) dialogue act identified at the interpretation level are consistent with the own information state.
- *Execution* level means being able to do something with the result of the evaluation. For example, in the case of a question, it consists of finding the information asked for; in the case of an answer, it means taking over the content of the answer.

Feedback acts express information about the feedback level reached and have consequences in the context update process. Negative auto-feedback acts have as a consequence the cancellation of beliefs. An utterance U by participant A addressed to participant B in relation to B's previous utterance U_{-1} that expresses negative perception or understanding has the effect of cancelling B's beliefs created as a consequence of U_{-1}. If participant A signals that he did not perceive or understand what participant B said, it means that participant A can not have any beliefs about what B said. Consequently, B has to cancel the effects of expectations of understanding, and, if it applies, also the expectations of adoption. If U expresses negative evaluation or negative execution, then it has the effect of cancelling B's beliefs about effects of expected adoption created as a consequence of U_{-1}.

The effects of positive auto-feedback acts on the belief model have as a consequence that the creation of beliefs as a result of the different types of effects proceeds as expected, and in after the necessary turns have occurred it will lead to participants creating a common ground.

4 Example dialogue

In this section, we will show how the context update model works in the case of the dialogue in (2), in which the User (U) asks the System (S) for information about how to operate a fax machine.

(2) (U1) **User**: Where should I put the paper that has to be copied?

(S2) **System**: In the feeder.

(U3) **User**: Thank you.

(S4) **System**: Sorry?

(U5) **User**: Thank you.

(S6) **System**: You're welcome.

In U1, the User puts a WH–QUESTION to the System. Questions in general have two preconditions: the speaker wants to know something and the speaker believes that the hearer has that information. In this case the User wants to know where the paper to be copied has to be put [u01][2] and the User believes that the System knows where the paper to be copied has to be put [u02].

After U1, the User expects that the System has understood his utterance, which is modelled as a weak belief that the System believes that the preconditions of the WH–QUESTION hold for the User. Recall that this belief is weak, because the User did not yet receive any positive feedback from the System. The effects of expected understanding also stated that this expectation is believed to be mutually believed by both User and System. All of this results in the creation of beliefs [u1,u2] in the User's context model and beliefs [s1,s2] in the System's context model.

Besides the effects of expected understanding as indicated above, also effects of understanding the WH–QUESTION apply in updating the System's context model. This results in the System believing that the User wants to know something and that the User believes that the System knows it [s1,s2].

Because the dialogue act belongs to the task-domain dimension, the beliefs resulting from utterance U1 are are recorded in the System's *Semantic Context*. Belief [s1] involves a user goal and forms a trigger for the System to generate a task-domain dialogue act in order to satisfy this goal. In the example, the System has been able to find the information it believes the user wants, and produces S2.

S2 is a WH–ANSWER by the System that answers the question put in U1, and at the same time gives implicit positive feedback to the User: the User may now conclude that the System understood U1 because the System has given a relevant reply. This understanding effect results in the beliefs [u3, u4]. The effect of the User understanding the System's answer results in [u5], i.e., the User believes that the System believes that the paper should be inserted in the feeder. Additionally, having successfully evaluated the answer and assuming the System is cooperative and a domain expert, the User adopts the information given by the System and now himself believes that the paper has to be put in the feeder [u6]. Now having the information he asked for in U1, the User can cancel the corresponding goal [u01].

In addition to these understanding effects, there are effects of expected understanding, i.e., both System and User believe that it is mutually believed that the System expects his utterance S2 to be understood. Again, this expectation is modelled via a weak belief, in this case, a weak belief by the System. Understanding S2 here includes both understanding the answer as such and the effects of implied positive feedback. Hence, in the User's context model we get beliefs [u7] to [u9] and in the System's context model [s5] to [s7]. On top of that, in the context models of both System and User, effects of expected adoption apply, i.e., they now both believe that it is mutually believed that the User will adopt the information provided by the System in S2 ([u10] and [s8]).

In U3 the User thanks the System with a THANKING function from the SOM dimension. He updates his context model with expected understanding effects, i.e., he expects that the System will interpret U3 as providing implied positive feedback about S2. This means that the User expects the System to believe that the User fully understood S2, as expressed in [u3] to [u6]. Together with the assumption that this is mutually believed, this results in beliefs [u11] to [u14] in the User's context model.

However, the System could not successfully process U3. Updating the context model with this event

[2] The numbers between brackets refer to belief numbers used in Table 1 below.

results in recording a perception level processing problem in the *Cognitive Context.* In the absence of an interpretation of U3 in terms of dialogue acts, no beliefs are created in the context of the System. The perception problem is the motivation for the System to produce a NEGATIVE AUTO-FEEDBACK PERCEPTION dialogue act in S4. After successful interpretation of S4 as this negative feedback act, the User has to cancel his beliefs about expecting the System to understand U3: [u11] to [u14]. As a consequence of S4 the User repeats U3 in U5.

U5 has the same effects in the User's context as U3: [u11] to [u14]. Additionally, it has effects in the System's context model because now the System correctly understood. The THANKING function has the effect of creating a so-called reactive pressure in the Social Context, which will be released in utterance S6. A THANKING function has also effects of implicit positive feedback, resulting in the System believing that the User fully understood S2, as expressed in beliefs [u3] to [u6]. Hence, the System creates beliefs [s9] to [s12] in his context model. Because the System now successfully processed U5, he also creates beliefs about the User expecting the System to fully understand U5: [s13] to [s16].

In S6 the System responds to the User's thanks with a THANKING-DOWNPLAY function, releasing the reactive pressure created after U5. The dialogue act also implies positive feedback, causing the corresponding effects of understanding in the User's context model: the User now believes that the System understood U5, as expressed by beliefs [s9] to [s10], resulting in the User's beliefs [u15] to [u18].

There are also effects of expected understanding in both participants' context models. Both User and System believe that it is mutually believed that the System expects S6 to be understood by the User, including the implied positive feedback provided. This leads to beliefs [s17] to [s20] in the System's context model, and [u19] to [u22] in the User's context model.

On top of that, this utterance has the effect of creating two more beliefs in the User's context model, [u23] and [u24], which are the result of strengthening beliefs [u1] and [u2]. So now the User now believes that it is mutually believed that the User (strongly, instead of weakly) believes that the System understood the initial User's question U1. The strengthening is justified by the presence of the beliefs [u3], [u4], [u15], and [u16].

Let us analyse the strengthening of belief [u1], expressing that "the User believes that it is mutually believed that the User weakly believes that the System believes that the User wants to know where he should put the paper to be copied". In [u1], the weak belief is about a System's belief, so the User cannot convert this into a strong belief until the System gives some positive feedback, implicit or explicit. This happens in S2, where the System replies with a WH–ANSWER that is relevant for the User's question. This is why the User creates then belief [u3], "the User believes that the System believes that the User wants to know where he should put the paper to be copied", which corresponds to (i) in the required beliefs for strengthening listed in (1).

Because in this case [s1] is a belief by the System ("System believes that the User wants to know where he should put the paper to be copied"), (i) and (ii) in (1) are expressed by the same belief, [u3]. Case (ii) should be: "the User believes that the System believes that System believes that the User wants to know where he should put the paper to be copied". We consider "the System believes that System believes" to be equivalent to "the System believes". Case (iii) in (1) is belief [u15], created after S6: "the User believes that the System believes that the User believes that the System believes that the User wants to know where he should put the paper to be copied".

5 Conclusions

We have presented a model for context updating in dialogue. The model provides an exact specification of how the participants' belief states evolve during a dialogue. The utterances produced are specified in terms of dialogue acts and have several types of effects on the belief states.

The context update model has been implemented in a dialogue manager that operates within an interactive question answering system. The input to the context update algorithm (Keizer and Morante, 2006) is an abstract representation of a system or user utterance. In the case of a user utterance, this representation is the result of the output produced by the various language analysis components. This consists of meta-information in the form of an understanding level reached. If there was successful dialogue act recognition, i.e., at least interpretation level understanding was reached, the representation will also contain a set of dialogue acts. In the case of a system utterance, the underlying dialogue acts are generated by the system himself, and therefore, the abstract representation will only consist of these dialogue acts.

The rich information in the context model allows us to experiment with dialogue act generation mechanisms for dialogues that are more complex both in the sense of flexible task execution and dealing with communication problems. For example, the common ground information in the System's context can be taken into account in order to decide if information has to be presented to the user as new or as known. Besides dialogue act generation, another interesting topic for future work is making the dialogue manager more powerful by enabling it to reason about the beliefs in the context model.

Acknowledgements

This research is partly funded through the PARADIME project, which falls under the IMIX research programme, funded by the Dutch national research foundation NWO. Roser Morante's research is funded by the EX2005–1145 grant awarded by the Spanish *Ministerio de Educación y Ciencia*. We thank the anonymous reviewers for their useful comments.

References

J. F. Allen and C. R. Perrault. 1980. Analyzing intention in dialogues. *Artificial Intelligence*, 15(3):143–178.

H. Bunt and R. Morante. 2007. The weakest link. Submitted manuscript.

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.

H. Bunt. 2006. Dimensions in dialogue act annotation. In *Proceedings 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1444–1449, Genova, Italy.

S. Keizer and H. Bunt. 2006. Multidimensional dialogue management. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, pages 37–45, Sydney, Australia.

S. Keizer and R. Morante. 2006. Context specification and update mechanisms for dialogue management. In *Proceedings of the 18th BeNeLux Conference on Artificial Intelligence BNAIC'06*, pages 181–188, Namur, Belgium.

S. Keizer and R. Morante. 2007. Dialogue simulation and context dynamics for dialogue management. In *Proceedings of the NODALIDA conference*, Tartu, Estonia.

R. Morante. 2007. Computing meaning in interaction. PhD Thesis. Forthcoming.

C. R. Perrault. 1990. An application of default logic to speech act theory. In P.R. Cohen, J. Morgan, and M.E. Pollack, editors, *Intentions in Communication*. MIT Press, Cambridge, MA.

D. R. Traum and S. Larsson. 2003. The information state approach to dialogue management. In Jan van Kuppevelt and Ronnie Smith, editors, *Current and New Directions in Discourse and Dialogue*, pages 325–354. Kluwer, Dordrecht.

Table 1: Analysis of the dialogue in Example 2.

Beliefs are numbered in columns 1 and 4 (**num**); the type of belief is indicated in columns 2 and 5 (**type**): precondition (prec); understanding effects (und); adoption effects (ad); expected understanding (exp.und); expected adoption (exp.ad). Operations on beliefs are indicated by *operation:number*, where *ad*, *ca* and *st* stand for adoption, cancellation and strengthening. Columns 3 and 6 contain the System's and User's beliefs. 'BEL_MBEL' stands for 'believes that it is mutually believed', 'WBEL' stands for 'weakly believes', and 'BEL' stand for 'believes'.

num	**type**	**beliefs System**	**num**	**type**	**beliefs User**
			u01	prec	WANT(U,KNOW(U,LOCATION_OF_PAPER))
			u02	prec	BEL(U,KNOW(S,LOCATION_OF_PAPER))
(U1) User: Where should I put the paper that has to be copied?					
s1	und.	BEL(S,u01)			
s2	und.	BEL(S,u02)			
s3	exp.und.	BEL_MBEL(S,WBEL(U,s1))	u1	exp.und.	BEL_MBEL(U,WBEL(U,s1))
s4	exp.und.	BEL_MBEL(S,WBEL(U,s2))	u2	exp.und.	BEL_MBEL(U,WBEL(U,s2))
s01	prec	BEL(S,LOCATION_OF_PAPER IS FEEDER)			
(S2) System: In the feeder.					
			u3	und.	BEL(U,s1)
			u4	und.	BEL(U,s2)
			u5	und.	BEL(U,s01)
			u6	ad:u5	BEL(U,LOCATION_OF_PAPER IS FEEDER)
s5	exp.und	BEL_MBEL(S,WBEL(S,u3))	u7	exp.und.	BEL_MBEL(U,WBEL(S,u3))
s6	exp.und	BEL_MBEL(S,WBEL(S,u4))	u8	exp.und.	BEL_MBEL(U,WBEL(S,u4))
s7	exp.und	BEL_MBEL(S,WBEL(S,u5))	u9	exp.und.	BEL_MBEL(U,WBEL(S,u5))
s8	exp.ad	BEL_MBEL(S,WBEL(S,u6))	u10	exp.ad	BEL_MBEL(U,WBEL(S,u6))
				ca:u01	
(U3) User: Thank you.					
		perception problems	u11	exp.und	BEL_MBEL(U,WBEL(U,BEL(S,u3)))
			u12	exp.und	BEL_MBEL(U,WBEL(U,BEL(S,u4)))
			u13	exp.und	BEL_MBEL(U,WBEL(U,BEL(S,u5)))
			u14	exp.und	BEL_MBEL(U,WBEL(U,BEL(S,u6)))
(S4) System: Sorry?					
				ca:	u11 to u14
(U5) User: Thank you.					
s9	und	BEL(S,u3)			
s10	und.	BEL(S,u4)			
s11	und.	BEL(S,u5)			
s12	und.	BEL(S,u6)			
s13	exp.und	BEL_MBEL(S,WBEL(U,s9))	u11	exp.und	BEL_MBEL(U,WBEL(U,s9))
s14	exp.und	BEL_MBEL(S,WBEL(U,s10))	u12	exp.und	BEL_MBEL(U,WBEL(U,s10))
s15	exp.und	BEL_MBEL(S,WBEL(U,s11))	u13	exp.und	BEL_MBEL(U,WBEL(U,s11))
s16	exp.und	BEL_MBEL(S,WBEL(U,s12))	u14	exp.und	BEL_MBEL(U,WBEL(U,s12))
(S6) System: You're welcome.					
			u15	und.	BEL(U,s9)
			u16	und.	BEL(U,s10)
			u17	und.	BEL(U,s11)
			u18	und.	BEL(U,s12)
s17	exp.und	BEL_MBEL(S,WBEL(S,u15))	u19	exp.und	BEL_MBEL(U,WBEL(S,u15))
s18	exp.und	BEL_MBEL(S,WBEL(S,u16))	u20	exp.und	BEL_MBEL(U,WBEL(S,u16))
s19	exp.und	BEL_MBEL(S,WBEL(S,u17))	u21	exp.und	BEL_MBEL(U,WBEL(S,u17))
s20	exp.und	BEL_MBEL(S,WBEL(S,u18))	u22	exp.und	BEL_MBEL(U,WBEL(S,u18))
			u23	st:u1	BEL_MBEL(U,BEL(U,s1))
			u24	st:u2	BEL_MBEL(U,BEL(U,s2))

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 17–24.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Incomplete Knowledge and Tacit Action: *Enlightened Update* in a Dialogue Game

Luciana Benotti
TALARIS Team - LORIA (Université Henri Poincaré, INRIA)
BP 239, 54506 Vandoeuvre-lès-Nancy, France
Luciana.Benotti@loria.fr

Abstract

This paper has two main aims. The first is to show how planning capabilities have been integrated into FrOz, a text adventure game presented in (Koller et al., 2004). Second, we demonstrate that the resulting system offers a natural laboratory for exploring the theory of enlightened update presented in (Thomason et al., 2006). In particular, we shall discuss how this theory applies in a setup with incomplete background knowledge.

1 Introduction

In this paper we investigate, in a simplified and formalised setup, how the information that allows two interlocutors to understand each other correctly is constructed and exploited during a conversation.

Let us start, right away, with an everyday example of the phenomenon we want to investigate. *A few days ago, my mother told my sister: "Please, buy some food for Tiffy." Then my sister took some money from a kitchen drawer, went to the grocery store that is near my primary school, bought a pack of low fat cat food with salmon flavour, and carried the food back home.* And this is exactly how my mother expected her to act. Why? Because both of them know that my sister is always low in cash, that at home there is always money in a particular kitchen drawer, that the grocery store near my primary school is the cheapest one, and that Tiffy is our pet cat, who is getting a bit fat and likes salmon. Is that all? Not quite. They also know that in order to buy something you need money, that in order to open a drawer you need to pull it, and many other things that are usually taken for granted.

Here, my mother and my sister exploited the large amount of information they share in order to leave several actions *tacit*. In conversation, this strategy is not merely valid, it is frequent and pervasive. We are going to investigate it in a 'dialogue game,' a conversational setup simplified in several ways. To start with, i) the interaction is restricted to a set of requests[1] between two interlocutors, with well defined preconditions and effects. Also, ii) the requests can be issued only by one of the interlocutors (who we will call 'the player'), the other (called 'the game') is limited to accepting and executing, or refusing the request. To complete the picture, iii) 'the game' has complete and accurate information about the conversational context (called 'the game scenario'), while 'the player' may have incomplete and even incorrect information.

Our setup is formalised in the implementation of a text adventure engine called FrOz Advanced (FrOzA). Text adventures are computer games that simulate a physical environment which can be manipulated by means of natural language requests (i.e., commands issued to the game). The system provides feedback in the form of natural language descriptions of the game world and of the results of the players' actions. FrOzA extends the text adventure FrOz (Koller et al., 2004) with planning capabilities. This added inference ability allows FrOzA to discover actions left tacit by the player.

This paper has two main aims. The first is to show (in Section 2) how planning capabilities can be integrated into the Description Logic (Baader et al., 2003) based inference architecture provided by FrOz. Second, we wish to demonstrate (in Section

[1] By 'request' we refer to the first part of an adjacency pair (request, acceptance/refusal) as defined in (Clark and Schaefer, 1989)

3) that the resulting system, namely FrOzA, offers a natural laboratory for the theory of *enlightened update* presented in (Thomason et al., 2006). The theory of enlightened update suggests how shared information (usually referred to as *common ground* (Clark, 1996)) is exploited and constructed in the light of a computational framework for reasoning in conversation. We use FrOzA not only to obtain a concrete account of enlightened update theory, but to extend it for handling incomplete background knowledge as well.

2 FrOzA

The architecture of FrOzA is shown in Figure 1; its three main processing modules are depicted as ellipses. The language understanding module parses the command issued by the player and constructs its semantic representation. The language generation module works in the opposite direction, verbalising the results of the execution of the command. The action handling module is in charge of performing the actions intended by the player.

All three modules make heavy use of inference services (represented as dashed lines in the figure) in order to query and update the components of a game scenario (depicted as rectangles). The processing modules are independent of particular game scenarios; by plugging in a different game scenario the player can play a different game.

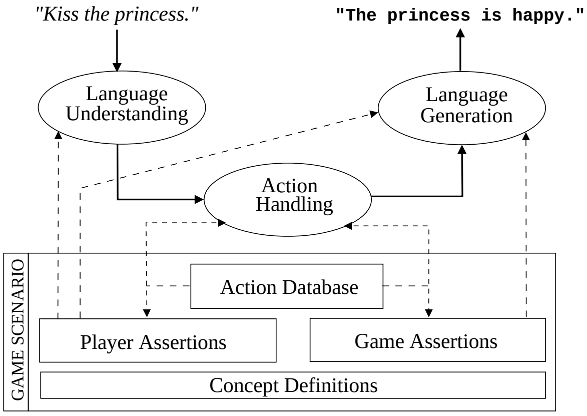


Figure 1: Architecture of FrOzA

In fact, it is in its reasoning abilities that FrOzA extends the original version of the system (FrOz). Thanks to its planning capabilities, FrOzA is able to discover actions intended by the player but left tacit by her. In order to infer these actions, FrOzA uses the planner Blackbox (Kautz and Selman, 1999). Like FrOz, FrOzA uses the theorem prover RACER (Haarslev and Möller, 2001) to query and modify the Description Logic knowledge bases according to the instructions encoded in the action database.

In the rest of the section we will describe the components of FrOzA that are relevant for the purposes of this paper. In Section 2.1 we will describe how FrOzA models a game scenario in its knowledge bases. In Section 2.2 and 2.3 we will explain in detail how actions are handled; in particular, we show how the execution of an action depends on the current game scenario, and how the successful execution changes the scenario. This will pave the way for our discussion of enlightened update in Section 3.

2.1 Modelling a game scenario

FrOzA uses Description Logic (DL) knowledge bases (KB) to codify assertions and definitions of the concepts relevant for a given game scenario. A DL knowledge base is a pair (TBox, ABox) where the TBox is a set of definitions and the ABox a set of assertions about the objects being described in the KB (such objects are usually called individuals). Actually, FrOzA uses two knowledge bases, which share the TBox and differ only in their ABoxes. The common TBox defines the key concepts in the game world and how they are interrelated. Some of these concepts are basic notions (such as *object*) or properties (such as *alive*), directly describing the game world, while others define more abstract notions like the set of all the individuals a player can interact with (the individuals that are *accessible* to the player).

The ABoxes specify properties of particular individuals (for example, an individual can be an *apple* or a *player*). Relationships between individuals are also represented here (such as the relationship between an object and its location).

One of the knowledge bases (the game KB) represents the *true state* of the game world, while the other (the player KB) keeps track of the player's *beliefs* about the game world. In general, the player KB will not contain all the information in the game KB because the player will not have explored the world completely, and therefore will not know about all the individuals and their properties. In fact, it might also be the case that the player KB contains information that is inconsistent with the game KB. The game can deliberately hide effects of an action from the player; pushing a button might have an effect that the player cannot see.

Crucially, a game scenario also includes the definitions of the actions that can be executed by the player (such as the actions *take* or *eat*). Each action is specified (in the action database) as a STRIPS-like operator (Fikes et al., 1972) detailing its arguments, preconditions and effects. The preconditions indicate the conditions that the game scenario must satisfy so that the action can be executed; the effects determine how the action changes the game scenario when it is executed.

2.2 Handling a single action

In this section we are going to explain in detail how an action issued by the player can change the game scenario.

To illustrate our explanation, let us consider a concrete input and analyse how it is handled by the system. Suppose that the player has just said "Take the key." The semantic representation of this command (obtained by the language understanding module) will be the ground term `take(key1)` (where `key1` represents the only key that the player can see in the current state of the game). This ground term will be passed to the next processing module in the architecture.

When a ground term is received by the action handling module, it is matched against the list of action schemas. The action schema that will match the ground term of our example is:

```
action:
  take(X)
preconditions:
  accessible(X),
  takeable(X),
  not(in-inventory(X))
effects:
  add: in-inventory(X)
  del: has-loc(X indiv-filler(X has-loc))
player effects:
  add: in-inventory(X)
  del: has-loc(X indiv-filler(X has-loc))
```

The term `X` in the above schema is a variable that gets bound to the actual argument of the action. In our example, `X` would be bound to the constant `key1`, and thus the preconditions and effects will become ground terms. Once the action schema is instantiated, it is time to check that the action can be executed. An action can be executed if all its preconditions are satisfied in the current game KB. The preconditions can require that individuals belong to certain concepts or that they are related by certain roles. For example, the execution of the action `take(key1)` requires that the key is accessible to the player (`accessible(key1)`), that it is small enough to be taken (`takeable(key1)`) and that it is not carried by the player already (`not(in-inventory(key1))`). The theorem prover RACER is used to query the current *game KB*, thereby checking that the preconditions are satisfied.

If the action can be executed, the *game KB* is updated according to the effects of the action. In our example, the key will no longer be in its original location but it will be carried by the player. The original location of the key is obtained by sending the query `indiv-filler(key1 has-loc)` to RACER. A RACER query is embedded in an action schema when the action depends on properties of individuals not explicitly mentioned in the player command (such as the location of the key).

Once the game executed the action, the player needs to know that the action succeeded. To this end, the player effects in the action schema are communicated to the player by the generation component and asserted in the *player KB*.

If the command cannot be executed in the current game scenario, the first precondition that failed is communicated to the player and both KBs remain unchanged.

2.3 Interpreting the player intention

Now that we know how the actions module handles a simple action, let us explain how *ambiguous commands* and *tacit actions* are handled in FrOzA.

The input of the action module is not a single ground term but a list of possible readings of the input sentence. The list will contain exactly one reading only if the sentence is not ambiguous (as in the example in the previous section). Otherwise, the list will contain one entry for each different reading. For example, the sentence "Unlock the door with the key" is syntactically ambiguous and has two possible readings, one in which the propositional phrase "with the key" modifies the verb "unlock" and another in which it modifies the noun phrase "the door." Sentences can also be referentially ambiguous. For instance, the sentence "Take it" has as many readings as there are salient referents in the game scenario. Each reading is itself a list which represents a sequence of actions to be performed one after the other. For example, every reading of the sentence "Take the key and unlock the door with it" will contain two ground terms, one for each action in the sequence.

If the input sentence has more than one reading, FrOzA decides among them by trying each action sequence in parallel. When an action fails, the entire reading it belongs to is discarded. For example, the reading of the command "Take it and eat it" which resolves both occurrences of "it" to a key, will be discarded because a key is not edible, although it can be taken.

If only one reading succeeds, the game assumes that this is the command the player had in mind, and commits to the end result of the sequence. If more than one sequence is possible, the game reports an unresolved ambiguity. For instance, the game will report an ambiguity if both readings of the command "Unlock the door with the key" are executable in the current game scenario.

The inference capabilities discussed so far are common to FrOz and FrOzA; we now turn to what sets FrOzA apart and will lead us to discuss the theory of enlightened update: *planning capabilities*. Planning is used when no reading is executable, for analysing whether the command includes *tacit actions*. For each failed reading FrOzA tries to find a *sequence of actions* (i.e., a *plan*) which transforms the current game scenario into a scenario where the reading can succeed. If no such plan exists, the reading is discarded, otherwise the plan is concatenated before the reading, enlarging the original sequence of actions. The new list of readings built in this way is reinserted into the action handling module and its execution proceeds as usual.

In order to illustrate the previous behaviour of FrOzA, let us consider again the command "Unlock the door with the key" but now suppose that none of its two readings is executable in the current game scenario. One of the readings fails because there is no "door with the key" in the current game scenario. The other reading cannot be directly executed because the key is not in the player's hands but on a table in front of her. However, for this second reading a plan can be found, namely "to take the key" before unlocking the door; although "take the key" was left tacit by the player, it can be inferred from the game scenario. This plan is concatenated before the original reading and the extended reading is processed again by the action handling module. This time, the input of the action module will be the sequence of actions "Take the key and unlock the chest with it", making explicit the tacit action.

In order to infer tacit actions, FrOzA uses the planning services provided by the planner Blackbox (Kautz and Selman, 1999). Blackbox works by fixing the length of the plan in advance and iteratively deepening it. This behaviour makes it particularly well suited for our needs because it finds optimal plans (minimal in the number of actions) and does it fast. Fast responses are essential for a natural interaction with the player. For a detailed description of the performance of Blackbox in FrOzA see (Benotti, 2006a; Benotti, 2006b). Moreover, optimal plans are crucial, otherwise actions which are executable in the game scenario but completely irrelevant to the player command might be included as tacit actions. For example, a non-optimal planner might not only "take the key" as in our example, but also take and drop other arbitrary objects as well.

The input required by Blackbox are STRIPS-style problems specified in the Planning Domain Definition Language (Gerevini and Long, 2005) which includes the standard elements of a planning specification: the initial state, the available actions, and the goal.

In next section we will present a theoretical account of the intuitions hinted at here by making use of the insights provided by the theory of enlightened update. In particular, we will analyse what information the elements of the planning specifications should contain.

3 Enlightened update in FrOzA

We will now use FrOzA as a laboratory for exploring the theory of enlightened update (Thomason et al., 2006). Using FrOzA we shall construct, step by step, an accurate account of the main principles behind this theory.

The intuition behind enlightened update theory is that when the speaker utters a sentence, as my mother did in our first example, she is not only trying to achieve the obvious effects of the utterance, but is also communicating the ways in which she assumes the world to be, and on which the success of the utterance depends.

Let us make this approach concrete through an example in our game setup. Suppose that the player is inside a room with a locked door while she is holding a golden key in her hands. Then she inputs the command "Unlock the door with the golden key," which is mapped to the semantic representation `unlock(door1 key1)`. The intention behind this utterance is twofold. It is clear that the player wants the game state to be updated according to the effects of the action, that is, she wants to have the door unlocked. But the player also expects the game to recognise the assumptions she is making and on which the success of the utterance depends. In particular, she assumes that the golden key fits into the door lock.

This strategy for updating the shared knowledge is stated formally as the following principle:

ENLIGHTENED UPDATE (EU): "An agent's public performance of an action [A] that is mutually known to require a commitment C for its successful performance will add to the mutual information the proposition that the agent believes C." (Thomason et al., 2006, p.15).

It is important to notice that in order to be able to perform an EU it must be mutually known that the action, which is being performed publicly, requires its preconditions. In our setup this means that we assume that the exact preconditions required for the successful performance of the action `unlock` are mutually known (by the player and the game). Such an assumption is represented in the action schema below, which specifies the player preconditions to be equal to the original preconditions of the action.

```
action:
  unlock(door1 key1)
preconditions:
  locked(door1), key(key1),
  in-inventory(key1), fits-in(key1 door1)
player preconditions:
  locked(door1), key(key1),
  in-inventory(key1), fits-in(key1 door1)
effects:
  del: locked(door1)
  add: unlocked(door1)
player effects:
  del: locked(door1)
  add: unlocked(door1)
```

After this action (`unlock(door1 key1)`) is executed successfully, the player will believe that "the golden key" is a key, and that it is in her hands, facts that she already knew. However, she will also believe that the door is now unlocked, the obvious effect of the action; and that the golden key fits in the door lock, the assumption she made and was confirmed by the success of the action. This means that, when an action is executed, the *player KB* will be updated not only with the effects of the action but also with its preconditions. When performing this update, the order in which the changes are made is important in order to leave the KB in the intended state. Concretely, the KB should be first updated with the player preconditions and then with the player effects. Otherwise, the preconditions might undo the effects of the action. Moreover, the updates that retract information from the KB have to be performed before the ones that assert information, in order to avoid introducing an inconsistency in the KB.

This is the easy case, but what if the action cannot be directly executed (that is, some of its preconditions are false) in the current game scenario? The EU principle extends naturally to cover these cases. And, in fact, these are the cases where the theory of enlightened update is able to bridge the gaps that arise in everyday interactions.

3.1 Enlightened Update with Tacit Actions

To analyse how the EU principle is extended, let us modify our running example a bit in order to return to the game scenario we analysed intuitively in Section 2.3. Suppose that the player does not have a key and she is looking around searching for a way to unlock the door when the game says that there is a golden key lying on a table in front of her. Then she inputs the command "Unlock the door with the golden key." Hence, according to the EU principle, the player knowledge base should be updated with the preconditions of the action. However, one of the preconditions of this action, namely `in-inventory(key1)`, is false in the current game scenario (that is, in both KBs). Clearly, the precondition cannot just be added to the player KB because this will cause a contradiction, but this precondition can be *made* true in the game scenario by *performing the appropriate actions*.

The theory of enlightened update defines the following refinement of the EU pattern to handle exactly this situation:

EU WITH TACIT ACTIONS (EU/TA): "[Assume that] C is a condition that can be manipulated by an audience H. An agent S is observed by H to be doing A while C is mutually known to be false. H then acts [tacitly] to make C true, and S expects H to so act." (Thomason et al., 2006, p.36)

In our example, the player is not holding the key and she knows it, and she is trying to unlock the door anyway, knowing that in order to unlock a door you need to have the key in your hands. Hence, FrOzA should act to make `in-inventory(key1)` true. And it does so by executing tacitly the action `take(key1)`.

We should notice here that an action can be left tacit by the speaker, and recognised correctly by the hearer, only if the *effects* of the action are *mutually known* by the conversation partners.

3.2 Enlightened Update and Incomplete Background Knowledge

In (DeVault and Stone, 2006) the theory of Enlightened Update is implemented and tested in COREF, a conversational agent which uses enlightened update to interactively identify visual objects with a human user. In FrOzA we also implement and test the theory of enlightened update but with an added kind of uncertainty: incomplete background knowledge. In COREF, both interlocutors are assumed to have the same background information. In FrOzA, on the other hand, the game has complete and accurate information about the game world, while the player starts the game without information and acquires it as the game evolves. In this setup, modelling enlightened update highlights the issues involved when one of the interlocutors has incomplete background knowledge. Moreover, it illustrates the point that, as conversation evolves, background knowledge accumulates and that a conversational system can use this information to engage in more flexible and robust conversation.

The key question in FrOzA is how it is able to infer the 'appropriate' tacit actions in a setup with incomplete background knowledge. In principle, it just needs to provide Blackbox with the 'appropriate' inputs mentioned in Section 2.3: the initial state, the goal, and the available actions. However, the question of 'what these three elements should contain' raises a number of subtle issues. Their discussion will highlight the kinds of problems that need to be considered when background knowledge is incomplete.

3.2.1 The initial state

The first question is to decide the information that is needed for the initial state. In FrOzA, two types of information are registered: the objective information in the game KB and a subjective view in the player KB. Which of these should be used in order to discover tacit actions? In fact, we need both. Let us analyse this decision by extending our example once again. Suppose that the golden key, which was lying on the table, was taken by a thief without the player knowing. As a consequence, the key is on the table in the player KB, but in the game KB the thief has it. In this new scenario, the player issues the command "Unlock the door with the golden key." If we included in the initial state the objective information of the game KB, FrOzA would automatically take the key from the thief (for example, by using the steal action) and unlock the door for the player, while the player is not even aware where the key actually was. This is clearly inappropriate. Now, if our initial state includes the information in the player KB, FrOzA would decide to take the key from the table and unlock the door with it. But this sequence of actions is not executable in the game world because the key is no longer accessible (the thief has it). More generally, a sequence of tacit actions found by reasoning over the player KB might not be executable in the game world because the player's KB may contain information that is inconsistent with respect to the game KB. Hence, we need both KBs: we infer the actions intended by the player using the information in her KB but we have to verify this sequence of actions on the game KB to check if it can actually be executed. The *action inference* step is done using the planning services provided by Blackbox on the subjective information, and the *action executability* step is done using the reasoning services provided by RACER on the objective information. In COREF, by way of contrast, once the tacit actions are inferred they do not need to be checked for executability on the objective information. This is because the COREF setup does not allow any of the interlocutors to have a subjective view of the information; both interlocutors are assumed to share the objective information and hence, tacit actions are inferred solely on the basis of objective information.

An interesting consequence of the fact that the FrOzA setup handles incomplete background knowledge is that we can investigate how this background knowledge accumulates and how it affects the interaction. And it turns out that the more the player knows about the game world, the more actions can be left tacit. For example, suppose that after opening the door, the player locked it behind her and continued to the following rooms investigating the game world. After a while she is back and wants to open the door again. This time it is enough for her to say "Open the door", instead of "Unlock the door with the golden key", because she already knows, and the game knows that she knows, which key fits into this lock.

As a consequence, we can drop a simplifying assumption made in (Thomason et al., 2006), namely that whether an action is public or tacit is a *static* matter, corresponding to an arbitrary split in the action database. In FrOzA this distinction is *dynamic* and correlates with the growth of background information.

3.2.2 The goal

The two remaining questions are what the goal and the actions of the planning problem should be. We believe that answering these two questions is also non-trivial, as it was not trivial to define the initial state. However, we have not yet analysed the subtleties involved in these two issues; here we simply present our initial approach.

Let us start defining what the goal should be. According to EU principles, the game should act to make the preconditions of the action true with two restrictions. First, it must be possible for the game to manipulate the preconditions. And second, the action must be mutually known to require its preconditions. Hence, we define the goal as the player preconditions of the action commanded by the player, excluding those that cannot be manipulated by the actions in the action database.

For example, when the player says "Unlock the door with the key" the goal of the planning problem will only include the atoms:

```
locked(door1),
in-inventory(key1),
```

The preconditions that cannot be manipulated by the actions available in the action database, such as `key(key1)` (something that its not a key cannot be transformed into one) and `fits-in(key1 door1)` (if the key does not fit into the lock it is not possible to make it fit) are not included in the goal.

3.2.3 The actions

To complete the picture, the actions available to the planner are all the actions in the game action database. Its preconditions will correspond to the player preconditions and its effects to the player effects. For the moment, we are assuming that the preconditions and the effects of the actions are shared by the game and the player. Hence, the player preconditions and the preconditions of an action coincide; as well as the player effects and the effects. Relaxing this simplifying assumption, would introduce more dynamism in the distinction between tacit and public actions, and hence would better reflect the case of real conversation.

4 Conclusions

In this paper we have described FrOzA and used it to explore the enlightened update theory.

The FrOzA setup shows that enlightened update can be implemented using an off-the-shelf reasoning tool such as Blackbox. At present, the solution provided by this setup is not logically complete because our two inference tools (RACER and Blackbox) work independently and are not capable of sharing information (see (Benotti, 2006b) for the technical details). However, we believe that the present implementation is the kind of laboratory that theories such as enlightened update needs. We leave the study of complete reasoning mechanisms and a comparison between our setup and the one implemented in (Thomason et al., 2006) for further research. We mention in passing that integrating planning capabilities in the framework of a Description Logic reasoner is a topic of current research (see (Baader et al., 2005; Liu et al., 2006)).

We have tested the theory of enlightened update with an added kind of uncertainty common in conversation: incomplete background knowledge. This test yielded two interesting consequences. First, the theory applies but raises a number of subtle issues on the kind of required information, and second, the division between tacit and public actions becomes dynamic. In (Thomason et al., 2006) whether an action is public or tacit is a static matter, corresponding to an arbitrary split in the action database. In FrOzA, this distinction correlates with the growth of background information; we believe this to be in line with 'the granularity of conversation' as defined in (van Lambalgen and Hamm, 2004) another point which requires further work.

But more remains to be done. There is a deeper kind of incomplete background knowledge, namely when the action preconditions and effects are not mutually known, i.e. when the task model is not shared by the interlocutors. We believe that accounting with such uncertainty in a conversation is one of the most challenging problems that theories such as enlightened update face nowadays.

Acknowledgements

I would like to thank the members of the Dialogue Discussion Group of the TALARIS Team at LO-

RIA for an enthusiastic welcome, and lively discussions.

References

Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Franz Baader, Carsten Lutz, Maja Milicic, Ulrike Sattler, and F. Wolter. 2005. Integrating description logics and action formalisms: First results. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI-05)*, Pittsburgh, PA, USA.

Luciana Benotti. 2006a. “DRINK ME”: Handling actions through planning in a text game adventure. In Janneke Huitink and Sophia Katrenko, editors, *XI ESSLLI Student Session*, pages 160–172.

Luciana Benotti. 2006b. Enhancing a dialogue system through dynamic planning. Master’s thesis, Universidad Politécnica de Madrid.

Herbert Clark and Edward Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13(2):259–294.

Herbert Clark. 1996. *Using Language*. Cambridge University Press, New York.

David DeVault and Matthew Stone. 2006. Scorekeeping in an uncertain language game. In *The 10th Workshop on the Semantics and Pragmatics of Dialogue (Brandial 2006)*, University of Potsdam, Germany.

Richard Fikes, Peter Hart, and Nils Nilsson. 1972. Learning and executing generalized robot plans. *Artificial Intelligence*, 3:251–288.

Alfonso Gerevini and Derek Long. 2005. Plan constraints and preferences in PDDL3. Technical Report R.T. 2005-08-47, Università degli Studi di Brescia, Italy.

Volker Haarslev and Ralf Möller. 2001. RACER system description. In *Proceedings of International Joint Conference on Automated Reasoning (IJCAR 01)*, number 2083 in LNAI, pages 701–705, Siena, Italy.

Henry Kautz and Bart Selman. 1999. Unifying SAT-based and graph-based planning. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99)*, pages 318–325, Stockholm, Sweden.

Alexander Koller, Ralph Debusmann, Malte Gabsdil, and Kristina Striegnitz. 2004. Put my galakmid coin into the dispenser and kick it: Computational linguistics and theorem proving in a computer game. *Journal of Logic, Language and Information*, 13(2):187–206.

Hongkai Liu, Carsten Lutz, Maja Milicic, and Frank Wolter. 2006. Reasoning about actions using description logics with general TBoxes. In Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa, editors, *Proceedings of the 10th European Conference on Logics in Artificial Intelligence (JELIA 2006)*, volume 4160 of *Lecture Notes in Artificial Intelligence*, pages 266–279. Springer-Verlag.

Richmond Thomason, Matthew Stone, and David DeVault. 2006. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In Donna Byron, Craige Roberts, and Scott Schwenter, editors, *Presupposition Accommodation*. Ohio State Pragmatics Initiative, draft Version 1.0.

Michiel van Lambalgen and Fritz Hamm. 2004. *The Proper Treatment of Events*. Blackwell.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 25–31.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Push-to-talk ain't always bad! Comparing Different Interactivity Settings in Task-oriented Dialogue

Raquel Fernández, David Schlangen and **Tatjana Lucht**
Department of Linguistics
University of Potsdam, Germany
{raquel,das,lucht}@ling.uni-potsdam.de

Abstract

Restrictions of interactivity in dialogue are often seen as having negative impact on the efficiency of the dialogue, as they affect the ability to give immediate feedback (Whittaker, 2003). We have conducted experiments with one such restriction common in spoken dialogue systems, namely *push-to-talk*. While our results confirm many predictions from the literature (fewer but longer turns; reduction of positive feedback), we found no significant impact on task-efficiency. Our analysis of the grounding strategies of the subjects shows that the restriction actually induced a more cautious strategy that proved advantageous for our matching task, and that giving *negative* feedback in the form of clarification requests was not affected by the restriction.

1 Introduction

Natural, freely regulated turn-taking as described for example in the seminal paper (Sacks et al., 1974) is still a long way off for spoken dialogue systems. Unable to interpret in real-time the various information sources that have been investigated as influencing turn-taking (see e.g. (Caspers, 2003) on the role of syntax and prosody in Dutch turn-taking), dialogue systems resort to simpler strategies like using *time-outs* (where a silence by the user is interpreted as the intention to yield the turn) and *push-to-talk*, where the turn is held explicitly by pushing a button when speaking (see e.g. (McTear, 2004) for a discussion of these methods).

In the work reported here, we wanted to investigate in isolation the effect of the latter strategy, *push-to-talk*, on the shape of task-oriented dialogue. For this we conducted an experiment where we let subjects do a conversational task (a variant of the matching tasks of (Krauss and Weinheimer, 1966; Clark and Wilkes-Gibbs, 1986) either with free turn-taking or with turn-taking controlled by *push-to-talk*. The theoretical literature makes clear predictions about such settings (fewer, longer turns with less efficient descriptions; see next section). While our findings confirm some of those, we found no negative impact on task success, which on further analysis seems due to a different grounding strategy induced by the restriction.

The remainder of the paper is structured as follows. In the next section, we briefly review some of the theoretical predictions of effects of interactivity restrictions. We then describe our task and the experimental conditions, procedure and method. In Section 5 we describe our analysis of the turn and dialogue act structure of the collected dialogues. The puzzling result that the restricted dialogues were not less efficient than the unrestricted ones is further analysed in Section 6 by looking at more global strategies used by the participants. We close by briefly discussing our results and possible further work that could be done to corroborate our findings.

2 Interactivity and the Shape of Dialogue

In pragmatics it is common to assume that conversation, like any other collaborative and interactive action, is governed by economy principles such as the Gricean maxims (Grice, 1975) or the more recently formulated *principle of least collaborative effort* (Clark and Wilkes-Gibbs, 1986). The latter states that participants will try to maximise the success of their collective purpose while minimising costs. As (Clark and Brennan, 1991) point out, the costs of communicative actions are dependent on features of the medium used, like copresence, visibility, audibility, cotemporality or simultaneity. For instance, using short feedback acts like “uhu”, which is effortless in face-to-face communication, becomes slightly more costly when communicating via email, while their cost is definitely much higher when communicating via non-electronic letters.

Mediums in which participants communicate by speaking (as opposed to for instance typing), receive messages in real time (cotemporality) and can communicate at once and simultaneously (simultaneity) afford full *interactivity* (Whittaker, 2003).

Interactivity plays a central role in theories of grounding like those of Clark and colleagues (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989). It enables speakers to interrupt and complete each other’s utterances and allows for constant feedback in the form of often concurrent backchannels, which help to determine whether the conversation is on track and facilitate quick repair of misunderstandings.

One of the predictions of these theories is that settings that preclude or restrict interactivity, like half-duplex channels, will disrupt understanding and quick repair and show less incremental content, thereby leading to more time and errors. This has been confirmed by several studies, like (Krauss and Weinheimer, 1966; Clark and Krych, 2004), that have investigated non-interactive settings that lack cotemporality and simultaneity. In these studies speakers, who are engaged in a referential communication task, talk to a tape recorder for *future* addressees. Interactivity is completely precluded and therefore speakers do not get any form of feedback. (Krauss and Weinheimer, 1966) found that speakers who do not get feedback from addresses take longer and make more elaborate references. Similarly, (Clark and Krych, 2004) showed that references designed without feedback are “inferior in quality” and some are even impossible to grasp.

The experiments we report here investigate the effects of restricting interactivity by using a half-duplex channel managed by *push-to-talk*, which allows cotemporality but inhibits simultaneity. As will be seen in subsequent sections, our results confirm many predictions from the literature, like the presence of fewer but longer turns and a significant reduction of positive feedback (as observed in other studies that used half-duplex channels like e.g. that of (Krauss and Bricker, 1967)). Surprisingly, however, we found that this did not lead to any significant impact on task-efficiency (Fernández et al., 2006). One of the aims of the present paper is to shed some light on the reasons behind this puzzle.

3 Task and Experimental Setting

The task we have asked our experimental subjects to do is a variant of the reference tasks pioneered by (Krauss and Weinheimer, 1964; Krauss and Weinheimer, 1966). In our task, a *player* instructs an *executor* on how to build up a *Pentomino* puzzle (see below). The player has the full solution of the puzzle, while the executor is given the puzzle outline and the set of loose pieces. The solution and the outline of the puzzle are shown in Figure 1.

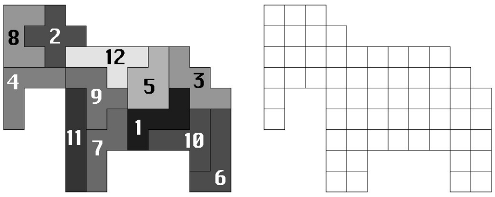


Figure 1: Solution and Outline

The player is asked to tell the executor how the puzzle is assembled following a particular order of the pieces, as given by the numbers on the solution in Figure 1. This enforces a reconstruction process common to all collected dialogues, which allows for more systematic comparisons. The pieces that the executor manipulates are not numbered and are all the same colour. Both player and executor were aware of the information available to each other.

During the experiment the player and the executor were in different rooms and communication between them was only verbal. They could not see each other and they did not have any visual information about the state of the task (i.e. the player could not visually monitor the progression of the reconstruction process).

We investigate two different conditions that differ in degree of interactivity. In a first fully interactive condition, player and executor communicate by means of headsets and the channel is continuously open, as it would be for instance in a telephone conversation. In the second condition interactivity is restricted. Here subjects communicate using walkie-talkies that only offer a half-duplex channel that precludes simultaneous communication. Speakers have to press a button in order to get the turn, hold it to keep it, and release it again to yield it (a 'beep' is heard when the other party yields the turn). We refer to these two conditions as *free turn-taking* (FTT) and *push-to-talk* (PTT), respectively.

4 Procedure and Methods

The experiments involved 20 subjects, 11 females and 9 males, grouped in 10 player-executor pairs. Five pairs of subjects were assigned to each of the two conditions: two female-female pairs, one male-male pair, and two female-male pairs used FTT, while two female-female pairs, two male-male pairs, and one female-male pair used PTT. All subjects were German native speakers between 20 and 45 years old, and the conversations were in German.

The 10 dialogues collected make up a total of 194.54 minutes of recorded conversation. The recordings were transcribed and segmented using the free software Praat (Boersma, 2001). The transcribed corpus contains a total of 2,262 turns and 28,969 words.

To keep a visual record of the progression of the task, the board with the outline and the pieces that the executor manipulated was videotaped during task execution. This gives us a corpus of 10 videos, which have been informally analysed but not systematically annotated yet.

5 Analysis 1: Turn & Act Structure

5.1 Coding

We used MMAX2 (Müller and Strube, 2001) to annotate each utterance with one or more dialogue acts (DAs). We distinguish between task and grounding acts. Task acts are further classified into task-execution (including a tag for description acts where a piece or a location are described) and task-management acts, while grounding acts include different types of feedback acts, as well as clarification requests (CRs). Table 1 shows an overview of the DA taxonomy used.

DA Tag	Meaning
`Task`	
⊢ `Task-Execution`	
`descr_piece`	Description of piece
`descr_pos`	Description of position
`req_info`	Request of task-related info
`req_action`	Request for action
`sugg_error`	Suggest error in task
⊢ `Task Management`	
`dis_sett`	Discuss setting
`dis_stra`	Discuss strategy
`coor_task`	Coordinate task execution
`Grounding`	
⊢ `pos_fback`	Acknowledgement
⊢ `neg_fback`	Rejection or correction
⊢ `ask_conf`	Request for acknowledgement
⊢ `CR`	Clarification request
`Other`	Incomplete and other acts

Table 1: DA Taxonomy

5.2 Results

An analysis of turn patterns shows that our PTT dialogues contain roughly half as many turns as the FTT dialogues, with the turns however being on average twice as long as the FTT turns (in seconds: 7.21 sec and 3.71 sec on average respectively; this difference is statistically significant at $p < 0.01$; in number of words: 20.2 vs 11.3 on average; $p < 0.05$).[1]

Figure 2 plots the number of turns per dialogue in each condition and for each participant role. The diagram allows us to see that the number of turns is rather constant across PTT dialogues, with equal number of contributions by player and executor. This indicates that in this condition player and executor do indeed take turns; i.e. each contribution by one is followed by one by the other. In the FTT dialogues there is a higher variation among pairs of participants and the number of turns contributed by the executor is higher. This in turn indicates that often executors' contribu-

[1] Unless otherwise stated, all significances reported in this paper are calculated with a t-test.

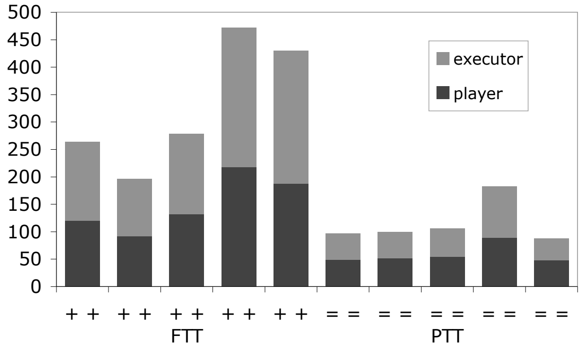


Figure 2: Number of turns per dialogue

tions are provided concurrently to those of the player. On average, around 35% of FTT turns are given in complete overlap; even when these turns are not counted, the number of turns in FTT is significantly higher ($p < 0.02$).

Despite the differences in turn patterns, pairs of participants in both conditions were able to finish the task in roughly the same time (18.7 min in PTT and 19.8 min in FTT on average; no significant difference). However, pairs in the PTT condition were able to do so using significantly fewer words (2253.6 vs 3540 on average; $p < 0.05$). Table 2 shows the mean number of words per condition and speaker role. As is common in this kind of instructional tasks (e.g. (Clark and Krych, 2004)), instruction givers (players) talk markedly more than instruction followers (executors).

	FTT	PTT
player	2127	1551.2
executor	1413.2	702.4

Table 2: Mean num of words per dialogue

The distribution and length of dialogue acts also helps to highlight some further differences between conditions. Distribution is shown in Table 3. The most significant difference regarding distribution is found in the amount of positive feedback acts, like backchannels and acknowledgements, which is consistently higher in FTT (33.8% vs 25.7% on average; $p < 0.01$ on a χ^2 test on raw numbers). This is still the case when ovelapping turns are not taken into account. The distribution of other grounding acts like negative feedback and CRs, however, is similar in both conditions. As for task acts, PTT dialogues contain a higher proportion of task-related acts than FTT dialogues (45.4% vs 36.7% on average; $p < 0.01$ on a χ^2 test on raw numbers).

	FTT	PTT
task_related	871 (36.7%)	444 (45.4%)
pos_fback	804 (33.8%)	250 (25.7%)
other fback	211 (8.9%)	70 (7.1%)
CRs	361 (15.2%)	161 (16.5%)
other acts	127 (5.4%)	52 (5.3%)

Table 3: Distribution of DAs

The diagram in Figure 3 shows the mean length in words of the four main DA types for each of the two conditions. As can be seen, the length in words of positive and negative feedback acts is roughly the same in PTT and FTT dialogues. CRs tend to contain more words in PTT, although this is not statistically significant. Finally, description acts (which are the lion's share of task acts) contain significantly more words in PTT dialogues than in FTT dialogues (19.8 vs 14.2 on average; $p = 0.05$).

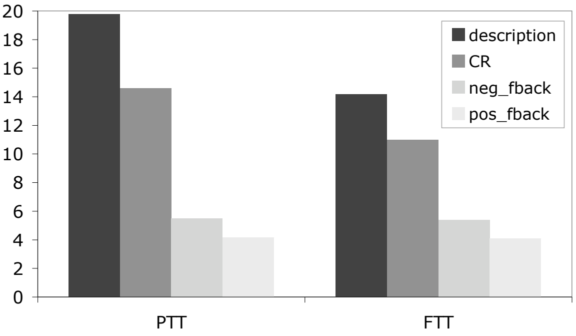


Figure 3: Mean num of words per DA type

5.3 Discussion

Our results confirm the predictions from the literature (e.g. (Krauss and Bricker, 1967; Whittaker, 2003)) that using a unidirectional channel produces less speaker switching and longer turns. We have also seen that description acts contain significantly more words in the PTT condition, which confirms the observation that contributions in non-interactive conditions tend to be more elaborate.

In Section 2, we pointed out that the lack of concurrent bidirectional communication is predicted to disrupt grounding behaviour leading to less shared understanding, which should have negative consequences at the task level. The analysis of dialogue acts has shown that grounding behaviour is certainly disrupted in the PTT condition. Although grounding acts do not vary in number of words across conditions, PTT dialogues show a significant reduction of the amount of positive feedback acts. This is presumably because positive feedback acts like acknowledgements, being very short acts and hence having a relatively high speaker-change overhead, are too costly in this condition. Interestingly, however, the proportion of other grounding acts like negative feedback acts and CRs (that also tend to be shorter) is not affected by the restriction. It seems that for our subjects, giving negative feedback was more essential, while positive feedback could presumably be taken as the default in a condition that made it coslty.

More surprising is perhaps the fact that the restricted interactivity of the PTT condition, with its lack of concurrent turns and its reduced positive feedback, did not lead to overall longer dialogues. Not only were pairs in the PTT condition not slower, but they were able to solve the task using significantly fewer words (see Table 2).

These observations pose a puzzle: Why does the reduction of interactivity in PTT dialogues not have a negative effect in terms of task efficiency (measured w.r.t. length of dialogue and number of words used)? To find an answer to this question, in the next section we analyse the dialogues on a level higher than individual acts, that of task-related moves.

6 Analysis 2: Task & Move Structure

6.1 Coding

The task of reconstructing the Pentomino puzzle can be divided into 12 *moves* or cycles, one for each of the pieces of the puzzle. A *move* as defined here covers all speech that deals with a particular piece, from the point when the player starts to describe the piece (*"Okay, so the next piece looks like a stair case"*) to the point when participants have agreed on the piece and its target location to their satisfaction and move on to the next piece. Sometimes moves are not successful and contain errors that are discovered later on in the dialogue. We call any stretch of speech that deals with the repair of a previous move that had already been closed a *repair sequence*.

Each dialogue contains 12 moves, while the number of repair sequences varies depending on the amount of errors and the uncertainty with which previous moves were grounded.

The video recordings of the board during task execution allow us to determine the grounding status of moves. By looking at the state of the board when a move is considered closed, we can determine whether the move has been successfully grounded or else whether there is a mismatch in common ground.

Using this visual information, we classified moves according to four categories: `correct`, `correct_rep`, `incorrect_inf` and `incorrect_rep`. Moves classified as `correct` were successful moves that did not require any subsequent repair nor double checking. Moves classified as `correct_rep` were successful but were grounded with low confidence and therefore required a repair sequence to confirm their correctness (usually after encountering problems with subsequent pieces). Moves classified as `incorrect_inf` were not successful but problems were discovered by inference by the executor after dealing with other pieces and the repair did not trigger an explicit repair sequence. Finally, moves classified as `incorrect_rep` were not successful and a repair sequence was performed at a later point in the dialogue to deal with the mismatch and repair the problems.

6.2 Results

The diagram in Figure 4 illustrates task progression with respect to the grounding success of the 12 moves (left to right) for each of the 5 dialogues in each of the two conditions.

We can compute a global *error score* for each dialogue by assigning values from 3 to 0 to moves classified as `incorrect_rep`, `incorrect_inf`, `correct_rep` and `correct`, respectively. The score of a dialogue is then the sum of the values obtained in each of the 12 moves, on a scale from 0 to 36. For instance, the top PTT dialogue in Figure 4 has an error score of 3, while the error score of the top FTT dialogue is 7.

In general PTT dialogues obtain lower error

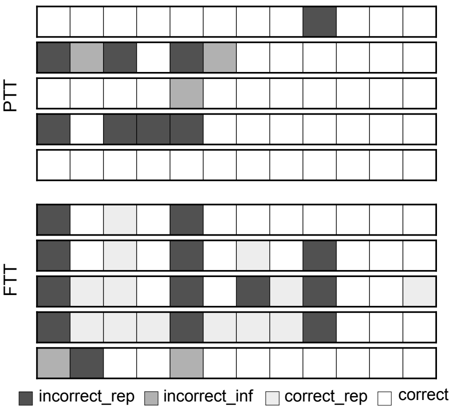


Figure 4: Move success

scores than FTT dialogues (5.8 vs 11.2 on average), although the difference is not statistically significant. This is probably not surprising given that in fact all pairs were able to finish the task successfully in roughly the same time. We find, however, that there is a correlation between error score and number of words in description acts per move (Pearson's correlation coefficient: $r = -0.7, p < 0.05$).

Further contrasts can be identified when looking at error score *per move*. The chart in Figure 5 plots the error score accumulated at each move for each of the two conditions. The score of a move within a condition is computed by adding the scores obtained in each of the five dialogues in that condition.

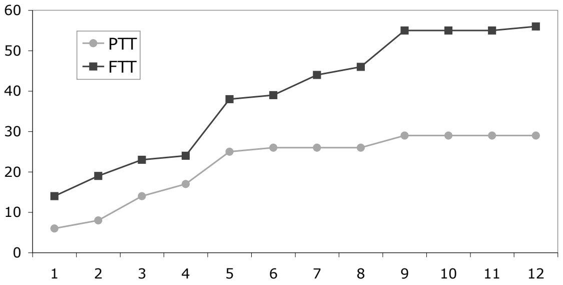


Figure 5: Error score per move

The chart allows us to see that after move 6 PTT pairs hardly make any more mistakes (the error score stays fairly constant from then on to the end of the task). Pairs in the FTT condition, on the other hand, keep on accumulating errors well until move 9. If we look at the amount of time spent on the first 6 moves in each dialogue, we see that, regardless of condition, the percentage of time spent on the first part of the task (up to the end of move 6) correlates with the global error score assigned to each dialogue (Pearson's correlation coefficient: $r = -0.69, p < 0.05$). For instance, the last PTT dialogue in Figure 4, which has an error score of 0, spends more than 8 minutes on the first 6 moves, while the third FTT dialogue, whose error score is 16, deals with the first 6 moves in less than 3 minutes. That is, more time leads to fewer errors.

6.3 Discussion

The analysis of task and move structure shows that, independently of conditions, a strategy whereby more time is spent on more detailed (=more words) descriptions making sure that moves are grounded before proceeding leads to fewer errors. The efficiency of PTT dialogues then can be explained by the fact that the restricted interactivity favours this kind of strategy. In Section 3 we showed that description acts contain a significantly higher number of words in PTT dialogues. Certainly, the fact that speakers can control the length of their turns allows for more detailed, perhaps better planned descriptions. Thus, what other studies of non-interactive settings have described as "overelaboration" (Krauss and Bricker, 1967) actually seems to be an advantage for the task at hand, which requires a fair amount of descriptive talent. 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.

We have seen that subjects in FTT dialogues tend to make more mistakes further ahead in the task. This is in part due to a cascading effect whereby earlier errors lead to more subsequent mistakes. However even when errors are made, they can be recovered relatively fast (there is no correlation between length of dialogue and error score). The time that is not spent on detailed moves is then used in repair sequences.

As the lack of constant feedback makes quick repair more costly in PTT dialogues, subjects in this condition tend to adopt a more cautious strategy where moves are better grounded on a first pass and hence require fewer subsequent repair sequences, or use inference to avoid explicit repair.

7 Conclusions

We have presented the results of experiments that compare two different turn-taking conditions that differ in degree of interactivity: a fully interactive free turn-taking condition and a restricted condition where subjects use a half-duplex channel managed by push-to-talk.

Our results confirm many predictions from the literature, like the presence of fewer but longer turns and a reduction of positive feedback in the restricted condition. Indeed, participants do not produce short acts like positive feedback backchannels when conditions make them expensive; negative feedback acts and CRs however (also being shorter) are produced even under adverse conditions.

The literature also predicts that a reduction of interactivity will disrupt shared understanding and ultimately lead to problems at the task level. However, we found that the restricted condition did not have any significant impact on task-efficiency. Our analysis of the grounding strategies employed by the subjects shows that the restriction in interactivity actually favoured a more adequate strategy (longer and more detailed descriptions) that proved advantageous for our task—a difficult task that requires identification of very abstract referents.

More generally, our results indicate that dialogue participants do not always use the grounding strategy that is best for the task at hand, and that a particular grounding strategy can be "primed" by imposing turn-taking restrictions.

We are currently analysing in detail the form and evolution of the referring expressions used by the subjects with the aim to provide a more qualitative analysis of the differences between the two interactivity settings. In the future we also plan to experiment with other tasks in order to determine to what extent the consequences of reducing positive feedback are dependent on the task to be carried out.

Acknowledgements. We are thankful to J. Dreyer for letting us use the lab at ZAS, to M. Stede and A. Corradini for early discussions of the setting, and to three anonymous reviewers for helpful comments. This work was supported by the EU (Marie Curie Programme) and DFG (Emmy Noether Programme).

References

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

J. Caspers. 2003. Local speech melody as a limiting factor in the turn-taking system in dutch. *Journal of Phonetics*, 31:251–276.

H. Clark and S. Brennan. 1991. Grounding in communication. In *Perspectives on Socially Shared Cognition*, chapter 7, pages 127–149. APA Books, Washington.

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, T. Lucht, K. Rodríguez, and D. Schlangen. 2006. Interaction in task-oriented human-human dialogue: The effects of different turn-taking policies. In *Proceedings of the first International IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba.

H. P. Grice. 1975. Logic and converstion. In *Syntax and semantics, Volume 3: Speech acts*, pages 225–242. Seminar Press, New York.

R. Krauss and P. Bricker. 1967. Effects of transmission delay and access delay on the efficiency of verbal communication. *Jounrnal of the aCoustic Society of America*, 41:286–292.

R. Krauss and S. Weinheimer. 1964. Changes in reference phrases as a function of frequency of usage in social interaction: A preliminary study. *Psychonomic Science*, 1:266–278.

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.

M. F. McTear. 2004. *Spoken Dialogue Technology.* Springer Verlag, London, Berlin.

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.*

H. Sacks, E. A. Schegloff, and G. A. Jefferson. 1974. A simplest systematics for the organization of turn-taking in conversation. *Language*, 50:735–996.

S. Whittaker. 2003. Theories and methods in mediated communication. In *The Handbook of Discourse Processes*, pages 243–286. Lawrence Erlbaum Associates.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 33–40.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Incorporating Asymmetric and Asynchronous Evidence of Understanding in a Grounding Model

Alexandre Denis, Guillaume Pitel, Matthieu Quignard, Patrick Blackburn
TALARIS team
UMR 7503 LORIA/INRIA Lorraine
Campus scientifique, BP 239
F – 54506 Vandoeuvre-lès-Nancy cedex
alexandre.denis@loria.fr

Abstract

The grounding process relies on the evidence that speakers give about their understanding (Clark and Schaefer, 1989). However in existing formal models of grounding (Cahn, 1992; Cahn and Brennan, 1999; Traum, 1999) evidence of understanding is assumed to be *symmetrically* and *synchronously* shared by the speakers. We propose a formal model, based on (Cahn, 1992), that removes these simplifications; we do so by distinguishing the phase of interpretation from the phase of evidence extraction and introducing the notion of floating contributions.

1 Introduction

A dialogue is a process that presupposes the collaboration of both participants. Each speaker in turn assumes that the other will show evidence of understanding or misunderstanding of her utterance, and evidence indicating its relevance to the previous utterance. This mutual assumption is the basis of the *grounding process* (Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989), the process by which speakers try to reach mutual understanding. Successful grounding does not guarantee mutual understanding though: it can happen that the grounding evidence leads two speakers to believe that they have achieved perfect understanding, whereas in reality they have understood two completely different things (Cherubini and van der Pol, 2005). But although this shows that successful grounding is not a *sufficient* condition for achieving mutual understanding, it does seem to be a *necessary* one.

Different models of the grounding process define when (and, sometimes, how) an utterance is added to the common ground (a representation of what is believed to have been mutually accepted). In the *Contribution Model* (Clark and Schaefer, 1989) the grounding process results in a recursively structured directed acyclic graph representation of the dialogue grounding structure, the basic unit of which is the *contribution*. Contributions are twofold units consisting of: (1) an utterance called the *presentation* (or *Pr*) and (2) an *acceptance* linked to a sequence of contributions or a single utterance. The acceptance (or *Ac*) contains the negotiation of the understanding of the presentation in order to reach the *grounding criterion*. The grounding criterion is a threshold defined by Clark and Wilkes-Gibbs (1986) to represent the level of understanding required by the contributor; we shall use the expression *grounding status* to mean the current state of the believed mutual understanding of an utterance. When the grounding criterion holds for a contribution, that is, when its status is grounded, both speakers consider it closed and can choose whether or not to integrate its semantic content as a mutual belief. The grounding status is established via simple evidence of understanding and relevance. The Contribution Model was the pioneering approach to the modeling of grounding and its insights influenced the subsequent development of formal models intended for computational applications.

Probably the best known of these subsequent models is the *Grounding Acts Model* (Traum and Allen, 1992; Traum, 1994; Traum, 1999). This model is based on the notion of *grounding acts*, low level communicative acts whose goal is to ground content at the utterance level. The basic unit of analysis provided by the Grounding Acts Model is a non-recursive sequence of utterances called a *Discourse Unit* (DU). The grounding process is modelled by an update of the state of a Discourse Unit by a grounding act; this makes the approach particularly suitable for integration into information-state based models of dialogue (such as Matheson et al. (2000)); transitions between states are modelled in (Traum and Allen, 1992) and many subsequent papers using finite state automata triggered by the various grounding acts. For example, a *RequestRepair* act by participant A would send a Discourse Unit into a state where a *Repair* by participant B and a subsequent *Acknowledge* by A would be needed to ground it.

The Grounding Acts Model makes the assumption that the grounding level can be distinguished from the intentional level. However, as was noted by (Stirling et al., 2000), it is often not easy to delineate DUs, which makes it difficult to clearly distinguish the grounding level from deeper levels of understanding that emerge via complex exchanges. Hence, as our primary motivation is to explore ways of uniformly integrating grounding at the utterance level with complex negotiations of understanding, we have not taken the Grounding Acts Model as our point of departure.

Instead we have chosen to develop the *Exchange Model* approach presented in (Cahn, 1992; Cahn and Brennan, 1999), which are more directly based on the original Contribution Model. The central innovation provided by Exchange Models is a level of *exchange* that is higher than the level of contributions (this central notion is very much in the spirit of the implicit adjacency pairs used in Clark and Schaefer (1989)). Like work based on the Grounding Acts Model, these Exchange Models have a formal definition and provide online models of grounding. What makes them particularly useful for our purposes, however, is that they follow the Contribution Model in producing graph-like representations of the dialogue grounding structure; in our view, this makes them particularly well-suited for modeling more complex negotiations of understanding.

Nonetheless, different as these three types of model are, they share a common deficiency: *they cannot deal with wrongly recognized or unrecognized grounding acts or evidence of understanding*. In the original Exchange Models, the evidence is always assumed to be *symmetric* and *synchronous*—that is *correctly* and *immediately* understood by the hearer. In the Grounding Acts Model, matters are a little more subtle. There an unrecognized grounding act would initiate a new Discourse Unit, and hence the model might be said to handle asymmetric grounding. Nonetheless, it is not obvious how, once grounded, this Discourse Unit should be reintegrated, nor how the effects of the newly understood grounding act could be taken into account with respect to previous Discourse Units. It may be the case that the Grounding Acts Model and related information states approaches (such as Matheson et al. (2000; Larsson and Traum (2000)) could be extended to handle this kind of reintegration, perhaps by providing additional update rules. But we have found that the (recursive) graph-like representations used by Exchange Models provides a particularly perspicuous setting for a preliminary explorations of the issues involved.

Accordingly, we shall proceed as follows. In Section 2 we discuss existing Exchange Models and their shortcomings in more detail. In Section 3, we present an augmented Exchange Model, inspired by (Cahn, 1992), which repairs these deficiencies. In Section 4, with the help of an example, we show in detail how the model works. Section 5 concludes.

2 A closer look at Exchange Models

The Exchange Models proposed in (Cahn, 1992; Cahn and Brennan, 1999) are intended to formalize the Contribution Model to enable it to be embedded in dialogue systems. Like the Contribution Model, they are based on (recursive) graph-like structures, but they add a level of *exchange* above the Contribution/Presentation/Acceptance levels present in the Contribution Model. An exchange is a pair of contributions defined relative to a task: the first contribution proposes a task while the second contribution executes the task. The grounding process itself is modeled by a decision table based on two features: (1) the evidence of understanding manifested in an utterance and (2) the role of the current utterance in an exchange (i.e. in a dialogue task). In these models, the grounded dialogue structure is represented from each speakers' individual point of view, and *"all contribution graphs are private models, and can represent the perspective of only one agent"* (Cahn and Brennan, 1999).

The model defined in (Cahn, 1992) (henceforth *EM92*) uses three categories of evidence: UNDERSTOODRELEVANT, NOTUNDERSTOOD and

u_1: Where does Dan work ?
s_2: In the natural language group
u_3: No, I meant his office

Figure 1: UNDERSTOODNOTRELEVANT evidence example (dialogue 6.3.3)

UNDERSTOODNOTRELEVANT. However it considers the integration of the utterances of the speaker based on the evidence she *intended* to produce, and does not take into account *how the hearer actually interprets this evidence*. This model can thus only render the perspective of the speaker. The model defined in (Cahn and Brennan, 1999) (henceforth *EM99*) is limited to the system's point of view, unlike *EM92* which is participant-agnostic. It is based on two categories of evidence: ACCEPTABLE and NOTACCEPTABLE for the user. This is not enough to cover all cases; for example, it is not possible for the system to warn that its utterance was misunderstood (though the user can do this). The main improvement of *EM99* over *EM92* is that it characterizes grounding from the hearer's point of view (in this case the system) and integrates the utterances of the user as well as those of the system.

So what is wrong with these models? An example should make matters clear. Consider Figure 1, which shows an example dialogue from (Cahn, 1992). When S utters s_2, she believes her utterance is a relevant answer to u_1. That is, she appends s_2 as the second contribution of the exchange initiated by u_1 (see Figure 2).[1]

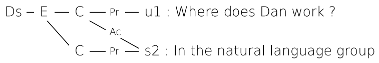

Ds – E — C — Pr — u1 : Where does Dan work ?
Ac
C — Pr — s2 : In the natural language group

Figure 2: Dialogue 6.3.3 after s_2

When S receives u_3 and extracts the UNDERSTOODNOTRELEVANT evidence it conveys, she has to restructure her view of the dialogue to take into account the fact that s_2 is *not* relevant with respect to u_1. That is, she must move s_2 into a new exchange in the acceptance phase of u_1 and append u_3 as the second contribution of this exchange (see Figure 3).

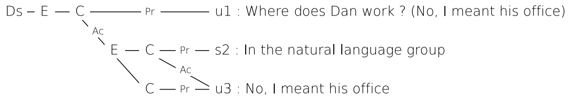

Ds – E — C — Pr — u1 : Where does Dan work ? (No, I meant his office)
Ac
E — C — Pr — s2 : In the natural language group
Ac
C — Pr — u3 : No, I meant his office

Figure 3: Dialogue 6.3.3 after u_3

But now take the point of view of U on the situation: she interprets s_2 as not relevant with respect to u_1 *as soon as she receives it*, and thus does *not* have to revise her model. Instead she integrates s_2 directly into the acceptance phase of her own utterance u_1. To properly model such situations from both points of view, we need to consider not merely the intended evidence of understanding, but also the actual interpretation of evidence.

However *EM92* cannot handle this behavior because it does not consider the interpretation of the hearer: it is focused on how integrate an utterance according to the evidence of understanding the speaker intended it to convey, and not on how the hearer actually interpreted the utterance. The model *EM99*, on the other hand, partially takes into account the hearer's interpretation in one case, namely when the user gives ACCEPTABLE evidence and does not propose a task. For example, consider the exchange shown in Figure 4. *EM99* handles such examples well. If the system believes it does not understand u_2, its next utterance would initiate a new exchange in the acceptance phase of u_2. On the other hand, if it believes it understands u_2, its next utterance would initiate a new exchange at the dialogue level.

s_1: Where's Dan ?
u_2: In his office (+ noise)

Figure 4: ACCEPTABLE evidence example

This is a step in the right direction, but it does not go far enough: the interpretation of the hearer needs to be taken into account in *all* cases, for *both* speakers, *whatever* the evidence conveyed by the interpreted utterance. Only if the hearer understands can she extract the evidence of understanding. If the interpretation of the hearer is not taken into account, it is impossible to consider the non-understanding (or the misunderstanding) of the evidence of understanding. As a consequence, models that make this simplification can only deal with *symmetric* and *synchronous* evi-

[1] In the paper we sometimes use the term utterance instead of contribution, or an utterance symbol to denote a contribution; in these case we always mean "the contribution presented by this utterance".

u_1: Where's Dan ?
s_2: Dan Smith or Dan Jones ? (+ noise)
u_3: Uh, what did you say ?

Figure 5: Not understanding a NOTUNDERSTOOD evidence of understanding

dence of understanding. Symmetric means that the presented evidence of understanding is always understood by the hearer as expected by the speaker. Synchronous means that it is also understood as soon as the utterance is emitted. Neither *EM92* nor *EM99* can handle the fact that the *acceptance function* of an utterance is not always played at the same moment for the two speakers. In the example in Figure 5, the user does not know that the utterance s_2 manifests NOTUNDERSTOOD evidence because she herself has not understood s_2 sufficiently to extract the evidence of understanding.

3 An extended Exchange Model

The aim of our work is to specify a grounding model that handles asymmetric and asynchronous evidence of understanding. We do so in a way that retains the advantages of both *EM92* and *EM99*, and follow these models in using a small collection of sharp understanding categories. We found *EM92*, which does not distinguish the user from the system and has better categories of understanding, to be a better starting point, and thus have reworked the key insights of *EM99* in the setting of *EM92*. We introduce asymmetry and asynchronicity into this model by distinguishing two steps in the interpretative process: the understanding of an utterance, and the extraction of the evidence of understanding it conveys. Our model is defined from the point of view of the hearer (who we will refer to by *self*). It considers how self understands an utterance *before* taking into account the evidence of understanding it shows. If the utterance is UNDERSTOODRELEVANT, the evidence it shows is extracted and is integrated as in *EM99*. If the utterance is UNDERSTOODNOTRELEVANT, it is integrated as initiating a new exchange under the acceptance phase of the previous contribution without considering the evidence of understanding it shows. If the utterance is NOTUNDERSTOOD, the contribution is introduced as *floating*, waiting for later integration in the graph.[2] Its integration into the main dialogue structure is possible only when its acceptance phase shows what the evidence of understanding was. Then the utterance has to be reinterpreted, taking into account the newly understood evidence. However the grounding status of floating items could remain pending and never be solved, for example if the acceptance phase is abandoned.

Do floating contributions have an analog in the Grounding Acts Model? We don't believe so. One could try comparing the collection of ungrounded states of a Discourse Unit to the acceptance phase of a contribution. That is, the open state of the acceptance phase of a contribution in an Exchange Model could be regarded as the analog of the ungrounded states of a Discourse Unit in the Grounding Acts Model. But the notion of a floating contribution is stronger than the notion of ungrounded states: when a contribution is floating, it means not only that it is not grounded yet, but also that the evidence it manifests is not known either. Furthermore, floating status isn't correctly captured by the ungroundable state used in the Grounding Acts Model either. The ungroundable state is a terminal state, reached by canceling the grounding process. However the floating status that some contributions acquire in our approach is intended to be temporary—if the evidence conveyed by the contribution comes to be understood, its floating status is cancelled and the contribution is integrated into the main dialogue structure.

So: how can we augment the Exchange Model to handle asymmetric and asynchronous evidence of understanding? The main additions we shall make are the following. First, in order to keep track of the floating contributions, we have to maintain another structure which contains the sequence of pending contributions. Second, we have to handle reinterpretation and specify how the newly acquired evidence of understanding is used to integrate a floating contribution. Third, an utterance can now give evidence of understanding concerning many previous utterances, because an utterance which closes an acceptance phase and gives evidence as such can now reveal how to interpret the accepted contribution too.

A formalization[3] is presented in Table 1 and Ta-

[2] In this version, NOTUNDERSTOOD means that the evidence is not understood either. Multiple degrees of understanding are possible though (Brennan and Hulteen, 1995).

[3] The tables are simplified. We do not discuss here dialogue beginning and ending nor the necessity of having a three-fold context $\langle S_i, O_j, S_k \rangle$ to manage reinterpretation. The actual implementation also deals with evaluation utter-

Self interpretation of O_i	*Integration of O_i*	*Integration of S_{i+1}*
UNDERSTOODRELEVANT w/r S_{i-1}	integrate O_i according to the evidences of understanding it shows: if O_i shows evidences about another utterance S_j, call Table 2 with S_j else call it with S_{i-1}	integrate S_{i+1} after O_i
UNDERSTOODNOTRELEVANT w/r S_{i-1}	integrate O_i as initiating a new exchange in the acceptance phase of S_{i-1}	integrate S_{i+1} after O_i
NOTUNDERSTOOD	O_i presents a fbating contribution, waiting to be understood to be integrated	integrate S_{i+1} as initiating a new exchange in the acceptance phase of O_i

Table 1: Integration of both utterances (O_i, S_{i+1})

ble 2. It covers the aforementioned cases with delays in the integration of utterances. We use the following notation: O_i stands for the utterance produced by the *other* speaker (O) at time i, and S_{i+1} stands for the utterance produced by *self* (S) at time $i + 1$. An utterance is said to initiate an exchange if it is the presentation of the first contribution of this exchange. An exchange is said to be open if its first contribution is set while its second contribution is not. The main dialogue structure is called D and the floating structure F. Finally, "integrate u_i after u_j" is a shorthand for:

- if u_j initiated an exchange, append u_i as the second contribution of this exchange;
- else append u_i as the second contribution of the closest upper level open exchange, if there is one;
- else (all exchanges are closed), u_i initiates a new exchange at the dialogue level.
- in all cases u_i closes the acceptance phase of u_j.

Reinterpretation of an utterance consists of calling the algorithm again with a new interpretation and new evidence of understanding. The only difference is that the contribution presented by this utterance does not have to be created because it already exists in the floating structure. If the utterance is eventually understood (relevant or not) it can be moved in the dialogue structure in accordance with its new interpetation, and the new evidence of understanding it shows. This evidence of understanding is consequently acquired asynchronously by the two participants.

The main simplifying assumptions made by our algorithm are the following:

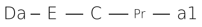
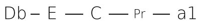
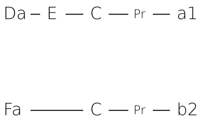
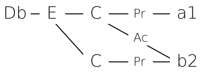
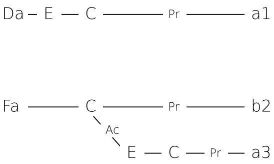
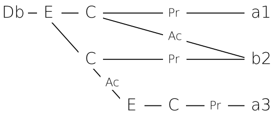
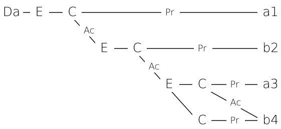
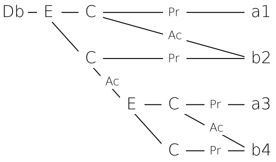
- We suppose a direct correlation between the result of an interpretation of O_i and the evidence of understanding conveyed by S_{i+1}. How S interprets O_i is manifested in the utterance she produces in turn S_{i+1}. That is, if an utterance is not understood or not relevant, one has to clarify the situation. This simplification is based on the collaborative dialogue hypothesis.
- In the version of the algorithm presented above, the evidence of understanding is either understood or not. That is, the asymmetry is binary and there cannot be any misunderstanding of the evidence of understanding. Such misunderstandings would be more complex to handle because of the increased divergence between the participants dialogue representation structures. But systems can be mistaken when extracting the evidence of understanding, and we think it will be necessary for dialogue systems to represent this.
- Contributions always alternate. The present algorithm does not actually manage several contributions in one speech turn because this would mean taking into account *interleaved* evidence of understanding. But, once again, we feel that this extension will be necessary to handle more realistic dialogues.

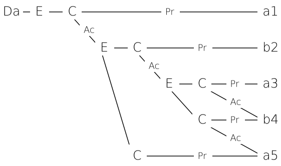
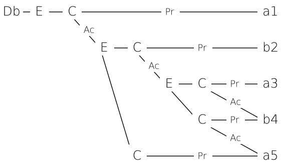
A tool illustrating our model has been implemented in Java. This takes as input a dialogue where each utterance is annotated by the evidence of understanding its speaker believes it to convey. The resulting output is the dialogue structure and the floating structure for both speakers at different steps. The tool was used to generate the diagrams used in this paper, and in particular, the diagrams in the example to which we now turn.

ances and abandons. For a complete description, please refer to `http://www.loria.fr/~denis`

Evidence of understanding of S_j showed in O_i	*S_j did not initiate an exchange or initiated an exchange at the dialogue level*	*S_j initiated an exchange in an acceptance phase*
UNDERSTOODRELEVANT w/r O_{j-1}	integrate O_i after S_j	integrate O_i after S_j, if the accepted contribution is floating, reinterpret its presentation O_k: call Table 1 where $O_i = O_k$
UNDERSTOODNOTRELEVANT w/r O_{j-1}	move the S_j contribution as the first contribution of a new exchange in the acceptance phase of O_{j-1}, integrate O_i after S_j	
NOTUNDERSTOOD	integrate O_i as initiating a new exchange in the acceptance phase of S_j	

Table 2: Integration of O_i when it is understood and thought relevant

Utterance and evidence of understanding it shows	*Point of view of A*	*Point of view of B*
a_1: Where does Dan work ?		
b_2: In the natural language group UNDERSTOODRELEVANT(a_1)		
a_3: What did you say ? NOTUNDERSTOOD(b_2)	 Da – E — C — Pr — a1 Fa — C — Pr — b2 Ac E — C — Pr — a3	 Db – E — C — Pr — a1 Ac C — Pr — b2 Ac E — C — Pr — a3
b_4: I said: in the natural language group UNDERSTOODRELEVANT(a_3)	 Da – E — C — Pr — a1 Ac E — C — Pr — b2 Ac E — C — Pr — a3 Ac C — Pr — b4	 Db – E — C — Pr — a1 Ac C — Pr — b2 Ac E — C — Pr — a3 Ac C — Pr — b4

a_5: No, I meant his office
UNDERSTOODRELEVANT(b_4)
UNDERSTOODNOTRELEVANT(b_2)
	 Da – E — C — Pr — a1
	Ac
	E — C — Pr — b2
	Ac
	E — C — Pr — a3
	Ac
	C — Pr — b4
	Ac
	C — Pr — a5
		 Db – E — C — Pr — a1
		Ac
		E — C — Pr — b2
		Ac
		E — C — Pr — a3
		Ac
		C — Pr — b4
		Ac
		C — Pr — a5

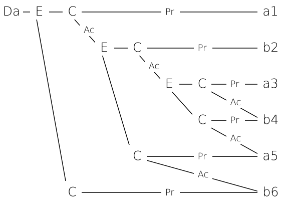
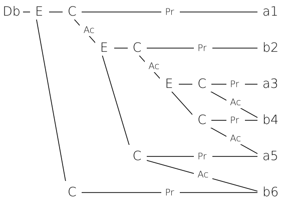
b_6: Near post H33 UNDERSTOODRELEVANT(a_5)	 Da – E — C — Pr — a1 Ac E — C — Pr — b2 Ac E — C — Pr — a3 Ac C — Pr — b4 Ac C — Pr — a5 Ac C — Pr — b6	 Db – E — C — Pr — a1 Ac E — C — Pr — b2 Ac E — C — Pr — a3 Ac C — Pr — b4 Ac C — Pr — a5 Ac C — Pr — b6

Table 3: Detailed example

4 Detailed example

The example in Table 3 is a modification of the example 6.3.3 in (Cahn, 1992), in which the second utterance is not understood by the user (called A whereas the other participant, the system, is called B). This dialogue illustrates the asymmetry and asynchronicity of the learning of the evidence of understanding showed by b_2. The left column presents the utterances and the evidence of understanding showed by them from their speaker's point of view. The two other columns present the dialogue structure according to each point of view.

The first utterance a_1 is believed UNDERSTOODRELEVANT[4] by B and is integrated normally as initiating an exchange at the dialogue level. The dialogue viewed by A is the same.

The second utterance b_2 shows a divergence. B believes that b_2 presents an UNDERSTOODRELEVANT evidence of understanding and thus integrates it as the second contribution of the first exchange. However this evidence is not shared by A, who does not understand b_2 and therefore cannot integrate it. She just keeps the contribution floating, awaiting to be integrated when it is sufficiently understood (see F_a in Table 3).

Utterance a_3 shows that b_2 was NOTUNDERSTOOD by A and that she requires clarification. Because a_3 is understood by B, the evidence of understanding it contains is used to integrate it as the initiator of a new exchange under the acceptance phase of b_2 contribution.

Utterance b_4 shows that a_3 was interpreted UNDERSTOODRELEVANT by B. Therefore it is integrated by both speakers as the second contribution of the clarification exchange. However there is a new divergence when processing the utterance b_4. For B, b_4 is only an answer to the clarification request. But with b_4, A can now interpret b_2. As b_2 is now understood by A, she can extract the evidence of understanding it showed, and act according to her own interpretation. In this case, because b_2 is UNDERSTOODNOTRELEVANT by A, she won't take into account the evidence of understanding showed by b_2. The acquisition of the evidence of understanding showed by b_2 is asynchronous but not taken into account; see Table 1. The reinterpretation of b_2, according to the UNDERSTOODNOTRELEVANT rule, leads to the b_2 contribution being embedded as initiating a new exchange under the acceptance phase of a_1.

The utterance a_5 is crucial for reaching the grounding criterion. It makes available two pieces of evidence of understanding: first it shows that b_4 is an UNDERSTOODRELEVANT reply to a_3 and second it shows that b_2 is UNDERSTOODNOTRELEVANT with respect to a_1. Its effect is to revise the B view on the dialogue to create a new exchange in the acceptance phase of a_1. Doing this means that the structures of the grounding model converge for both speakers; they now agree on the current view of dialogue.

The utterance b_6 is the final answer to the first question. It shows that a_5 was UNDERSTOODRELEVANT by B and UNDERSTOODRELEVANT by A. It is integrated as a relevant reply to the first contribution of the upper level exchange.

5 Discussion and further work

This paper discusses the problems posed by asymmetric and asynchronous evidence of understanding, and gives a preliminary model of how such evidence could be handled. It does so by distinguishing the phase of interpretation from the phase of evidence extraction and introducing the notion of floating contributions into the Exchange Model. Such contributions cannot be immediately attached to the dialogue structure because the evidence of understanding they show is not known. When these contributions are accepted, they have to be reinterpreted in order to extract the evidence of understanding they manifest.

A side effect of our model is that it provides a novel solution to the recursive acceptance problem defined in (Traum, 1994; Traum, 1999): if an acceptance utterance needs to be accepted *before* it can play its acceptance function, then no contribution would ever be complete. To solve the problem, we make the assumption that a participant may form the belief that she has understood (or not) an utterance as soon as she receives it; she does *not* have to subordinate her belief to further acceptance (we believe that this assumption can be motivated by the ideas on timing in joint actions in Chapter 3 of (Clark, 1996)). The acceptance function of an utterance can be played, from the hearer's point of view, *as soon as she understands the utterance*. On the other hand, to check whether what she said successfully played its intended acceptance role, the speaker of the utterance has to wait for the hearer's response. However, *as soon*

[4] The first utterance is assumed relevant when it is understood

as the hearer responds, the appropriate acceptance function may be played. But when misunderstanding occurs, the acceptance role of an utterance is delayed up to the moment it is sufficiently understood to be integrated into the common ground.

The implemented model we have presented still suffers from a number of limitations; for example it does not deal with misunderstanding of the evidence of understanding. Planned future work will cover these more complex divergences in dialogue structure in addition to multi-contributions, that is, when several contributions by the same speaker in the same turn. We hope that this model and its implementation will be the first stage of a larger enterprise: specifying the grounding status of the contents of a contribution in terms of dialogue structure.

Acknowledgements

We would like to thank TALARIS members Luciana Benotti and Daniel Coulon for feedback and discussion during the writing of this paper.

References

Susan E. Brennan and Eric A. Hulteen. 1995. Interaction and feedback in a spoken language system: a theoretical framework. *Knowledge-Based Systems*, 8:143–151.

Janet E. Cahn and Susan E. Brennan. 1999. A psychological model of grounding and repair in dialog. In *AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, pages 25–33.

Janet E. Cahn. 1992. A computational architecture for the progression of mutual understanding in dialog. Technical Report 92-4, Music and Cognition Group M.I.T. Media Laboratory.

Mauro Cherubini and Jakko van der Pol. 2005. Grounding is not shared understanding: Distinguishing grounding at an utterance and knowledge level. In *CONTEXT'05, the Fifth International and Interdisciplinary Conference on Modeling and Using Context*, Paris, France, July 5-8.

Herbert H. Clark and Edward F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294.

Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as collaborative process. *Cognition*, 22:1–39.

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

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural Language Engineering*, 6(3-4):323–340.

Colin Matheson, Massimo Poesio, and David Traum. 2000. Modelling grounding and discourse obligations using update rules. In *Proceedings of NAACL 2000*, May.

Lesley Stirling, Ilana Mushin, Janet Fletcher, and Roger Wales. 2000. The nature of common ground units: an empirical analysis using map task dialogues. In *Gotalog 2000, 4th Workshop on the Semantics and Pragmatics of Dialogue*, pages 159–165, Gothenburg, Sweden.

David R. Traum and James F. Allen. 1992. A ”speech acts” approach to grounding in conversation. In *International Conference on Spoken Language Processing*, pages 137–40.

David R. Traum. 1994. *A Computational Theory of Grounding in Natural Language Conversation*. Ph.D. thesis, University of Rochester, Rochester, New York.

David R. Traum. 1999. Computational models of grounding in collaborative systems. In *working notes of AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, pages 124–131.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 41–48.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Towards Modelling and Using Common Ground in Tutorial Dialogue[*]

Mark Buckley and **Magdalena Wolska**
Department of Computational Linguistics
Saarland University
66041 Saarbrücken, Germany
{buckley|magda}@coli.uni-sb.de

Abstract

In order to avoid miscommunication participants in dialogue continuously attempt to align their mutual knowledge (the "common ground"). A setting that is perhaps most prone to misalignment is tutoring. We propose a model of common ground in tutoring dialogues which explicitly models the truth and falsity of domain level contributions and show how it can be used to detect and repair students' false conjectures and facilitate student modelling.

1 Motivation

In order to communicate efficiently, participants in a dialogue take into account the information believed to be mutually known to them: the "common ground" (Clark and Schaefer, 1989). This concerns not only knowledge accumulated in the course of dialogue, but also common ground (context) that is presupposed prior to the interaction. In order for a piece of information to become common ground (henceforth CG), it must be explicitly or implicitly acknowledged by the interlocutor in the process called *grounding* (Clark and Schaefer, 1989; Traum, 1994). Lack of grounding may lead to incorrect beliefs about CG or, using Stalnaker's term, "defective context" (Stalnaker, 2002) i.e. a situation in which discourse participants presuppose different things. This, in turn, may lead to miscommunication.

A setting that is perhaps most prone to misalignment in discourse participants' beliefs is tutoring. Student-tutor teaching interactions are characterised by an inherent asymmetry of knowledge possessed by the tutor and the learner (Munger, 1996; Lee and Sherin, 2004). In order to effectively guide the student in a problem-solving task, the tutor must make inferences as to the student's state of knowledge based on his/her interpretation of student's utterances. Empirical research shows, however, that untrained tutors tend to perform specific targeted moves (e.g. use curriculum scripts, example problems, give immediate evaluative feedback) that locally address the student's progress (or lack thereof) on the task at hand, instead of focusing mainly on coordinating the CG, i.e. establishing complete understanding of the students' state of beliefs, and so cognitive alignment (Graesser et al., 1995). Considering this, it is difficult for tutors to identify, let alone repair, deep misconceptions that underlie students' errors. However, when a misalignment in student's beliefs as to CG becomes apparent based on the linguistic content of the student's utterance, the tutor may choose to explicitly address it by challenging the student's statement. Moreover, an explicit model of CG can feed into a module that monitors the student's performance (McArthur et al., 1990; Merrill et al., 1995).

In this paper, we propose a preliminary model of CG in tutoring dialogues on problem solving, in the context of building a conversational intelligent tutoring system for mathematical proofs. As the dialogue progresses the CG in our model develops as a store of the truth and falsity of the contributions that the student has made, based on the evaluations that the tutor has given. In addition, discourse referents for formulas are introduced to support modeling their salience. The CG forms the context for judging whether an utterance constitutes evidence that misalignment has occurred.

We begin by discussing examples of student-tutor interactions that exemplify the use of CG in this domain and motivate the need for modelling CG in an automated system (Section 2). We present the structure of the CG maintained by the

[*] This research has been supported by the Collaborative Research Centre on *Resource-Adaptive Cognitive Processes*, SFB 378.

dialogue model, the mechanism of updating the CG, and discuss the uses of CG in the tutorial dialogue: detecting and repairing student's false conjectures and facilitating student modelling (Section 3). We give a walk-though of our examples in Section 4 and related work is discussed in Section 5.

2 Linguistic Data

To collect data about naturalistic human-computer tutorial dialogue interactions, we conducted two experiments in a Wizard-of-Oz paradigm (Kelley, 1984). The subjects and the wizards were using natural language (German; typed on the keyboard) and mathematical symbols available on the GUI to interact with the simulated system, and they were unconstrained in their language production.

In the first experiment (Wolska et al., 2004) the subjects were tutored using one of the following strategies: *minimal feedback* (students were given feedback only on correctness and completeness of proposed proof-steps), *didactic* (when the student made no progress, the tutor disclosed the expected reasoning step), or *socratic* (a pre-defined hinting algorithm was used to guide the student toward the solution). In the second experiment (Benzmüller et al., 2006) the tutors did not follow any tutoring algorithm, but were given general instructions on *socratic* tutoring. The first experiment concerned naive set theory and the second binary relations. In both, students were given study material before the tutoring session containing the basic mathematical knowledge required to solve the exercises. Overall, the collected corpora consist of 22 and 37 dialogue logfiles for the first and second experiment respectively.

2.1 Dialogue Phenomena

The CG that student and tutor believe currently exists in the dialogue can become misaligned as the dialogue progresses, for instance due to misunderstanding, misconception, or the boundedness of attentional resources. Evidence of misalignment of CG can be observed, for example, in certain situations in which informationally redundant utterances (IRUs) (Karagjosova, 2003; Walker, 1993) are performed. An utterance is informationally redundant if the proposition it expresses is entailed, presupposed or implicated by a previous utterance in the discourse.

If an unmarked IRU is performed then the information it contains, which has already been grounded, is being repeated without the speaker indicating that this repetition is being done on purpose. This indicates to the hearer that this information was not in what the speaker believes to be the CG of the dialogue. The hearer must then conclude that the CG has become misaligned.

Sometimes it is necessary to repeat information that has already been grounded, for example to make a known fact salient again to support an argument (Walker and Rambow, 1994). Such utterances are informationally redundant. To prevent the hearer concluding from such utterances that misalignment has occurred, the speaker explicitly indicates that he is aware that he is repeating shared information (i.e. that the fact should be CG) by marking with phrases such as "of course". In tutorial dialogue hints which remind students of previously proved facts and concepts from the tutoring domain are IRUs. Students also use IRUs to check if what they believe the CG to be is actually that which the tutor believes it to be. In the following examples from the corpus we give English translations and include the original German utterances where they are illustrative.

In (1) the domain content of utterance S10, that the assertion that the formula embedded in the utterance holds, is repeated in utterance S18.

(1) **S10:** It holds that $(R \cup S) \circ T = \{(x, y) | \exists z (z \in M \wedge (x, z) \in (R \cup S) \wedge (z, y) \in T\}$

T10: That's right!

. . .

S18: By definition it holds that $(R \cup S) \circ T = \{(x, y) | \exists z (z \in M \wedge (x, z) \in (R \cup S) \wedge (z, y) \in T\}$

T18: That's right! You've already performed this step.
(*German: Diesen Schritt haben Sie vorhin schon vollzogen.*)

The confirmation (T10) of this fact puts the truth of the formula in S10 in CG, and therefore when utterance S18 is performed it is an IRU. Because S18 is unmarked for informational redundancy, the tutor concludes that misalignment of context has occurred, i.e. the fact concluded in S10 is no longer CG for the student. He augments his confirmation (T18) with the indication ("already") that the step in S18 had already been performed, telling the student that misalignment occurred and realigning their CG.

In example (2) the student explicitly introduces by assumption a fact that he has already proved.

(2) **S3:** Let $(a,b) \in (R \circ S)^{-1}$. Then it holds that $(b,a) \in (R \circ S)$

T3: That's right.

...

S6: Let $(b,a) \in (R \circ S)$. Then ...

T6: Since you already know that $(b,a) \in (R \circ S)$, you don't need to postulate it again.

In S3 the student has successfully proved $(b,a) \in (R \circ S)$, and the tutor's confirmation of this (T3) makes it CG. The student later wants to use this fact as the antecedent of a derivation (S6), but wrongly introduces it as a new assumption. Thios shows that the truth of this formula is no longer CG for the student, i.e. misalignment has taken place. In order to realign the CG the tutor reminds the student that he has previously proved the formula, and this utterance is marked with "already" (T6).

In example (3) an IRU performed by the tutor is marked so that the student does not mistakenly conclude that misalignment has taken place.

(3) **S2:** $A \cap B = \emptyset$

...

T4: Right. Now what?

...

T8: ...The justification could for instance be: Let x be an arbitrary element of B , then it can't be in A (since of course $A \cap B = \emptyset$) ... (*German: ... (da ja $A \cap B = \emptyset$) ...*)

The student has proved a formula in S2 which was confirmed in T4, making it CG. In T8 the tutor recaps the solution proof. The formula $A \cap B = \emptyset$ is part of the proof, and is thus in the CG, so the tutor marks the reminder of this fact with the particle "of course".

An example of a student's marked IRU is shown in utterance S4 of (4), in which the IRU is used by the student to check suspected misalignment. "Doch" is a modal particle which, when deaccented, marks old or shared information.

(4) **S3:** and for the powerset it holds that: $P(C \cup (A \cap B)) = P(C) \cup P(A \cap B)$

T4: Do you really mean: $P(C \cup (A \cap B)) = P(C) \cup P(A \cap B)$?

S4: But I think: $P(A) \cup P(B) = P(A \cup B)$ (*German: ich denke doch: $P(A) \cup \ldots$*)

T5: That's not right! Maybe you should have another look in your study material.

S5: sorry, it holds of course that: $P(C \cup (A \cap B)) \subseteq P(C) \cup P(A \cap B)$

T6: Really?

S6: oh, no.. . the other way around

T7: That's right at last!

In S3 the student claims a fact which is then questioned by the tutor. This causes the student to suspect a misalignment, because a rule he used in deriving the fact and which he believed to be true is in fact false. In S4 the student then checks whether this rule is in fact in the CG by stating it explicitly. He considers S4 to be uninformative, and therefore marks it explicitly with "doch" (meaning "but I thought..."). However S4 actually is informative, in the sense that it is not subsumed by the set of facts in the CG when it is uttered. This leads the tutor to conclude that misalignment has taken place. In addition to rejecting the rule, he also directs the student to the study material. The next student proof step (S5) is again rejected (T6). In S6 the student gets the rule right, which is confirmed in T7. The student adds the corrected rule to his CG, completing the realignment that began in S4.

The data shows that misalignment occurs between student and tutor, and that it can be observed in the case of informationally redundant utterances. Unmarked IRUs (such as S18 in example (1) and S6 in example (2)) are evidence that CG has become misaligned, and should trigger strategies for realignment. Conversely, when IRUs are to be generated as part of pedagogical strategies (T8 in example (3)), these should be marked as such in order to avoid the student falsely concluding that misalignment has occurred. Finally, misalignment can be evidenced by utterances which are marked for informational redundancy but are in fact not IRUs (S4 in example (4)). To account for such phenomena a model of CG is necessary that allows the detection of which utterances are informationally redundant and which are not, at the level of truth in the domain. The CG must therefore model the utterances that were performed and whether their content was accepted by the tutor, and thus grounded.

3 Modelling Common Ground

Our model is developed within the wider scope of a tutorial environment for mathematics. The student's task is to build a proof of a mathematical theorem. The student does this by conducting a dialogue with the tutor in which he/she performs utterances which may contain proof steps. The environment includes study material for the theory at hand. Turn-taking during tutoring sessions is strictly controlled. Each correct proof step extends the current partial proof. The task is completed when the student has constructed a complete proof of the theorem.

3.1 Elements of the Architecture

We now briefly describe the roles played by those system modules in the environment which are directly relevant to analysing CG.[1]

Discourse Interpreter A discourse interpretation module analyses students' natural language utterances.[2] The result of the analysis is the linguistic meaning of the utterance, the dialogue move that represents its function and, in case of domain contributions, a formal representation p for the proof manager (realised as, for example, in (Wolska and Kruijff-Korbayová, 2004)). In particular, the linguistic meaning of modal particles such as "doch" is reflected in the representation in that a feature MARKED is present.

Proof Manager A proof manager maintains the solution the student is building and evaluates proof steps in this context (Dietrich and Buckley, 2007). It can check the correctness and relevance of proof steps by accessing a domain reasoner such as ΩMEGA (Siekmann et al., 2006) or Scunak (Brown, 2006).

Tutorial Manager A tutorial manager stores pedagogical expertise on when and how hints should be given and maintains a student model (Tsovaltzi et al., 2004). The tutorial manager computes what dialogue moves the system should perform. Two possible dialogue moves which are relevant for this model are `accept` and `reject`. It performs the content selection step for output generation, which includes deciding whether utterances which are informationally redundant should be marked as such. It also decides whether to realise moves in the declarative or interrogative mood, as in utterance T4 in example (4).

3.2 Our Model

We model CG as being similar to that of DeVault and Stone (2006). Their *objective normative context* is a product of the actions taken by the dialogue participants. In our case, actions in the dialogue result in the dialogue participants having beliefs about the truth (or falsity) of the propositions that are contributed by the student and evaluated by the tutor. This is combined with the knowledge in the study material that the students are given before the tutorial session. We assume that it is part of the CG at the start of the dialogue. In our model the CG contains the facts that propositions were uttered, the evaluations of those utterances by the tutor, and the facts that the student knows about the domain as a result of preparatory study.

3.2.1 Types of Entities in the Model

There are two types of entities in the model: discourse referents (for entities introduced in the discourse) and propositions.

Domain contributions contain or refer to formulas that the student uses or concludes. For each domain contribution the discourse interpreter delivers the discourse referent for the proposition that the utterance expresses. Our model includes these discourse referents in the common ground. When references are made to substructures of formulas, for instance "the left-hand side of ..." we add new referents as needed.[3]

The fact that a proposition was uttered is modeled as $\texttt{uttered}(speaker, p)$, where speaker is the dialogue participant who performed the utterance. Having $\texttt{uttered}(speaker, p)$ in the CG tells us only that the event took place, and does not tell us anything about the truth of the content P of the utterance. Evaluations of the propositions that were uttered have the form either $\texttt{holds}(p)$ or $\neg\texttt{holds}(p)$, depending on whether they were accepted or rejected by the tutor. For previous knowledge that the student is assumed to have we use $\texttt{prev}(p)$.

Finally we model the utterances which are performed in the dialogue as objects u, and access the proposition p expressed by an utterance u with $p = \texttt{expresses}(u)$. In this way we can access the proof step that an utterance contained. The propositions expressed by utterances are treated as verbatim formulas.

The entities described above are represented in the dialogue model as shown in Figure 1, where

[1] We omit a discussion of other modules which are part of the architecture.

[2] For the purposes of this exposition we only consider assertion-type dialogue moves which contain domain contributions (here labelled with `domcon`), that is, possibly underspecified proof steps. For example we do not treat questions or meta-level communication management etc.

[3] This account of discourse referents is intentionally simple — a full account would require for instance referents for actual utterances in order to resolve references like "what I said above".

$$\left[\begin{array}{ll} \textsc{cg} & \left[\begin{array}{ll} \textsc{refs} & \langle i_1 \rangle \\ \textsc{props} & \left\langle \begin{array}{l} \text{prev}\big(P(A)\cup P(B)\subseteq P(A\cup B)\big), \\ \text{uttered}\big(\text{student},p_1\big),\ldots \end{array}\right\rangle \end{array}\right] \\ \textsc{lu} & \left[\begin{array}{ll} \textsc{speaker} & \texttt{student} \\ \textsc{utterance} & u_1\texttt{:``And for powerset it} \\ & \texttt{holds that ...''} \\ \textsc{moves} & \{\text{ASSERT}_{\texttt{domcon}}\} \\ \textsc{entities} & \{i_1\} \\ \textsc{marked} & \texttt{n} \end{array}\right] \end{array}\right]$$

Figure 1: Dialogue model representation

CG is the common ground and LU is the last utterance in the dialogue. CG/REFS contains the discourse referents which have been introduced so far. CG/PROPS contains the three types of propositions that we model, namely `uttered`, $(\neg)$`holds` and `previous`. Both of these slots have the type ordered set which provides functions for membership, push and pop. We use this representation as simple account of salience. In CG/REFS, the most salient discourse entity is the one whose referent is at the top of the stack.

The LU part of the dialogue model stores a representation of the last utterance. It includes which dialogue participant performed the utterance, the actual utterance itself, the set of dialogue moves that the utterance realised as well as the discourse referents that were addressed. The flag `marked` indicates whether the utterance was marked for informational redundancy.

3.2.2 Updating the Common Ground

Information is added to the CG as a result of utterances performed by the student and the tutor. This corresponds to implicit grounding at the understanding level (Clark, 1996).[4]

We model three dialogue moves which lead to an update of the CG: `domcon`, `accept` and `reject`. Domain contributions claim the truth of formulas derived by proof steps, and in our model they correspond to Clark's proposal or presentation phase. In the case of a `domcon` we make the updates `push`$($`uttered`(s,p),CG/PROPS$)$ and `push`$(i$,CG/REFS$)$, where p is the content of the proposition and i is the discourse referent introduced by p.

Predicate	Definition
exists(i)	$i \in$ CG/REFS
exists(p)	holds(p) $\in$ CG/PROPS $\vee$ $\neg$holds(p) $\in$ CG/PROPS
salient(i)	$i =$ top(CG/REFS)
iru(u)	exists(expresses(u))

Table 1: Predicates on common ground.

The tutor's evaluations of domain contributions performed by the student are represented by `accept` and `reject` moves, which lead to the updates `push`$($`holds`(p),CG/PROPS$)$ and `push`$(\neg$`holds`(p),CG/PROPS$)$ respectively. Here we rely on the discourse interpreter being able to determine which domain contribution the tutor is responding to and, in turn, what its propositional content p was. Because they have the effect of putting things in CG, `accept` and `reject` moves correspond to Clark's acceptance phase. We make the assumption that the tutoring scenario gives the tutor the authority to ground content simply by evaluating it. In effect, the student is expected to accept what the tutor says as being true. A further update is made to the CG when a substructure of a formula is accessed. New discourse referents j for subformulas are generated on demand and added to the CG by `push`$(j$,CG/REFS$)$.

3.2.3 Testing and Using the Common Ground

Now that we can update the CG to reflect the current state of mutual belief, we define a set of predicates (see Table 1) that test properties of given propositions and referents. The predicate `exists`(x) holds when the discourse referent or proposition x has already been introduced, and `salient`(i) holds of the most salient discourse referent i. Utterances are IRUs if they satisfy the predicate `iru`(u), that is, if the proposition they express is already in the CG. Since `expresses` treats formulas verbatim, `iru`(u) can only hold when the formula is a case-insensitive match of u. We also define an operation `makesalient`(i) which promotes an existing discourse referent i to the top of CG/REFS, making it the most salient referent. The `makesalient` operation is performed when `iru`(u) is detected because a formula is being mentioned for a second time and should become salient again.

The `exists` predicate allows us to determine whether utterances are informationally redundant

[4] In this simplified account, we assume that the tutor understands what the student says, and that when the tutor tells the student what the evaluation of a step was, that the student understands this without having to acknowledge it, implicitly or otherwise.

in the context of the dialogue, and using `exists` we can now define a test on the dialogue model which tells us whether the last utterance is evidence that the common ground has become misaligned. Informally, we can conclude that misalignment has occurred if an IRU is not linguistically marked ($-$) for informational redundancy or if a non-IRU is marked ($+$) for informational redundancy. In terms of the predicates introduced above, we express this condition with the predicate `misaligned`:

$$\text{misaligned } \textit{iff } (\text{LU/MARKED}- \wedge \text{iru}(\text{LU/UTTERANCE})) \vee \text{LU/MARKED}+ \wedge \neg\text{iru}(\text{LU/UTTERANCE})$$

In determining when to mark tutor utterances as informationally redundant, the tutoring manager uses the CG as input when it is asked to generate tutorial content. This way it can check if $\texttt{iru}(u)$ holds of a planned utterance u and if so add marking.

4 Examples

We now illustrate how our model accounts for the examples above. For the purpose of example (1) we let p_1 stand for the proposition embedded in utterance S10, so that $p_1 = \texttt{expresses}(\texttt{S10})$. The domain contribution realised in S10 triggers the updates $\texttt{push}(\texttt{uttered}(s, p_1),\text{CG/PROPS})$ and $\texttt{push}(i_{p_1},\text{CG/REFS})$. The `accept` performed by the totor then triggers the update $\texttt{push}(\texttt{holds}(p_1),\text{CG/PROPS})$, and the resulting CG is

$$\left[\text{CG}\begin{bmatrix}\text{REFS} & \langle i_{p_1}, \ldots\rangle \\ \text{PROPS} & \langle \texttt{uttered}(s, p_1), \texttt{holds}(p_1), \ldots\rangle\end{bmatrix}\right]$$

When S18 is performed the predicate `misaligned` becomes true. This is because S18 is an IRU (the proposition it expresses is a match of p_1) but is unmarked for informational redundancy. The system concludes that misalignment has taken place and the tutoring module generates the reminder that the student should already believe that $\texttt{holds}(p_1)$, helping him to realign.

Example (2) shows a direct reference to a fact that the student should know. As in example (1), the contribution S3 followed by the acceptance T3 results in CG/PROPS containing $\texttt{holds}(p_2)$, where $p_2 = (b, a) \in (R \circ S)$. In S6 the student assumes this fact again and the system can determine that `misaligned` holds because the formula in S6 matches p_2. It performs $\texttt{makesalient}(i_{p_2})$, and the tutoring module reminds the student that he already knows the fact p_2 (T6).

As a result of the utterances S2 and T4 in example (3), the CG includes $\texttt{uttered}(s, p_3)$ and $\texttt{holds}(p_3)$, where p_3 is the formula $A \cap B = \emptyset$. When the system recapitulates the solution in T8, one of the utterances expresses a proposition which matches p_3. That means that $\texttt{exists}(p_3)$ holds and that this utterance is an IRU. So that the student does not mistakenly conclude that misalignment took place, the system generates the utterances augmented with the marking “of course” to indicate informational redundancy.

We treat example (4) in more detail because it shows how misalignment can be detected and repaired. In Figure 1 we saw the state of the dialogue model after utterance S3. T4 is a `reject`, so the model is updated to

$$\left[\text{CG}\begin{bmatrix}\text{REFS} & \langle i_{p_4}, \ldots\rangle \\ \text{PROPS} & \langle \neg\texttt{holds}(p_4), \texttt{prev}(P(A) \cup \ldots)\rangle\end{bmatrix}\right]$$

where $p_4 = \texttt{expresses}(\texttt{S3})$. The student has a misconception that $P(A) \cup P(B) = P(A \cup B)$ holds. Since this rule is not correct, there is no proposition $\texttt{prev}(P(A) \cup P(B) = P(A \cup B))$ in CG/PROPS. That means that when the utterance S4, in which the student checks whether the misconceived rule is correct or not, is performed, we have $\neg\texttt{iru}(\texttt{S4})$. However, the marking of S4 with the particle “doch” signals that the student assumes it to be shared knowledge. From these two facts the system detects that misalignment occurred. This type of misalignment informs the tutoring module to execute a strategy to resolve a misconception, namely, the student is referred to the study material. The resulting state is

$$\left[\text{CG}\begin{bmatrix}\text{REFS} & \langle i_{p_5}, i_{p_4}, \ldots\rangle \\ \text{PROPS} & \langle \neg\texttt{holds}(p_5), \neg\texttt{holds}(p_4), \ldots\rangle\end{bmatrix}\right]$$

In S5, with $p_6 = \texttt{expresses}(\texttt{S5})$, the student tries to correct the proof step that was rejected in utterance S3 by using a different rule, but the rule he applies (that $P(A) \cup P(B) \supseteq P(A \cup B)$) is not the correction of his original misconception, and the step is rejected (T5). The update is analogous, and we now have

$$\left[\text{CG}\begin{bmatrix}\text{REFS} & \langle i_{p_6}, i_{p_5}, i_{p_4}, \ldots\rangle \\ \text{PROPS} & \langle \neg\texttt{holds}(p_6), \neg\texttt{holds}(p_5), \ldots\rangle\end{bmatrix}\right]$$

The domain contribution S6 is an accepted (T7), correct application of the previously misconceived rule. By applying the correct rule the student shows that he has resolved the misconception that became apparent in utterance S4. The CG has now been realigned by adding $\texttt{holds}(P(A) \cup P(B) \subseteq P(A \cup B))$, and this information can be passed to the tutoring module.[5]

5 Related Work

Jordan and Walker (1996) compare models of how agents in a collaborative setting decide to remind each other about salient knowledge, and argue for an approximate rather than detailed model of the attentional state. In tutoring, the decision to remind is further influenced by pedagogical strategies. Our model provides input to this decision making process, however, the decision itself is made by the tutoring module. For example, the contents of CG could be used to realise a *Point-to-information* hint which is part of the hint taxonomy proposed by (Tsovaltzi et al., 2004).

Baker et al. (1999) argue that learning from grounding is the basis of collaborative learning and Pata et al. (2005) show how student grounding acts serve to inform tutoring scaffolds. Intelligent tutoring systems, such as AutoTutor (Person et al., 2000) and Ms Lindquist (Heffernan and Koedinger, 2002), with simple dialogue models have no model of CG, but capture misconceptions using explicit buggy rules. In those systems, there is no clear separation between modeling the dialogue itself and modeling the tutoring task. The dialogue advances according to the local tutoring agenda. (Zinn, 2004) presents a dialogue-based tutoring system in which discourse obligations are generated from a store of task solution descriptions and the CG is maintained in the dialogue model. However, the choice of tutoring actions is not informed by the state of the CG, but rather is explicitly encoded.

6 Conclusion and Further Work

We presented a preliminary model of common ground for a domain where grounding propositional content is crucial. However in principle this model is general enough to be be applied to other domains. The model takes into account Clark's distinction between proposal and acceptance of dialogue contributions.

Indeed the current model is somewhat simplistic. There are a number of aspects of grounding which we observe in our corpus which this model does not account for but could be extended for, for instance domain content which is in a "pending" state when argumentation is taking place. Our further work will include extending the model to a larger set of dialogue moves including grounding acts. To obtain a more fine-grained model of context we need to further investigate what additional information about the problem-solving steps the domain reasoner can provide to the dialogue model, and thus to the tutoring manager. Furthermore we need a model of salience of propositions and steps in the problem-solving task, which may require a more flexible data structure. In a broader context it may be necessary to consider deletion of propositions however the conditions under which deletion rather than decay should occur need to be investigated. Current work includes implementation in TrindiKit.

References

Michael Baker, Tia Hansen, Richard Joiner, and David Traum. 1999. The role of grounding in collaborative learning tasks. In Pierre Dillenbourg, editor, *Collaborative Learning. Cognitive and computational approaches*, Advances in Learning and Instruction Series, pages 31–63. Pergamon, Amsterdam, The Netherlands.

Christoph Benzmüller, Helmut Horacek, Henri Lesourd, Ivana Kruijff-Korbayová, Marvin Schiller, and Magdalena Wolska. 2006. A corpus of tutorial dialogs on theorem proving; the influence of the presentation of the study-material. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2006)*, pages 1766–1769, Genoa, Italy.

Chad Edward Brown. 2006. Dependently Typed Set Theory. SEKI-Working-Paper SWP–2006–03, Saarland University.

Herbert H. Clark and Edward F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13(2):259–294.

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

David DeVault and Matthew Stone. 2006. Scorekeeping in an uncertain language game. In *Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue (brandial)*, pages 139–146, Potsdam, Germany.

Dominik Dietrich and Mark Buckley. 2007. Verification of Proof Steps for Tutoring Mathematical Proofs. In *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, Los Angeles. To appear.

[5] The tutoring manager should record and make use of the fact that the student had and repaired a defective context. However we do not treat this topic here.

A. C. Graesser, N. K. Person, and J. P. Magliano. 1995. Collaborative dialogue patterns in naturalistic one-on-one tutoring. *Applied Cognitive Psychology*, 9:495–522.

Neil T. Heffernan and Kenneth R. Koedinger. 2002. An Intelligent Tutoring System Incorporating a Model of an Experienced Human Tutor. In *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, pages 596–608, London, UK.

Pamela W. Jordan and Marilyn A. Walker. 1996. Deciding to Remind During Collaborative Problem Solving: Empirical Evidence for Agent Strategies. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI*, volume 1, pages 16–23, Portland, OR.

Elena Karagjosova. 2003. Marked informationally redundant utterances in tutorial dialogue. In Ivana Kruijff-Korbayová and Claudia Kosny, editors, *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Language (DiaBruck)*, pages 189–190, Saarbrücken, Germany.

J. F. Kelley. 1984. An iterative design methodology for user-friendly natural language office information applications. *ACM Transactions on Information Systems*, 2(1):26–41.

Victor R. Lee and Bruce L. Sherin. 2004. What makes teaching special? In *ICLS '04: Proceedings of the 6th international conference on Learning sciences*, pages 302–309, Santa Monica, CA.

D. McArthur, C. Stasz, and M. Zmuidzinas. 1990. Tutoring techniques in algebra. *Cognition and Instruction*, 7(3):197–244.

D. C. Merrill, B. J. Reiser, S. K. Merrill, and S. Landes. 1995. Tutoring: Guided learning by doing. *Cognition and Instruction*, 13:315–372.

Roger H. Munger. 1996. Asymmetries of knowledge: What tutor-student interactions tell us about expertise. Paper presented at the Annual Meeting of the Conference on College Composition and Communication, Milwaukee, WI.

Kai Pata, Tago Sarapuu, and Raymond Archee. 2005. Collaborative scaffolding in synchronous environment: congruity and antagonism of tutor/student facilitation acts. In D. D. Suthers T. Koschman and T.-W. Chan, editors, *Computer Supported Collaborative Learning 2005: The next 10 years*, pages 484–493. Kluwer.

Natalie K. Person, Laura Bautista, Roger J. Kreuz, Arthur C. Graesser, and the Tutoring Research Group. 2000. The Dialog Advancer Network: A Conversation Manager for AutoTutor. In *Proceedings of the Workshop on modeling human teaching tactics and strategies at the 5th International Conference on Intelligent Tutoring Systems (ITS-00)*, LNCS, pages 86–92, Montreal, Canada.

Jörg Siekmann, Christoph Benzmüller, and Serge Autexier. 2006. Computer Supported Mathematics with ΩMEGA. *Journal of Applied Logic*, 4(4):533–559.

Robert Stalnaker. 2002. Common ground. *Linguistics and Philosophy*, 25(5):701–721.

David R. Traum. 1994. A computational theory of grounding in natural language conversation. Technical Report TR545, University of Rochester, Rochester, NY, USA.

Dimitra Tsovaltzi, Armin Fiedler, and Helmut Horacek. 2004. A multi-dimensional taxonomy for automating hinting. In James C. Lester, Rosa Maria Vicari, and Fábio Paraguaçu, editors, *Intelligent Tutoring Systems — 7th International Conference (ITS-04)*, number 3220 in LNCS, pages 772–781. Springer.

Marilyn Walker and Owen Rambow. 1994. The Role of Cognitive Modeling in Achieving Communicative Intentions. In *Proceedings of the 7th International Workshop on Natural Language Generation.*, pages 171–180, Kennebunkport, ME.

Marilyn Walker. 1993. *Informational Redundancy and Resource Bounds in Dialogue*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA.

Magdalena Wolska and Ivana Kruijff-Korbayová. 2004. Analysis of Mixed Natural and Symbolic Language Input in Mathematical Dialogs. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL-04)*, pages 25–32, Barcelona, Spain.

Magdalena Wolska, Bao Quoc Vo, Dimitra Tsovaltzi, Ivana Kruijff-Korbayova, Elena Karagjosova, Helmut Horacek, Malte Gabsdil, Armin Fiedler, and Christoph Benzmüller. 2004. An annotated corpus of tutorial dialogs on mathematical theorem proving. In *Proceedings of International Conference on Language Resources and Evaluation (LREC-04)*, pages 1007–1010, Lisbon, Portugal.

Claus Zinn. 2004. Flexible dialogue management in natural-language enhanced tutoring. In *Konvens 2004 Workshop on Advanced Topics in Modeling Natural Language Dialog*, pages 28–35, Vienna, Austria.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 49–56.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Managing ambiguities across utterances in dialogue

David DeVault and **Matthew Stone**
Department of Computer Science
Rutgers University
Piscataway, NJ 08845-8019
David.DeVault@rutgers.edu, Matthew.Stone@rutgers.edu

Abstract

Task-oriented dialogue systems exploit context to interpret user utterances correctly. When the correct interpretation of a user utterance is ambiguous, a common response is to employ a special process of clarification that delays context update until important ambiguities are resolved, so that the main dialogue task can proceed with an unambiguous context. In this paper, we describe an implemented dialogue agent which instead translates ambiguities in interpretation into uncertainty about which context has resulted from an utterance. It then uses question-asking strategies, including clarification as a special case of questions about speaker meaning, to manage its uncertainty across multi-utterance subdialogues. We analyze the agent's use of these strategies in an empirical study of task-oriented dialogues between the agent and human users.

1 Introduction

Dialogue agents cannot always understand their human partners. Indeed, we ourselves do not always understand what others say to us. Nevertheless, *our* conversational abilities allow us to follow up provisional interpretations of what has been said and eventually arrive at a sufficient understanding. This paper reports work on designing dialogue agents that can do the same.

The specific problem we address in this paper is how to reason about context-dependence while working to reduce ambiguity and achieve common ground. Every utterance in conversation gets its precise meaning in part through its relationship to what has come before. This applies to the clarificatory utterances interlocutors use to acknowledge, reframe or question others' contributions just as it does to fresh contributions. The distinctive issue with such followups is that they must be formulated for a context about which speaker or addressee may be uncertain. The speaker must be able to assess that addressees will understand and respond helpfully to them no matter what the context might be.

In this paper, we present a model that frames this reasoning as ordinary collaborative language use in the presence of contextual ambiguities. We describe how dialogue agents come to be uncertain about what their interlocutors have contributed, and offer a precise characterization of how agents can formulate context-dependent utterances that help pinpoint the context and resolve ambiguity. A dialogue agent that uses such utterances can play its collaborative role in working to understand its interlocutors.

Our model is implemented in COREF, a task-oriented dialogue system that collaboratively identifies visual objects with human users. We show empirically that to interact successfully in its domain, COREF does need to work collaboratively to resolve ambiguities, and moreover that our model makes COREF to some degree successful in doing so. At the same time, we highlight qualitative aspects of COREF's behavior that depend on our new synthesis of linguistic and collaborative reasoning. For example, we show how COREF needs both linguistic reasoning and collaborative reasoning to formulate followups that offer alternative descriptions of things it judges its interlocutors might have meant.

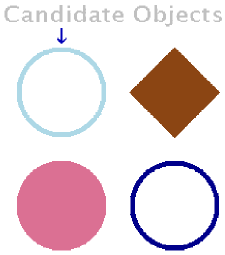
Our work is part of a larger project on reconciling linguistic reasoning and collaborative reasoning in conversation (Stone, 2004; DeVault and

Stone, 2006; Thomason et al., 2006). In particular, we build on the account of communicative intentions of Stone (2004), on the account of context update for communicative intentions of DeVault and Stone (2006), and on the model of collaboration in conversation from Thomason et al. (2006). We advance this program here by weakening many of the idealizations about mutuality that we have made explicitly or implicitly in earlier work. Thus, we are able to go significantly further towards an account of the reasoning and skills that agents use to overcome differences in achieving mutual understanding.

2 Related Work

Our work is an attempt to use the theory of collaboration to bridge two different traditions for specifying dialogue agency. The first is engineering approaches to spoken dialogue systems, where researchers have shown that systems should represent the uncertainty of their automatic speech recognition results and take that uncertainty into account in their dialogue management strategies. For example, maintaining a probability distribution over alternative recognition results can help a system to choose whether to clarify user input or proceed with a possibly incorrect interpretation (Roy et al., 2000; Horvitz and Paek, 2001; Williams and Young, 2007). It also allows statistical inference to combine evidence about user intentions from multiple utterances (Bohus and Rudnicky, 2006). Such research connects uncertainty to systems' high-level choices, but because it focuses on modeling user state rather than utterance context, it cannot connect uncertainty to principled compositional linguistic reasoning such as decision-making in natural language generation.

The other tradition is deep approaches to dialogue coherence, where researchers provide detailed models of evolving utterance context in dialogue and of the linguistic constructions that exploit this context. These models go much further in accounting for the specific utterances speakers can use in context for grounding and clarification. However, these models often create explanatory tension by running together descriptions of how utterances update the context with descriptions of how interlocutors manage uncertainty. For example, when a new utterance occurs, its content may be marked *ungrounded* to reflect the fact that its content must be acknowledged by the hearer before it can be assumed to have been understood (Traum, 1994; Poesio and Traum, 1997). However, acknowledgments in dialogue don't really always function to put specified content unambiguously onto the common ground (Clark and Schaefer, 1989). For example, Figure 1 provides a naturally occurring fragment of human–human dialogue in COREF's domain, where interlocutors treat an utterance of *okay* as ambiguous. In this interaction, S15 and S14 converse via teletype from separate rooms. S15 begins by instructing S14 to click on a certain object in S14's display. S14 does so, but S15 cannot observe the action. This leads S15 to perceive an ambiguity when S14 says *okay*: has S14 merely grounded S15's instruction, or has S14 also clicked the object? The ambiguity *matters* for this task, so S15 engages the ambiguity with a followup question.



S15: Okay, add the light blue empty circle please.
[S14 privately adds the object]
S14: okay
S15: Okay, so you've added it?
S14: i have added it. It is in the top left position.

Figure 1: An ambiguous grounding action by subject S14 in a human-human dialogue.

Similarly, utterances that are perceived as ambiguous in important ways may be modeled as suspended until a special process of clarification resolves the relevant ambiguity (Ginzburg and Cooper, 2004; Purver, 2004). But the problem of recognizing and responding to perceived ambiguities in a collaboration is more general than the problem of clarifying utterances. For example, in the task domain of Figure 1, the question *you've added it?* serves to resolve ambiguity just like a clarification might, but it arises from the non-public nature of the "add object" action rather than from any grammatically-specified dynamics of context update (Purver, 2004).

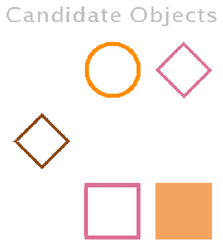
Finally, connecting context update to the resolution of perceived ambiguities may guarantee common ground, but leaving ambiguities open can make a collaborative agent more flexible. An agent that demands a clear context but lacks the resources to clarify something may have no recourse but to take a "downdate" action—to signal to the user that their intended contribution was not understood, and discard any alternative possible contents. If the agent can proceed, however, the agent may get evidence from what happens next to resolve its uncertainty and complete the task.

We view uncertainty management and context update as necessary but independent processes; this positions our work between the two traditions. We follow more applied work in representing uncertainty in the context probabilistically, and modeling grounding and clarification as collaborative mechanisms interlocutors can use to reduce but perhaps not eliminate this uncertainty. But we follow deeper models in using a precise dynamic semantics to characterize the evolving utterance context and its effects on utterance interpretation.

3 Technical Approach

We present our ideas through examples of referential communication. Our specific setting is based on the collaborative reference task studied in pairs of human subjects by Clark and Wilkes-Gibbs (1990). Each interlocutor perceives a collection of visual objects, as illustrated in Figures 1–2. The interlocutors perceive identical objects, but with shuffled spatial locations. One interlocutor, who we call the director, sees a target object highlighted on their display with an arrow, and is charged with conveying to their partner, who we call the matcher, which of the displayed objects is the target. The interlocutors go through the objects one by one, with the matcher attempting to identify and click on the correct target at each step.

We have implemented an agent COREF which can participate in these dialogues (DeVault and Stone, 2006). Figure 2 shows a sample interaction between COREF and a human user. We will use this interaction to illustrate how COREF frames clarification as an ambiguity management problem. Here, COREF has perceived an ambiguity in the user's intention in uttering *it is brown*, and decides to clarify with *do you mean dark brown?*



possible contexts	agent	actor
c_1	COREF:	is the target pink?
c_2	A18:	no
c_3	A18:	it is brown
c_{4l}, c_{4d}	COREF:	do you mean dark brown?
c_{5l}, c_{5d}	A18:	yes
c_6	COREF	(privately adds the brown diamond)
c_6	COREF:	done

Figure 2: COREF asks a clarification question.

The model that realizes COREF's behavior here incorporates three new principles. First, the model exposes ambiguity about what the user means as uncertainty in the dialogue state that results from the user's utterance. Here COREF assumes that the user intends to identify the color of the target object with *it is brown* and therefore finds two possible interpretations: one for the dark brown color of the empty diamond and one for the light brown color of the solid square. After the utterance, COREF is uncertain about which meaning was intended and thus which constraint the user has contributed.

Second, the model allows the specification of dialogue strategies that allow COREF to proceed with appropriate high-level dialogue moves despite having more than one alternative for what the context is. Here COREF settles on a clarification move, because we have specified a policy of clarifying ambiguities reflecting different constraints on the target object. In other kinds of uncertain contexts, COREF will proceed without clarifying.

Third, COREF plans its generation decisions so that the user will recover a specific and useful interpretation of what it says no matter what the context is. Here COREF explicitly constructs the utterance *do you mean dark brown* by carrying out an incremental derivation using a lexicalized grammar. The rich representation of the utterance context allows the system to recognize the applicability of forms that cohere with what has gone before, such as the use of the frame *do you mean* to refer to content from the previous utterance, whatever it may have been. The model predicts that this underspecification is unproblematic, but predicts that the ambiguity of *brown* must be eliminated and therefore motivates the adjunction of the modifier *dark*.

In this section, we sketch the implementation of COREF, briefly summarizing the details we carry over from previous presentations, and highlighting the differences that support the implementation of the three new principles.

3.1 Context, Tasks, Actions, and Uncertainty

We follow DeVault and Stone (2006) in understanding the utterance context at each point in a dialogue as an *objective* and *normative* product of prior interlocutor action. The context for COREF describes both the state of the ongoing referential activity and the semantic and pragmatic status of information in the dialogue. Activity is represented through a stack of ongoing tasks, drawn from an inventory including COREF's overall multi-object reference task, its single-object reference task, a yes/no question task, a reminder question task, a clarification task, and an ambiguity management task (`ManageAmbiguity`) that is automatically pushed after each utterance or action. The linguistic context, meanwhile, includes aspects of the discourse history such as specifications of recent utterances and of salient referents.

Dialogue allows interlocutors to change the context by manifesting a suitable communicative intention—in other words by taking an observable action with a specific commitment as to how the action will link up with and update the context (Stone, 2004). This is formalized by a function $\texttt{update}(c, i)$ which describes the context that results from acting with intention i in context c. However, interlocutors actually observe actions rather than intentions and so must recognize the intention from knowledge of language and of the ongoing task. Thus, while COREF tries to identify the true context at each point in time, it is sometimes uncertain about it, as when there is perceived ambiguity in its interlocutor's intentions. The basic interpretive operation in COREF is not *updating*—that is, tracking deterministic context change—but *filtering*—propagating uncertainty about the context at time t to uncertainty about the context at time $t+1$ based on an observed action.

We follow Thomason et al. (2006) in characterizing filtering in COREF's domain through *tacit actions* as well as observable actions. Tacit actions include task-relevant cognitive actions like identifying the target object or abandoning a task. A speaker is free to use tacit actions as well as observable actions to update the context. However, successful coordination requires the speaker to provide sufficient evidence in their observable actions to reconstruct any tacit actions they have committed to. Formally, for any context c and interlocutor S, we can use the next actions that could contribute to the pending tasks in c to determine a set of alternative contexts $Z(c, S)$ that could be reached by S from c just using tacit actions. We call this set of alternative contexts the *horizon*.

The horizon allows us to make an agent's filtering operation precise. Let us write $c\!:\!i$ to denote an interpretation which shows the speaker (or actor) acting in context c with a commitment to intention i. In understanding, an agent H starts from a prior probability distribution over the initial context at time t given the evidence E available so far: $P_H(c_t|E)$. H observes an action a_t (carried out by agent S), and must infer $\hat{c}_t\!:\!i_t$ to explain that action. H can assume that the new context $\hat{c}_t$ must be some element of $Z(c_t, S)$, and that i_t must match action a_t into $\hat{c}_t$ so as to contribute to the ongoing tasks. H will inevitably bring substantial background knowledge to bear, such as grammatical knowledge and interpretive preferences. However, H's evidence may still leave multiple options open. We summarize H's intention recognition as a probabilistic likelihood model $P_H(\hat{c}_t\!:\!i_t|c_t, a_t)$. (As usual, we assume the context tells you everything you need to know about the current state to interpret the action.) Filtering combines update, prior and likelihood:

$$P_H(c_{t+1}|a_t, E) \propto \sum P_H(\hat{c}_t\!:\!i_t|c_t, a_t) P_H(c_t|E)$$

where the summation ranges over all values of c_t, $\hat{c}_t$, and i_t such that $c_{t+1} = \texttt{update}(\hat{c}_t, i_t)$.

We illustrate this model through COREF's reasoning on A18's utterances *it is brown* and *yes*, the third and fifth utterances from Figure 2. For the first of these utterances, COREF starts with just one context c_3 with any probability. There are two possible interpretations i_{3l} and i_{3d} corresponding to the different colors (*light* and *dark* brown respectively) that might be picked up by *brown*; COREF's model happens to assign them equal probability. Each interpretation involves a tacit move to a context $\hat{c}_3$ which implicitly completes any discussion of the contribution of the user's previous utterance *no*. Filtering therefore results in two possible values for the next context, $c_{4l} = \texttt{update}(\hat{c}_3, i_{3l})$ and

$c_{4d} = \texttt{update}(\hat{c}_3, i_{3d})$. Each is assigned probability 0.5. Ambiguity in interpretation has been exposed as uncertainty in the context.

For the second of these utterances, *yes*, COREF starts with *two* equally probable contexts c_{5l} and c_{5d} which (as we shall see further below) are derived from taking into account the effect of COREF's tacit actions and clarification question in contexts c_{4l} and c_{4d}. Here the context-dependence of *yes* means that COREF must find an interpretation in which the user gives an appropriate affirmative answer to the salient question (in the context $\hat{c}_{5l}$ or $\hat{c}_{5d}$ following a tacit action closing discussion of COREF's meaning). That question is whether the user meant *dark brown* by *brown*. The *yes* answer is appropriate in contexts derived from c_{4d} because that is what the user meant there, but not in contexts derived from c_{4l} where the user meant something else. So across all the candidate contexts only one interpretation i_5 can be assigned nonzero probability. Accordingly filtering restores all the probability mass to $c_6 = \texttt{update}(\hat{c}_{5d}, i_5)$.

3.2 Minimizing Ambiguity

Our discussion thus far has shown how interlocutors can interpret utterances in succession as creating and resolving temporary ambiguities. Our goal, however, is to design dialogue agents that can not only deal passively with ambiguity, but can collaborate actively to resolve ambiguities with their interlocutors. This means giving agents high-level strategies that are helpful in dealing with uncertainty, and generating natural language utterances that do not exacerbate the problems of ambiguity even when used in uncertain contexts.

COREF includes a hand-built action policy that decides which contributions to the conversation would be *acceptable* for the agent to take *given its current uncertainty*. For example, COREF's policy deems it acceptable to ask for clarification any time COREF is uncertain which constraint a speaker intended to add with an utterance, as in Figure 2. Similarly, COREF's action policy deems it acceptable for the agent to ask whether a non-public action m has occurred, if some possible contexts but not others indicate that m has taken place. For example, COREF translates an ambiguous acknowledgment like that of Figure 1 into uncertainty about whether the "add object" action has tacitly occurred in the true context; COREF follows up such an *okay* by asking *did you add it?*

COREF's generation module is tasked with formulating an utterance that makes these contributions in a way its interlocutor will understand. In Thomason et al. (2006) we investigate a *strong* notion of recognizability. Each utterance must result in a *checkpoint* where speaker and hearer agree not only on a unique interpretation for the utterance but also on a unique resulting context. Enforcing this constraint supports the traditional attribution of mutual knowledge to the two interlocutors at each point in the conversation.

Here we develop a more flexible notion of *weak recognizability* that allows for uncertain contexts and makes interpretation more robust to potential differences in their perspectives. In interpreting a user utterance, COREF expects to find zero, one, or multiple interpretations in each possible context. In generation, COREF is sometimes willing to take the risk of using an action or utterance that may not be interpretable in all possible contexts. Taken together, this means new utterances can serve not only to present the speaker's intention, but also in some cases to introduce or defuse uncertainties about the true context. Checkpoints, where COREF achieves certainty about the true context, arise as side effects of this dynamic rather than as a strict requirement in the architecture. While there is no guarantee that any given speaker contribution will ever become common ground, COREF's dialogue policies are designed to try to achieve common ground when it is practical to do so.

Our formal development assumes that agents can take their own probabilistic models of interpretation as good indicators of their partners' disambiguation preferences (for example by slightly overspecifying their utterances). More precisely, we will allow each interlocutor to discard certain interpretations whose probability falls below a threshold ϵ and so are of sufficiently low probability, relative to others, that they can safely be ignored. Consider then an observable action a by S. If there were only a single possible context c, the set of recognized interpretations for a would be $R(c, a) = \{\hat{c} : i | P(\hat{c} : i | c, a) \gg \epsilon\}$. But in general, S is uncertain which of $C = \{c_1, ..., c_k\}$ is the true context, and expects that H may give any of these a high prior and take seriously the corresponding interpretations of the utterances. Indeed, S must also be prepared that S is actually making any of these contributions. In other words H and S will consider any interpretation in $R^*(C, a) = \cup_{c \in C} R(c, a)$. $R^*(C, a)$ is weakly recognizable if and only if each $c_i \in C$ is associated with at most one interpretation in $R^*(C, a)$.

The formalism explains why, in generation, COREF chooses to elaborate its utterance *do you mean brown* by adding the word *dark*. COREF's policy makes a clarification question acceptable across all of the candidate contexts after the user says *it is brown*. But *do you mean brown* is not weakly recognizable. For example, in c_{4d}, there are two interpretations, which could be paraphrased *do you mean light brown* and *do you mean dark brown*. COREF therefore chooses to coordinate more finely on the alternative interpretations of its clarification action. The utterance *do you mean dark brown* has only one interpretation in each of c_{4l} and c_{4d} and therefore represents a solution to COREF's communicative goal.

3.3 Strategically Discarding Ambiguities

To keep search tractable for real-time interaction, COREF tracks a maximum of 3 contexts. If more than 3 are possible, the 3 most probable are retained, and the others discarded. Further, after each object is completed, COREF discards all but the most probable context, to avoid retaining unilluminating historical ambiguities. In fact, according to COREF's action policy, it is acceptable to complete an object despite an ambiguous context, provided the ambiguity does not affect the agent's judgment about the target object—this is COREF's analogue of a "grounding criterion".

4 Empirical Results

We recruited 20 human subjects[1] to carry out a series of collaborative reference tasks with COREF. The study was web-based; subjects participated from the location of their choice, and learned the task by reading on-screen instructions. They were told they would work with an interactive dialogue agent rather than a human partner. Each subject worked one-by-one through a series of 29 target objects, for a total of 580 objects and 3245 utterances across all subjects. For each subject, the 29 target objects were organized into 3 groups, with the first 4 in a 2x2 matrix, the next 9 in a 3x3 matrix, and the final 16 in a 4x4 matrix. As each object was completed, the correct target was removed from its group, leaving one fewer object in the matrix containing the remaining targets. The roles of director and matcher alternated with each group of objects. Either COREF or the subject was randomly chosen to be director first.

`correct`	`no object`	`skipped`	`wrong`
75.0%	14.3%	7.4%	3.3%

Table 1: Overall distribution of object outcomes.

1 context	2 contexts	3 contexts
83.4%	6.8%	9.8%

Table 2: Number of possible contexts perceived when utterances or actions occur.

The experiment interface allows an object to be completed with one of four outcomes. At any time, the matcher can click on an object to add it to her "scene," which is another matrix containing previously added objects for the same group. An object is completed when the director presses either the `continue` or `skip` button, or when the matcher presses `skip`. An outcome is scored `correct` if the director presses `continue` after the matcher has added the correct target to her scene. It is scored `skipped` if either interlocutor presses the `skip` button.[2] It is scored `no object` or `wrong` if the director presses `continue` before the matcher adds any object, or after the matcher adds the wrong object, respectively.

Table 1 shows COREF's overall performance in the task. We would like to understand this performance in terms of COREF's uncertainty about the context. To begin, Table 2 shows the distribution in the number of alternative contexts perceived by COREF across all subjects. COREF is usually completely certain what the true context is, but is uncertain about 17% of the time.[3] To better understand how this uncertainty affects object outcomes, we investigated the agent's performance during the subdialogues associated with individual objects, which had a mean length of 5.6 utterances. Figure 3 shows the relation between the mean number of possible contexts during an object subdialogue and the outcome for that dialogue. The figure shows that high mean uncertainty has a clear negative impact on object outcomes, but a smaller degree of uncertainty is less harmful, if at all. In total, 13.1% of COREF's `correct` ob-

[1] Most of the subjects were undergraduate students participating for course credit at Rutgers University.

[2] Though note that COREF never presses `skip`.

[3] Since COREF truncates its uncertainty at 3 possible contexts, the higher frequency of 3 possible contexts relative to 2 here very likely masks a longer underlying tail.

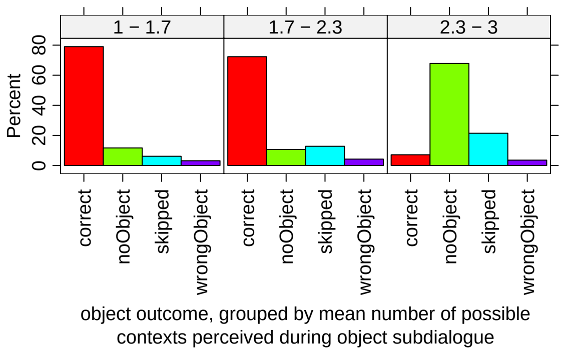


Figure 3: Object outcome vs. context uncertainty.

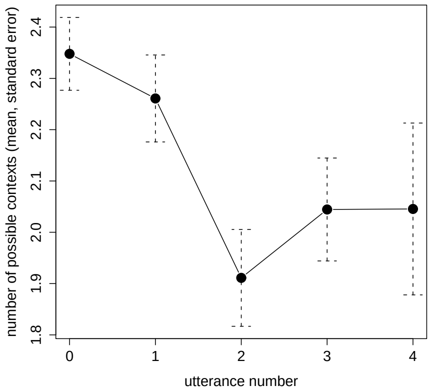


Figure 4: Effect of ambiguity management questions on COREF's uncertainty. At utterance 0, COREF faces an ambiguous context. At utterance 1, COREF has asked a question. Utterance 2 is typically an answer by the subject.

ject outcomes occur at a moment when COREF is uncertain what the true context is (9.7% two contexts, 3.4% three contexts).

While certainty about the context is not strictly necessary for a `correct` outcome, COREF nevertheless does often try to reduce its uncertainty according to its question-asking policy. Figure 4 illustrates the effectiveness of COREF's question-asking policy at reducing uncertainty. As the figure shows, when COREF asks questions in an ambiguous context, the mean reduction in the agent's uncertainty is about 0.4 contexts. Figure 2 is an example where the subject's answer eliminates a context. But the subject's answer does not always reduce uncertainty, because it may introduce a new ambiguity.[4] Figure 1 actually gives such an example in a human-human dialogue. In this dialogue, from S15's perspective, it is possible that S14 had already added the object to the scene; but it is also possible that S14 took the question as a reminder to add the object to the scene and answered in the affirmative only after correcting the error. This distinction does not matter for task success, but it does introduce a potentially lasting ambiguity into the dialogue history. When COREF's questions do not resolve an ambiguity, COREF does not force a downdate; it tries instead to proceed with the task. Figures 3 and 4 suggest that COREF's ambiguity management mechanisms are relatively successful in cases of mild or short-lived ambiguities.

5 Discussion and Future Work

We have presented a framework that allows task-oriented dialogue agents to use language collaboratively despite uncertainty about the context. We have presented empirical evidence that managing ambiguity is a key task for dialogue agents such as ours, and that it can be addressed successfully within a uniform architecture for collaboration under uncertainty. In particular, our model shows how dialogue agents can support grounding acknowledgments, clarification of ambiguous utterances, and task-oriented question asking using generic linguistic resources and goal-oriented ambiguity management strategies. For such an agent, what is distinctive about acknowledgments and clarification is simply their reference and relation to prior utterances; they play no special role in a language-specific context-update mechanism.

The proposed model is most applicable to situations in which the speaker's true intention is always among the alternative interpretations derived by the hearer. This is the case for the acknowledgments and clarifications of speaker meaning that occur frequently in COREF's domain, and that have been our focus to date. We believe our model could also be extended to clarifications of perceived ambiguities in phonology and syntax, drawing on the work of Ginzburg and Cooper (2004). Perceived phonologic or syntactic ambiguities could be translated into ambiguities in the context resulting from an utterance, entirely analogously to COREF's response to ambiguities of meaning.

However, our work does not immediately cover

[4] Other ways a question can fail to reduce uncertainty are if the user chooses not to answer the question or if the agent fails to understand the user's answer.

clarification questions that are not designed to resolve perceived ambiguities, but rather are asked in situations where *no* interpretations are found. Such examples occur; see Ginzburg and Cooper (2004) or Purver (2004) for examples. When COREF finds no interpretations for a user utterance, it notes the utterance and signals an interpretation failure (currently by saying *umm*), but it otherwise leaves its context representation as it was, and is unable to address the failure with its usual ambiguity management policy. Alternative characterizations of agents' reasoning in such cases are still required, and work such as Purver's provides a natural starting point.

Moreover, traditional classifications of grounding actions (Traum, 1999) include a variety of other cases as well. For example, we do not treat repair requests like *what?* or *what did you say?*, which can signal interpretation failure or the hearer's incredulity at the speaker's apparent (but correctly and uniquely identified) meaning. Similarly, we do not treat self-repairs by speakers. These can exclude a possible but unintended interpretation, to avoid a foreseen misunderstanding—an example in COREF's domain would be, *A: I moved it. A: I mean I moved the blue circle.* They can also correct a prior verbal mistake, as when a speaker has mistakenly used the wrong word: *A: I moved the circle. A: I mean I moved the square.* It would be interesting to explore whether richer models of domain uncertainty and dialogue context would enable us to account for these utterance types.

Ultimately, our framework suggests that agents face uncertainty from various sources, but that their experience provides quantitative evidence about what kinds of uncertainty arise and how best to resolve them. A final direction for our future research, then, is to analyze records of agents' interactions to develop decision-theoretic strategies to optimize agents' tradeoffs between asking clarification questions, resolving ambiguity to the most likely interpretation, and proceeding with an uncertain context.

Acknowledgments

We thank our anonymous reviewers for helpful comments. This work was supported by NSF HLC 0308121, IGERT 0549115 & HSD 0624191.

References

D. Bohus and A. Rudnicky. 2006. A k hypotheses + other belief updating model. In *Proceedings AAAI Workshop on Statistical and Empirical Approaches for Spoken Dialogue Systems*.

H. H. Clark and E. F. Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294.

H.H. Clark and D. Wilkes-Gibbs. 1990. Referring as a collaborative process. In P. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 463–493. MIT.

D. DeVault and M. Stone. 2006. Scorekeeping in an uncertain language game. In *Proceedings of the 10th Workshop on the Semantics and Pragmatics of Dialogue (SemDial-10)*, pages 139–146.

J. Ginzburg and R. Cooper. 2004. Clarification, ellipsis and the nature of contextual updates in dialogue. *Linguistics and Philosophy*, 27(3):297–365.

E. Horvitz and T. Paek. 2001. Harnessing models of users' goals to mediate clarification dialog in spoken language systems. In *User Modeling Conference*, pages 3–13.

M. Poesio and D. R. Traum. 1997. Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347.

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

N. Roy, J. Pineau, and S. Thrun. 2000. Spoken dialogue management using probabilistic reasoning. In *Proc. of ACL*, pages 93–100, Hong Kong.

M. Stone. 2004. Communicative intentions and conversational processes in human-human and human-computer dialogue. In Trueswell and Tanenhaus, editors, *World-Situated Language Use*, pages 39–70. MIT.

R. H. Thomason, M. Stone, and D. DeVault. 2006. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. To appear in Byron, D., Roberts, C., and Schwenter, S., eds, Presupposition Accommodation.

D. R. Traum. 1994. *A Computational Theory of Grounding in Natural Language Conversation*. Ph.D. thesis, Rochester.

D. R. Traum. 1999. Computational models of grounding in collaborative systems. In S. E. Brennan, A. Giboin, and D. Traum, editors, *AAAI Fall Symposium on Psychological Models of Communication*, pages 124–131.

J. Williams and S. Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 57–63.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Unifying Self- and Other-Repair

Jonathan Ginzburg
Department of Computer Science
King's College London
UK

Raquel Fernández and **David Schlangen**
Department of Linguistics
University of Potsdam
Germany

Abstract

We discuss similarities between mid-utterance self-correction, which is often seen as a phenomenon that lies outside the scope of theories of dialogue meaning, and other discourse phenomena, and argue that an approach that captures these similarities is desirable. We then provide a sketch of such an approach, using Ginzburg's KoS formalism, and discuss the implications of including 'sub-utterance-unit' phenomena in discourse theories.

1 Introduction

Unlike written language, spoken conversational language is full of what can be described as explicit traces of editing processes, as in the following example:[1]

(1) *I was one of the . I was responsible for all the planning and engineering*

In this example, the brief silence after *one of the* (represented here by a full stop) seems to prepare the 'editing operation' that is to follow in the form of a partial repetition of material, the result being the 'cleaned up' utterance *I was responsible for all the planning and engineering*, with the fragment *I was one of the* being struck from the record.

To our knowledge, this phenomenon of self-correction has not been dealt with in theories of dialogue meaning. And indeed, described as above, it seems that it is something that can safely be sourced out to 'earlier' processing stages which do the cleaning up, with the dialogue meaning then being defined over the cleaned up utterances.[2]

In this paper we will argue, following much work in the tradition of *conversational analysis* beginning with (Schegloff et al., 1977),[3] that there are, in fact, strong similarities between self-correction and other discourse phenomena (Section 3), which make an approach that captures these similarities desirable. In contrast to conversation analytic work, however, we actually ground our proposal in a formal model: in Section 4 we sketch such an approach, couched in terms of the KoS formalism (Ginzburg and Cooper, 2004; Purver, 2004; Ginzburg, (forthcoming)). We also discuss there the implications of making such a move for the grammar/parser–discourse interface and for discourse theories in general. Some conclusions are provided in Section 5.

Before coming to this, however, we briefly give some background on speech dysfluencies in the next section and review some of the terminology from the literature.

2 Form and Function of Dysfluencies

In this section we discuss the 'syntax' of self-correction, classifications according to the relation of problematic material and replacement, and the kinds of problems that can be corrected with self-correction.

As has often been noted (see e.g. Levelt (1983), and references therein for earlier work), speech dysfluencies follow a fairly predictable pattern. The example in Figure 1 is annotated with the labels introduced by Shriberg (1994) (building on (Levelt, 1983)) for the different elements that can occur in a self-repair.

[1] From the Switchboard corpus (Godfrey et al., 1992).

[2] This division of labour also seems to be presupposed by much of the computational work on automatically detecting and repairing dysfluent speech, as expressed e.g. in the following quote from (Heeman and Allen, 1999): "we propose that these tasks [a.o. detecting and correcting speech repairs, the authors] can be done using local context and early in the processing stream."

[3] "Although self-initiation and other-initiation of repair are distinct types of possibilities [...] There are quite compelling grounds for seeing self and other-initiations to be related, and for seeing their relatedness to be organized." (Schegloff et al., 1977)

until you're	*at the le-*		*I mean*	*at the right-hand*	*edge*
start	reparandum	moment of interruption	editing terms	alteration	continuation

Figure 1: General pattern of self-repair

Of these elements, the editing term is always optional (although some marking, like an extended pause, seems to be always present (McKelvie, 1998)). The relation between reparandum and alteration can be used as the basis of a further classification:[4] if the alteration differs strongly from the reparandum and does not form a coherent unit together with the start, or if alteration and continuation are not present at all, the dysfluency can be classified as an *aborted utterance / fresh start*. Other classes are *repair* (alteration 're-places' reparandum) and *reformulation* (alteration elaborates on reparandum). The following gives examples for all three classes:[5]

(2) a. { *I mean* } [*I,* + *I,*] *-/* [*there are a lot,* + *there are so many*] *different songs,*

b. [*We were* + *I was*] *lucky too that I only have one brother.*

c. *at that point,* [*it,* + *the warehouse*] *was over across the road*

Within the class of repairs, finally, a further distinction can be made (Levelt, 1983) into *appropriateness-repairs* that replace material that is deemed inappropriate by the speaker given the message she wants to express (or has become so, after a change in the speaker's intentions), and *error-repairs*, where the material is erroneous.

3 From Other to Self

Figure 2 shows (constructed) examples of 'normal' discourse correction (a), two uses of clarification requests (b & c), correction within a turn (d), other-correction mid-utterance (e), and two examples of self-correction as dicussed above (f & g). The first four examples clearly are instances of phenomena within the scope of discourse theories. What about the final two?

There are definite similarities between all these cases: (i) material is presented publicly and hence is open for inspection; (ii) a problem with some of the material is detected and signalled (= there is a 'moment of interruption'); (iii) the problem is addressed and repaired, leaving (iv) the incriminated material with a special status, but within the discourse context. That (i)-(iii) describe the situation in all examples in Figure 2 should be clear; that (iv) is the case also for self-corrections can be illustrated by the next example, which shows that self-corrected material is available for later reference and hence cannot be filtered out completely:[6]

(3) [*Peter was* + {*well*} *he was*] *fired*

Further evidence that the self-corrected material has a discourse effect is provided by Brennan and Schober (2001), who found that in a situation with two possible referents, the fact that a description was self-corrected enabled listeners to draw the conclusion that the respective other referent was the correct one, before the correction was fully executed. Similarly, (Lau and Ferreira, 2005) showed that material present in the reparandum can influence subsequent sentence processing.

The structural similarities established, we come to the question of the potential differences. There is a clear difference in the contextual possibilities across utterances, depending on whether a turn change occurs or not, as illustrated in (4) and (5):

(4) *A: Who likes Bo? Bo?* (= Does Bo like Bo?)

(5) *A: Who likes Bo?*
B: Bo? (= Does Bo like Bo? or Who do you mean 'Bo'? or Are you asking who likes BO?)

Indeed, in line with the observations of (Schegloff et al., 1977), it seems that the range of utterances that occur within utterance by a single speaker are distinct though not disjoint from those that occur by a distinct speaker at a transition relevance point:

[4] This classification is based on (McKelvie, 1998; Heeman and Allen, 1999).

[5] The examples in this section are all taken from the Switchboard corpus (Godfrey et al., 1992), with dysfluencies annotated according to (Meeter et al., 1995): '+' marks the moment of interruption and separates reparandum from alteration, '{}' brackets editing terms and filled pauses.

[6] The example is taken from (Heeman and Allen, 1999).

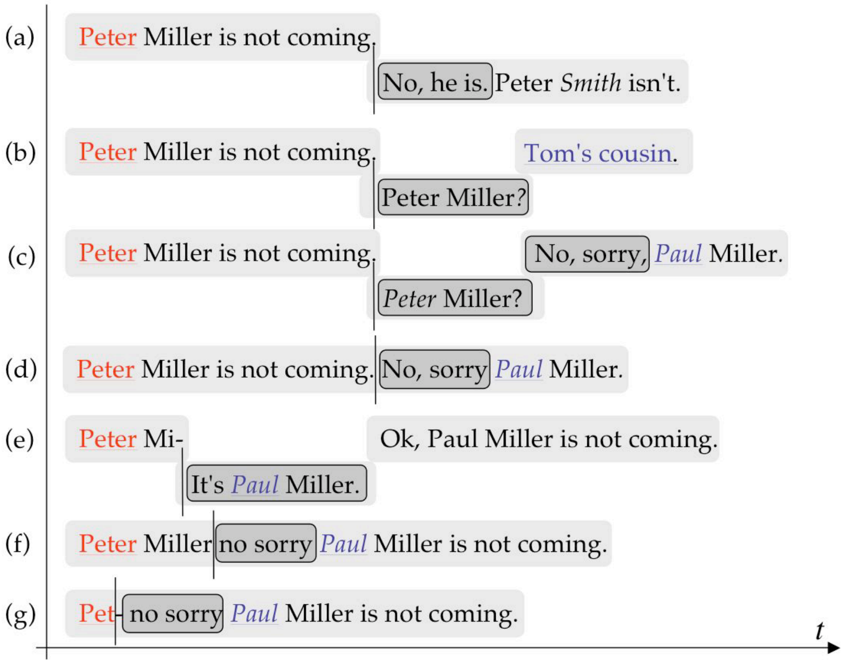

(a)
Peter Miller is not coming.
No, he is. Peter Smith isn't.
(b)
Peter Miller is not coming.
Tom's cousin.
Peter Miller?
(c)
Peter Miller is not coming.
No, sorry, Paul Miller.
Peter Miller?
(d)
Peter Miller is not coming. No, sorry Paul Miller.
(e)
Peter Mi-
Ok, Paul Miller is not coming.
It's Paul Miller.
(f)
Peter Miller no sorry Paul Miller is not coming.
(g)
Pet- no sorry Paul Miller is not coming.
t

Figure 2: Types of Corrections / Clarifications

(6) a. *Jo . . . {wait/no/hang on/. . . } Jack is arriving tomorrow* (= I meant to say Jack, not Jo is arriving tomorrow)

b. *Jo . . . {wait/no/hang on/. . . } yeah Jo is arriving tomorrow* (= I did mean to say Jo is arriving tomorrow)

c. *Jo. . . {wait/no/hang on/. . . } Jo*? (= Did I say/mean 'Jo'?). . . {*wait/no/hang on/. . .* } *Jack is arriving tomorrow* (= I meant to say Jack, not Jo is arriving tomorrow)

d. *A: Jo . . . um, uh quit* (= The word I was looking for after 'Jo' was 'quit').

e. *A: Jo . . . um, uh B: quit?* (= Was the word you were looking for after 'Jo' 'quit'?).

Our task, then, is to develop a formal model that can capture the similarities exhibited by self-initiated within-utterance repair and other-initiated cross-utterance repair, without neglecting the important characteristics that differentiate them. To this we turn now.

4 A Model of Other- and Self-Repair

4.1 KCRT: A Theory of Inter-Utterance, Other-Initiated Repair

For concreteness we take as our starting point the theory of CRification developed in (Ginzburg and Cooper, 2004; Purver, 2004; Ginzburg, (forthcoming)) (henceforth Kos CR Theory (KCRT)). This theory attempts to explain a.o. the coherence of CRs/corrections such as the following:[7]

(7) a. *A: Did Bo leave? B: Bo?* (= Who do you mean 'Bo'? or Are you asking if BO left?)

b. *A: Did Bo phone? B: You mean Mo.*

c. *A: Should we. . . B: leave?* (= Is 'leave' the word to be said *after* 'we'?)

The main features of KCRT are:

Initialization: Utterances are kept track of in a contextual attribute PENDING (cf. the G/DU bifurcation in PTT (Poesio and Traum, 1997).) in the immediate aftermath of the speech event. Given a presupposition that u is the most recent speech event and that T_u is a grammatical type that classifies u, a record of the form $\begin{bmatrix} \text{sit} = \text{u} \\ \text{sit-type} = \text{T}_u \end{bmatrix}$ of type LocProp (*locutionary proposition*), gets added to PENDING.

Contextual/phonological instantiation: In so far as A's information state IS_0 enables her to

[7] How to analyze examples like (7c) is actually only mentioned in passing in (Purver, 2004), given certain formal difficulties it involves, not least of which is parsing an incomplete utterance.

fully instantiate the contextual parameters specified in T_u, and $T_u.phon$ is uniquely specified, $\begin{bmatrix}\text{sit = u}\\ \text{sit-type} = \mathrm{T}_u\end{bmatrix}$ can trigger an illocutionary update of IS_0 (i.e. a new move is added to MOVES—an assertion, query etc.)

CR/Correction coherence: Failure to fully instantiate contextual parameters or recognize phonological types triggers CRification. This involves accommodation of questions into context by means of Clarification Context Update Rules (CCURs). Each CCUR specifies an accommodated MaxQUD built up from a sub-utterance u1 of the target utterance, the maximal element of PENDING, *MaxPending*. Common to all CCURs is a license to follow up MaxPending with an utterance whose qud-update is *co-propositional* with MaxQud[8]: either a CR which differs from MaxQud at most in terms of its domain, or a correction—a proposition that instantiates MaxQud. The CCURs differ primarily in the question whose accommodation into QUD they give rise to. (8) is a simplified formulation of one CCUR, (9)-(11) provide a specification of the MaxQud instantiation of other CCURs:

(8) Parameter identification:

$$\text{Input: } \begin{bmatrix}\text{Spkr : Ind}\\ \text{MaxPending : LocProp}\\ \text{u0} \in \text{MaxPending.sit.constits}\end{bmatrix}$$

$$\text{Output: } \begin{bmatrix}\text{MaxQUD = What did spkr mean by u0?}\\ \text{LatestMove : LocProp}\\ \text{c1: CoProp(LatestMove.cont,MaxQUD)}\end{bmatrix}$$

(9) Parameter focussing: raises as MaxQud λxMaxPending.content(u1.content $\mapsto$x)

(10) Utterance repetition: raises as MaxQud λxUtter(A,u1,x) (*What did A utter in u1*? "What did you say?")

(11) Utterance prediction: raises as MaxQud λxUtterAfter(A,u1,x) (*What will A utter after u1*? "What were you going to say?")

Answers: Accepting an answer to a CR/correction gives rise to an modified MaxPending via **Contextual/phonological instantiation**: (in the case of content–related CRs (corrections): the contextual assignment of u is extended (replaced by a substitute); in the case of phonological CRs this applies to $T_u.phon$.)

Speaker/hearer asymmetry: Speakers cannot self-CR because their own utterance is downdated from PENDING following successful contextual parameter instantiation (which always applies to a speaker's own utterance.). Hence, the different contextual possibilities, exemplified in (4) and (5).

CR accommodation: If A utters u and B follows up with a CR/correction, A accommodates the MaxQud B accommodated and $\begin{bmatrix}\text{sit = u}\\ \text{sit-type} = \mathrm{T}_u\end{bmatrix}$ becomes MaxPending.

4.2 Extending KCRT to Self-Initiated Mid-Utterance Repair

How do we extend this model to mid-utterance self and other correction? As things stand, there are two things that prevent KCRT from accounting for self-repair: (1) all CR/corrections are forced to occur after complete utterances, and (2) CR/corrections can only be posed by others (given that the speaker downdates PENDING immediately). Let us take up each of these issues in turn.

The first move we make is indeed to extend PENDING to incorporate utterances that are *in progress*, and hence, incompletely specified semantically and phonologically. Conceptually this is a natural step to make. Formally and methodologically this is a rather big step, as it presupposes the use of a grammar which can associate types word by word (or minimally constituent by constituent), as e.g. in Categorial Grammar, Dynamic Syntax, (Steedman, 2000; Kempson et al., 2000). It raises a variety of issues with which we cannot deal in the current paper: monotonicity, nature of incremental denotations etc.

For our current purposes, the decisions we need to make can be stated independently of the specific grammatical formalism used, modulo the fact that as in the KCRT work, we need to assume that grammatical types specify a feature/label/field CONSTITS which keeps track of all not just immediate constituents of a given speech event (gram-

[8] A query q updates QUD with q, whereas an assertion p updates QUD with p?. Two questions q0 and q1 are copropositional if there exists a record r such that q0 (r) = q1 (r). This means that, modulo their domain, the questions involve similar answers.

matical type). The main assumptions we are forced to make concern where pending instantiation and contextual instantiation occurs, and more generally, the testing of the fit between the speech events and the types assigned to them. We assume that this takes place incrementally, say word by word.

The incrementalization of PENDING has good consequences, as well as certain seemingly undesirable ones. On the positive side, since PENDING now includes also incomplete utterances, we can now account also for CRs/other corrections that occur mid-utterance, dispreferred as they might be (Schegloff et al., 1977). One such corpus example is (12a). The constructed (12b) shows that in such contexts the same ambiguities are maintained as in cross-utterance cases exemplified above:

(12) a. *A: There are subsistance farmers that . . . B: There are what?* (attested example from the Potsdam Pentomino Corpus)

b. *A: Did Bo. . . (no pause) B: Bo?* (= Who do you mean 'Bo'? or Are you asking something about BO?) *A: I mean Mo/Yeah, Mo's partner.*

On the other hand, without saying more, it will overgenerate in precisely the way we were trying to avoid, given (4) and (5). We can block this via a route any dialogue theory has to go through in any case: moves such as acceptances involve obligatory turn change. For this reason KCRT already keeps track of speaker/addressee roles, while underspecifying these where the turn is up for grabs (as e.g. following the posing of a query.). So the CCURs we specified above will now carry information that ensures that the various interpolated utterances do indeed involve a turn change.

This in turn means that **simply enlarging the scope of what goes into PENDING has not offered a route to characterize the potential for mid-utterance self correction.** But this is probably inevitable: while there may be some cases such as (12) involving *other* participants, *self*-correction in mid-utterance (and elsewhere) involves, as we discussed earlier, the presence of an editing phrase (EditP) (encompassing also extended silences.). What we need to do, therefore, is to provide a means for licensing EditPs. This is simple to do: all we need to say is that an EditP can be interpolated essentially at any point, or more precisely, at any point where PENDING is non-empty. (13) is an informal such specification. It enforces turn continuity and the non-inclusion of the EditP in PENDING:

(13) Edit Move Update Rule:

$$\text{Input: } \begin{bmatrix} \text{Spkr : Ind} \\ \text{MaxPending : LocProp} \end{bmatrix}$$

$$\text{Output: } \begin{bmatrix} \text{Spkr = Input.spkr : Ind} \\ \text{Pending = Input.MaxPending: LocProp} \\ \text{LatestMove = Edit(Spkr,MaxPending)} \end{bmatrix}$$

The output state this brings us to is a state where PENDING contains repairable material and the LatestMove is an EditP. Now we can specify coherent Self/Other corrections in a manner akin, though not identical to (8)-(11). We will assume the following as a tentative characterization, though clearly it is not exhaustive:

(14) . . . u0. . . EditP u1 (= Spkr meant to utter u1)

(15) . . . u0. . . EditP u0'? (= Did Spkr mean to utter u0?)

(16) A: . . . u0. . . $\{um, uh\}$ u1 (= Spkr meant u1 to be the next word after u0)

We sketch here only a rule that will capture (14) and (15). The URs in (17) take as input a state where the LatestMove is an EditP and specify a new state in which the MaxQUD is *What did spkr mean to utter at u0?* and where the new utterance has to be an instantiation of MaxQud (propositional or polar question):

(17) Utterance identification:

$$\text{Input: } \begin{bmatrix} \text{Spkr : Ind} \\ \text{MaxPending : LocProp} \\ \text{LatestMove = EditP(Spkr,MaxPending)} \\ \text{u0} \in \text{MaxPending.sit.constits} \end{bmatrix}$$

$$\text{Output: } \begin{bmatrix} \text{MaxQUD = What did spkr mean to say at u0?} \\ \text{LatestMove : LocProp} \\ \text{c2: InstPropQ(LatestMove.cont,MaxQUD)} \end{bmatrix}$$

With this machinery in hand, we can now consider some examples:

1. Self-correction mid-utterance:

(18) *A: Peter. no Paul quit.*

1.a After utterance of 'Peter': in A's FACTS (shared assumptions etc—whatever underwrites presuppositions) the presuppositions that the most recent speech event is u0 ('Peter'), classified by a type T_{u0}; PENDING gets updated with the following record:

$$\begin{bmatrix} \text{sit = u0;} \\ \text{Sit-Type = ‘Utterance whose first word} \\ \text{is Peter; involves reference to p...’} \end{bmatrix}$$

1.b This allows for an EditP to be interpolated: LatestMove = Edit(A,MaxPending).

1.c This allows for utterance identification: MaxQUD = What did spkr mean to say at u0?; LatestMove: Assert(A, MeanUtter(A,‘Paul’))

1.d Accepting this gives rise to an application of Contextual/phonological instantiation: PENDING is modified to the following record:

$$\begin{bmatrix} \text{sit = u1;} \\ \text{Sit-Type = ‘Utterance whose first word} \\ \text{is Paul; involves reference to p’...’} \end{bmatrix}$$

1.e Note: the utterance u0 is still in the information state, though not as a compnent of PENDING—PENDING was originally initialized due to the presence in FACTS of the proposition that the most recent speech event is u0 (‘Peter’), classified by a type T_{u0}. Hence, anaphoric possibilities to this utterance are not eliminated.

2. Self-correction after utterance:

(19) *A: Peter quit. Hang on. Not Peter, I meant Paul.*

Same procedure as in **1.**, initiated with the completed utterance as MaxPending.

3. Other-correction, indirect:

(20) *A: (1) Peter is not coming.*
B: Peter? (in ‘indirect correction’ reading)
A: Oh, sorry, I meant Paul.

In consequence of B’s utterance A applies CR accommodation, which makes *What did A mean by ‘Peter’* MaxQud and (1) MaxPending. Applying Contextual/phonological instantiation after A’s correction leads to a modification in (1).

4. Other-correction, direct:

(21) *A: (a) Peter is not coming.*
B: (b) No, (c) Peter is, Paul isn’t.

This is simply a disagreement at the illocutionary level: A’s assertion pushes ?Coming(peter) to MaxQud but not to FACTS, giving rise to the discussion which B initiates. If A accepts B’s assertion (c) will be added to FACTS, whereas ?Coming(peter) gets downdated from QUD.

5 Conclusions

In this paper we have related self- and other-initiated repair. We have argued, following a long but unformalized tradition in Conversation Analysis, that the two processes bear significant similarities: a problem is detected with an utterance, this is signalled, and then the problem is addressed and repaired, leaving the incriminated material with a special status, but within the discourse context. We provide a unified account: a single repository, PENDING carries CR/correct-able material within and across utterances. Consequently, a single set of rules regulate the up- and downdating of PENDING, as well as the modification of its elements by answers to CRs or corrections, regardless of whether the utterances that are in progress or completed. Different rules trigger within and cross-utterance CRs/corrections, but that is as should be, as the form and content of these differ, as we have shown.

Acknowledgements This work was carried out while the first author was a senior visitor at the University of Potsdam within the DEAWU project. The work was supported by the EU (Marie Curie Programme) and Deutsche Forschungsgemeinschaft (Emmy Noether Programme).

References

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

J. Ginzburg. forthcoming *Semantics and Interaction in Dialogue*. CSLI Publications, Stanford: California. Draft chapters available from http://www.dcs.kcl.ac.uk/staff/ginzburg.

J. Ginzburg and R. Cooper. 2004. Clarification, ellipsis, and the nature of contextual updates. *Linguistics and Philosophy*, 27(3):297–366.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and devlopment. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, pages 517–520, San Francisco, USA.

P. A. Heeman and J. F. Allen. 1999. Speech repairs, intonational phrases and discourse markers: Modeling speakers’ utternaces in spoken dialogue. *Computational Linguistics*, 25(4):527–571.

R. Kempson, W. Meyer-Viol, and D. Gabbay. 2000. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell, Oxford.

E. F. Lau and F. Ferreira. 2005. Lingering Effects of Disfluent Material on Comprehension of Garden Path Sentences *Language and Cognitive Processes*, 20(5):633–666

W. J. Levelt. 1983. Monitoring and self-repair in speech. *Cognition*, 14(4):41–104.

M. Meeter and A. Taylor. 1995. *Dysfluency Annotation Stylebook for the Switchboard Corpus*. University of Pennsylvania.

D. McKelvie. 1998. The syntax of disfluency in spontaneous spoken language. HCRC Research Paper HCRC/RP-95, Human Communication Research Centre, Edinburgh.

M. Poesio and D. Traum. 1997. Conversational actions and discourse situations. *Computational Intelligence*, 13:309–347.

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

E. A. Schegloff, G. Jefferson, and H. Sacks. 1977. The preference for self-correction in the organisation of repair in conversation. *Language*, 53(2):361–382.

E. E. Shriberg. 1994. *Preliminaries to a theory of speech disfluencies*. Ph.D. thesis, University of California at Berkeley, Berkeley, USA.

M. Steedman. 2000. *The Syntactic Process*. Linguistic Inquiry Monographs. MIT Press, Cambridge.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 65–72.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Clarification Requests: An Incremental Account

Ruth Kempson
King's College London
ruth.kempson@kcl.ac.uk

Andrew Gargett
King's College London
andrew.gargett@kcl.ac.uk

Eleni Gregoromichelaki
King's College London
eleni.greg@blueyonder.co.uk

Abstract

In dialogue models, fragment clarification requests (CRs) characteristically involve pre-processing, lifting the fragment to sentential level or coercing the context (Ginzburg and Cooper, 2004; Purver, 2004). This paper introduces an incremental account of CRs using the Dynamic Syntax (DS) dialogue model (Purver et al., 2006) that allows CRs as interruptions with no structure-specific stipulations. Generation of CRs is licensed at any point in the tree growth process which constitutes the build-up of utterance interpretation. The content attributed to CRs in context is one step more advanced than what has been achieved by the (interrupted) parse, either querying lexical content, checking/querying means of identification in context, or checking/querying resulting content (in the last of these, update may be trivial). Fragment responses (FRs) may reconstruct the apparent source of difficulty from the CR parse providing/confirming update from that reconstructed partial tree. However, the FR may constitute a trivial update of the clarifier's own context (the latter being the tree-representation of their initiating utterance), as the CR has been equally parsed via trivial context-update. All ambiguities arise from interaction of lexical specification, available partial structure as context, and available means of update: no irreducible ambiguity is required.

1 Introduction

Accounts of clarifications presume, following Ginzburg and Cooper (2004), that clarification-request fragments (CR) bifurcate according to whether what is queried concerns contentious content (the "clausal reading") or problematic identification of the meaning of the word used ("constituent reading"), the latter taken as a distinct "anaphoric utterance" use, with both being assigned a propositional-type construal. However, not only can it be shown that propositional-type analyses are not necessary in accounting for such ellipsis construals, as we shall see in due course, but it is also well-known that clarification requests and their fragment response can be made incrementally at a sub-sentential level:

(1) A. They X-rayed me, and took a urine
sample, took a blood sample.
A: Er, the doctor
B: Chorlton?
A: Chorlton, mhm, he examined me, erm,
he, he said now they were on about a slight
[shadow] on my heart.

Furthermore, there is a broad range of readings associated with such fragments which do not seem to fall easily into two such clear-cut categories. To illustrate, we set out the following possible modes of clarifying the subject of a statement using the repeated fragment (CR) with its equally fragmentary reply (FR), and outline some of the different possible CR construals when the time-linear dimension of the parse is taken into account:

(2)		A (female):	Bill left.
	(i)	B (male):	Bill?
	(ii)	B:	"Bill"?
		A:	Bill (Smith).

Case (i) of B's responses is a CR that can be paraphrased in terms of the whole of A's original utterance, in other words, as *Bill left?*. One might distinguish three reasons to justify the utterance of such a CR: (a) the entire utterance has been understood, but the CR conveys doubt of the involvement of the individual referred to; (b) although who is intended has in principle been identified, confirmation is still requested for certainty; (c) the meaning of the word is understood, hence the sentence successfully understood *qua* sentence, but the query is a request for provision of information to identify who is being referred to in the face of lacking this information. B's response (ii), as annotated, might seem to be construed as making a meta-linguistic response, and there are arguably three bases which suggest this form of construal: (a) the word *Bill* has been parsed, but uncertainty as to who A is talking about has led to B abandoning the parse at that juncture without establishing a full understanding of the sentence; (b) B fears he has misheard, and (on the basis of some word segment he has heard) is guessing what was said (e.g. here B might say *Bill* and be right, or *Jill* and be wrong), and (c) where B is explicitly asking for a repeat of the information provided by that word. There are thus a considerable number of different ways of grounding CR uses.

Three features of CRs provide clues as to how best to model them. First, they repeat specific material from the context. Unlike standard questions, this type of clarification is not about requesting new information from interlocutors (as with WH-questions), but focuses on repeating items from (the immediate) context. Second, their brevity opens up a range of possible interpretations, not always distinguishable. Third, they have a characteristic intonation, whose function is to indicate some non-canonical mode of interpretation in response to the immediate context (Rodriguez and Schlangen, 2004).

This paper presents the claim that the Dynamic Syntax (DS) model of dialogue (Purver et al., 2006) extends seamlessly to these phenomena. The account of clarificatory requests (CR) and fragment replies (FR) allows incremental request/provision of clarification at arbitrary points in the dialogue, while retaining a unitary characterisation of the lexical input. There is no need for coercion operations in order to resolve the fragment (Ginzburg and Cooper, 2004; Purver, 2004; Purver, 2006). We shall also argue that the distinction between clausal and constituent CRs emerges as a consequence of clarification being possible for all licensed tree transitions, including those involving the update provided by the word itself, so there is no recourse to stipulated input ambiguity between clausal and constituent CR's. The analysis of CR's and FR's furthermore fits directly within an overall account of ellipsis that construal of fragments is determined by structures/formulae/actions that context directly provides (Purver et al., 2006; Cann et al., 2007).

2 Previous Literature

As a form of nonsentential utterance (NSU), CRs have typically been modelled through preprocessing of some kind. Approaches adopt either a syntactic approach lifting them to sentence level (assuming missing information is "hidden"), or a semantic one, raising the information presented by some previous sentence so this can combine with the content of the fragment to yield back a propositional content (for representative papers see Elugardo and Stainton, 2005). A third approach associated with Ginzburg and colleagues (eg Ginzburg and Sag, 2000; Purver, 2004; Ginzburg and Cooper, 2004; Fernández-Rovira, 2006) both lifts the fragment to a clausal level and processes contextual information (which they term *context coercion*) (Purver, 2006).

This last approach has been described as incremental in involving phonological, syntactic, and semantic projection of subparts of complex signs in parallel as information becomes available (Ginzburg and Cooper, 2004). However, it is also desirable that computational accounts meet a notion of incrementality in which projection of structure/interpretation follows as closely as possible word-by-word processing with progressive interaction between linguistic and contextual information for which there is psycholinguistic evidence (see Pickering and Garrod among others); and the DS model of dialogue (Purver et al., 2006) purports to match this, as part of meeting the Pickering and Garrod challenge that formalisms for language modelling should be evaluated by how good a basis they provide for reflecting patterns that occur in conversational dialogue.

3 Dynamic Syntax: Background

Dynamic Syntax (DS) is a parsing-based approach to linguistic modelling in which syntax is defined as the progressive projection of semantic representations from the words taken in left-to-right sequence. Such representations take the form of decorated (linked) binary branching trees representing predicate-argument structures, with each node decorated with a sub-term of some propositional formula. The interpretation process is defined as goal-directed growth along various dimensions of tree decoration: type and formula decorations ($Ty(X)$, $Fo(Y)$), tree-node identification ($Tn(Z)$), and tree-relations (see below). Formula decorations are lambda terms of the epsilon calculus, with all quantified terms of type e, their restrictor being part of the term.[1]

The central tree-growth process of the model is defined in terms of the procedures whereby such structures are built up; taking the form of general structure-building principles (*computational actions*) and specific actions induced by parsing particular lexical items (*lexical actions*). The core of the formal language is the modal tree logic LOFT, which defines modal operators $\langle\downarrow\rangle$, $\langle\uparrow\rangle$, which are interpreted as indicating daughter and mother relations, respectively, $\langle\uparrow_*\rangle$, $\langle\downarrow_*\rangle$ operators characterizing *dominate* and *be dominated by*, and two additional operators $\langle L\rangle$, $\langle L^{-1}\rangle$ to license paired *linked* trees. Tree nodes can then be identified from the rootnode $Tn(0)$ in terms such as $\langle\uparrow\rangle Tn(0)$, $\langle\uparrow_*\rangle Tn(0)$, etc. The actions defined using this language are transition functions between intermediate states, which monotonically extend tree structures and node decorations. The concept of *requirement* is central to this process, $?X$ representing the imposition of a goal to establish X, for any label X. Requirements may thus take the form $?Ty(t)$, $?Ty(e)$, $?Ty(e \rightarrow t)$, $?\langle\downarrow\rangle Ty(e \rightarrow t)$, $?\exists x Fo(x)$, $?\exists x Tn(x)$, etc.

All aspects of underspecification have an associated requirement for update. Pronouns illustrate formula underspecification, the pronoun *he* being assigned lexical actions from a trigger $?Ty(e)$ that projects a metavariable $Fo(\mathbf{U}_{Male(\mathbf{U})})$ of $Ty(e)$ with requirement $?\exists x Fo(x)$ (also a case requirement); and such metavariables are replaced by a Substitution process from a term available in context. We assume that the restriction $Male(\mathbf{U})$ would be specified as resulting from an action to construct a LINK transition to a tree of topnode to be decorated as $Male(\mathbf{U})$ as part of the actions encoded by the pronoun *he* (the mechanism of constructing a LINK relation being the means of constructing paired trees to be evaluated as compound forms of conjunction: Cann et al., 2005).

The process is thus essentially representational: the resolution of pronoun construal is established as part of the construction process. We propose that names too project a metavariable, eg *Bill* projecting a metavariable which we annotate as $Fo(\mathbf{U}_{Bill'(\mathbf{U})})$, with instruction to construct a LINK transition to a linked tree of topnode $Ty(t)$ decorated with the formula value $Bill'(\mathbf{U})$, characterising the predicate 'being named Bill', this constituting a constraint on the logical constant to be assigned as construal of the use of that name in the particular context.[2] We shall represent such logical constants, m_{21}, m_{22} etc, as having an attendant predicate attribute, eg $(m_{21,Bill'(m21)})$, but these are short-hand for the projection of such a pair of linked trees, one containing an argument node decorated with a formula (m_{21}) of type e, linked to a tree with topnode decorated with the formula $Bill'(m_{21})$.

The construction of structurally underspecified relations is also licensed (displayed in trees as a dashed line), with construction of nodes through an operation **Adjunction* licensing construction from a node $Tn(a)$ of a node described only as $\langle\uparrow_*$ $\rangle Tn(a)$, an underspecification which is resolved, if introduced early on in a parse, only at a later point in the parse, when this characterisation can be satisfied by some introduced node of appropriate type. A variant, *Late*Adjunction*, applies to an initiating node of a given type to induce a dominated node requiring the same type, which with subsequent parse provides a basis for update to that initiating node, hence to some interim metavariable decorating it: Cann et al. (2005) analysed expletive pronouns in these terms.

Since, in any parse sequence, there may and characteristically will be more than one update possibility, a parse state P is defined as a set of triples $\langle T, W, A\rangle$, where: T is a (possibly partial)

[1] These take the form of variable-binder, variable of type e, and restrictor. Composite restrictors can be constructed through the building of linked trees, the resulting propositional content then by a step of LINK-evaluation, taken as an enrichment of the restrictor-specification (Kempson et al., 2001).

[2] Such an analysis suggests presuppositions in general involve constructing linked trees (Cann et al., 2005, ch.8).

tree; W is the associated sequence of words; A is the associated sequence of lexical and computational actions. Context is then defined in similar terms. At any point in the parsing process, the context $\mathcal{C}$ for a particular partial tree T in the set P can be taken to consist of: a set of triples $P' = \{\ldots, \langle T_i, W_i, A_i \rangle, \ldots\}$ resulting from the previous sentence(s); and the triple $\langle T, W, A \rangle$ itself.[3] Wellformedness is then definable as the availability of at least one sequence of transitions from some partial tree-specification as output to some complete tree with topnode decorated with a formula of type t having used all the words in sequence and with no outstanding requirements, a characterisation which Cann et al. (2007) extend to define a concept of context-dependent wellformedness.

In Purver et al. (2006) generation is defined to follow the parsing dynamics, this being the core mechanism, but it too is goal-directed: speakers have a goal tree representing what they wish to communicate, and each licensed step of the update transition defined by the core formalism constitutes the grounding for some possible generation step subject to a requirement of a subsumption relation between the constructed parse tree and the goal tree, in the sense of allowing a successful derivation from the parse tree as updated to the goal tree. Incremental (word-by-word) parsing, and lexicon search for words which provide appropriate tree-update relative to this goal tree enables speakers to produce the associated natural language string (see Purver et al., 2006). A generator state G is thus a pair (T_G, X) of a goal tree T_G and a set X of pairs (S, P), where: S is a candidate partial string; P is the associated parser state (a set of $\langle T, W, A \rangle$ triples). Search for appropriate words is said to be made from context wherever possible, reducing the production task.

Ellipsis, in both parsing and generation equally, involves use of context in a number of different ways. Strict readings of (VP) ellipsis involve taking some formula value as the value of the metavariable supplied at the ellipsis site. Sloppy readings of such fragments reuse sequences of actions stored in context, leading to different information given their re-application relative to the partial tree provided by the construal of the fragment itself. Answers to questions involve using some structure in context as their point of departure, the answer expression providing the update to that structure to yield some propositional formula. In the generation of such ellipses, the same parse actions are subject to the added restriction that the update to the partial tree under construction subsume the goal tree. What integrates these accounts of different elliptical forms is that each makes direct use of some attribute of context, without any coercion of the context prior to such use, thereby dramatically reducing the parsing/production task, as full lexicon search is side-stepped.

4 Towards an Incremental Account of CRs

In the general case, parsing and generation are presumed to start from the Axiom, the initial one-node tree $?Ty(t)$ and reach some goal tree $Ty(t), Fo(\alpha)$ via an accumulated sequence of transitions across partial trees, but this restriction is not essential: both parse and generation tasks may start from arbitrarily rich partial trees and end at any richer partial tree (see Purver et al., 2006 for an account of split utterances that depends on this). It is these partial tree inputs and outputs which constitute the core of the CR account.

The general schema is as follows. We take questions overall to be an encoding of a explicit request for coordination with some other party with respect to input provided by the question form. There are two core cases: those where some particular (*wh*-marked) constituent is signalled as being the information to be provided by the answer;[4] and those where the request concerns some whole propositional unit (polar interrogatives), which may be marked by word order or often merely by intonation alone. However, there is also a whole range of cases where individual words, their intrinsic system-based meaning, or their particular context-established construal may constitute the request for explicit coordination. These are the CR cases – a fragment associated with an explicit coordination request. Given the DS account of dialogue, all such fragments are taken to be both understood or have their production licensed relative to whatever structure is provided in context, whether a partial tree representation, with pointer indicating where the emergent growth of some tree

[3] For simplicity, we shall generally take this to comprise the triple P' resulting from A's initial utterance, and any partial trees established in subsequent parsed fragments associated with clarification of aspects of P'.

[4] See Kempson et al., 2001, where it is argued that *wh* expressions encode specialised meta-variables.

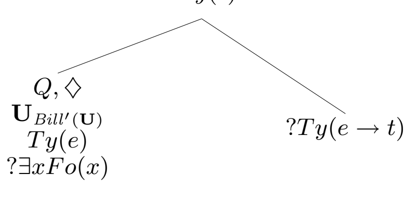
structure has got to, or a completed propositional tree representation. The encoding that this is a coordination request we take, at this juncture, simply to be a primitive Q annotation, to be removed by rule in the process of response.[5]

Informally, then, the dynamics of how such CR's or FR's come to be uttered is as follows. The idea is that the formal account should model very directly the dynamics of information processing – A, the initiator starts to say something, and clarification can be requested and provided at any point whenever B "gets stuck". What this will mean in DS terms is the construction of some partial pointed tree which will then constitute the context for B's interruption: the goal of the CR is then the request for provision of a value at exactly that point, with intonation playing the critical role of indicating where that is. The goal tree for such an interruption is characteristically just one point of update from the partial tree established at that achieved parse state, or may even be identical to it. A then may reply with the needed clarification, often also a fragment (FR), both being able to rely on re-use of actions from context to round out the interpretation intended. With uncertainty in the parse process in principle possible at any point in the parse sequence, requests for clarification may occur at any point in the parse-update process.

In the case of (2), this yields at least the following possibilities, each tree displaying the construction step immediately upon uttering the word *Bill*.

I: B may have failed to fully parse the first word but makes a stab at what it was, his goal tree being a structure constituting a request for confirmation of the provided name-based update (Q is taken to decorate the node indicated by intonation):

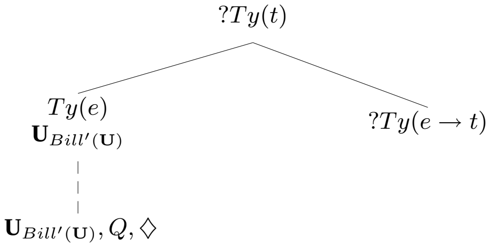
I



In this case, the goal tree set up contains a decoration as provided by the name but no identification of the individual in question. It is notable that if a word is even to be guessed at, it will induce tentative update of the partial tree, hence characterising even clarifications at the word-level as just one among a whole set of possible bases for clarification, without need of any concept of anaphoric utterance (contra Ginzburg and Cooper, 2004)

II: B may have successfully processed the first word but, not knowing who is being talked about, wishes to establish this before proceeding to the development of the predicate node (the analogue of incremental anaphora resolution). Such a request can be achieved by repeating the name, because a licensed parse step is to build an unfixed node (by *Late*Adjunction*) from the node already decorated with $\mathbf{U}_{\mathbf{Bill'(U)}}$ thereby providing an additional node decorated with $?Ty(e)$ which will license the update induced by the parse of the repeated name and lead via unification of the two nodes back to the parse tree which constituted the source of his request for clarification:

II

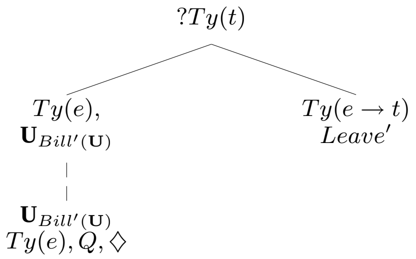

$?Ty(t)$
$Ty(e)$
$\mathbf{U}_{Bill'(\mathbf{U})}$
$?Ty(e \rightarrow t)$
$\mathbf{U}_{Bill'(\mathbf{U})}, Q, \Diamond$

In other words, the assumption of a goal tree and a (distinct) parse tree can be retained even in cases where some understood word is nevertheless being repeated.

III: B may have understood the first word, using it to establish a type value and a place-holding metavariable, without having been able to identify who is being talked about. Nonetheless, because the word itself is processed the parse can continue. The predicate value can then be established (which may help to identify the individual in question). Yet, in coming to build up a whole propositional content, B may still yet fail to identify who is being talked about and so need to repeat the word as before. This would be the analogue of expletive pronouns, for which a type value is assigned to the node in question early on in the interpretation process, but the formula value is established at a relatively late point:

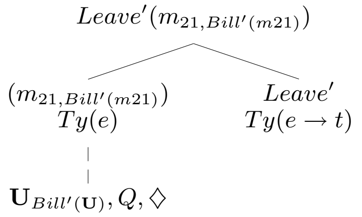
[5] This is clearly only an intermediate formulation, but the critical aspect is that it not be presented as itself in predicate-argument form in the representation, unless this is explicitly made clear through words whose content is to present such a request.

III


$?Ty(t)$
$Ty(e)$, $\mathbf{U}_{Bill'(\mathbf{U})}$
$Ty(e \rightarrow t)$ $Leave'$
$\mathbf{U}_{Bill'(\mathbf{U})}$ $Ty(e), Q, \diamondsuit$

IV: B may have fully understood what A has said but might wish to have this confirmed, and thus decides on a goal tree hopefully identical with that which he has just retrieved in parsing A's utterance. *Late*Adjunction* can be applied relative to this structure also (with pointer return to the subject node as in parsing the answer to questions), again enabling the parse actions associated with *Bill* to be added to the introduced node, albeit one which will turn out to be a trivial update:

IV


$Leave'(m_{21,Bill'(m21)})$
$(m_{21,Bill'(m21)})$ $Ty(e)$
$Leave'$ $Ty(e \rightarrow t)$
$\mathbf{U}_{Bill'(\mathbf{U})}, Q, \diamondsuit$

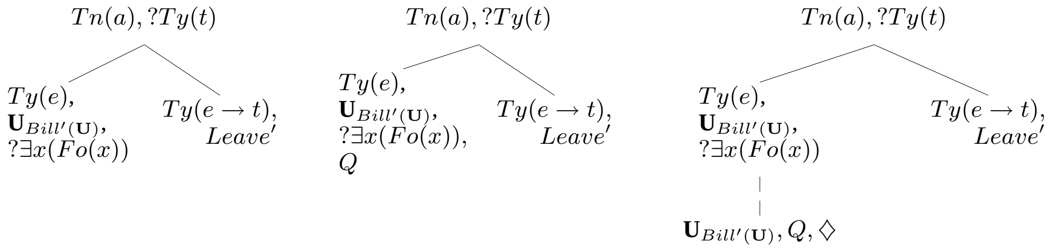
There are more possible interim goal trees on the basis of which B might seek clarification. In each case, the effect of adopting a goal tree which is some minimal (possibly null) enrichment from the partial tree established in the parse process (taken as context for the generation task) ensures that clarificatory fragments are assigned a context, parse tree and goal tree which are (all but) identical. This may seem to render the account trivial, with so little differentiation between context input to the generation task, goal output of the task, and parse-tree indicating the current update; but the near-identity is definitive of the clarification task. The goal tree in such cases is not some radical enrichment of the input parse tree: to the contrary, what is requested is either suggestions for or confirmation of some current putative tree-growth step, in order to proceed. With all these signals, in conjunction with the indicative intonation across the fragment in question, it follows that as long as A successfully parses B's fragment, she will have a basis for identifying the source requiring clarification, and hence the basis for a reply.

4.1 Detailing a CR/FR Exchange

In each such clarification exchange, there are altogether three turns, with each turn having a goal and parse tree for the speaker, and a parse tree for the hearer. Figures 1-2 detail trees for a simple request for clarification, where both *Bill* and *left* have been parsed, upon the assumption that B has parsed A's input and recovered a decoration for the subject-node but without identifying which Bill is being talked about. Figure 1 schematically represents the generation of B's CF. With the current parse tree as context and input to the generation task, and the goal of querying the update to that subject decoration, B can make use of *Late*Adjunction*, licensing the construction of a node decorated with $?Ty(e)$ in order to provide a vehicle for licensing the lexical actions of the word *Bill*, i.e. the update with metavariable $\mathbf{U}_{Bill'(\mathbf{U})}$ which would then license unification with the already decorated subject node, yielding back the partial tree which was his parse tree as context differing from it only in the decoration Q which constitutes the request. The focus here is on modelling CR as an interactive strategy for repairing potential misalignment (eg Pickering and Garrod, 2004). For interactive repair of the misalignment to occur, A and B must agree on the node for which clarification is requested. The question is: how does B signal to A where to start? Here is where repetition and intonation jointly determine (re-)positioning of the pointer for both parties.

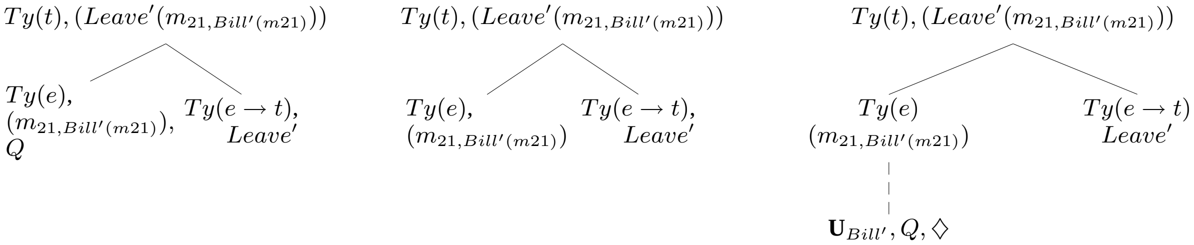
Figure 2 displays the update involved in A's fragment reply by licensing empty modification of her own initially established tree. On the tree under construction, the Q feature remains at the point of retrieval of the word *Bill*, but will be removed with identification of m_{21} as the value, hence falling within the subsumption constraint as defined. B, then, given the update provided by parsing A's FR, this time applies *Substitution* using the context provided (possibly by a more explicit utterance on A's part), and recovers $Leave'(m_{21,Bill'(m21)})$. The result, if so, is that A and B have re-aligned, and whatever failure in communication there had been in A's first utterance is successfully re-aligned.

On this account, we would expect that FR's can be made on the basis of a number of different assumptions. A may merely repeat the word used relative to her own context as in Fig.2. She may, however repeat the word *Bill* relative to a re-start,


$Tn(a), ?Ty(t)$
$Ty(e)$, $\mathbf{U}_{Bill'(\mathbf{U})}$, $?\exists x(Fo(x))$
$Ty(e \rightarrow t)$, $Leave'$
$Tn(a), ?Ty(t)$
$Ty(e)$, $\mathbf{U}_{Bill'(\mathbf{U})}$, $?\exists x(Fo(x))$, Q
$Ty(e \rightarrow t)$, $Leave'$
$Tn(a), ?Ty(t)$
$Ty(e)$, $\mathbf{U}_{Bill'(\mathbf{U})}$, $?\exists x(Fo(x))$
$Ty(e \rightarrow t)$, $Leave'$
$\mathbf{U}_{Bill'(\mathbf{U})}, Q, \diamondsuit$

(a) **B speaking**; context tree (left), goal tree (centre), tree under construction (right)

Figure 1: Clarification Query: Result of B's CR


$Ty(t), (Leave'(m_{21,Bill'(m21)}))$
$Ty(e)$, $(m_{21,Bill'(m21)})$, Q
$Ty(e \rightarrow t)$, $Leave'$
$Ty(t), (Leave'(m_{21,Bill'(m21)}))$
$Ty(e)$, $(m_{21,Bill'(m21)})$
$Ty(e \rightarrow t)$, $Leave'$
$Ty(t), (Leave'(m_{21,Bill'(m21)}))$
$Ty(e)$ $(m_{21,Bill'(m21)})$
$Ty(e \rightarrow t)$ $Leave'$
$\mathbf{U}_{Bill'}, Q, \diamondsuit$

(a) **A speaking**; context tree (left), goal tree (centre), tree under construction (right)

Figure 2: Clarification Response: Result of A's FR

introducing an unfixed node, then re-using actions from her own context as appropriate to yield a possibly different tree. This is in any case needed in cases of mistaken construal. In order to understand such a case, in which B utters say "Jill", A will have to parse that name as providing an utterance whose interpretation has to be constructed independently: to the contrary, merely to add decorations to her own tree as context would lead to it being discarded as inconsistent, thus preventing her from assigning B's fragment an interpretation. But with *Adjunction available, A can build an interpretation for Bill's utterance from a new goal of $?Ty(t)$ straightforwardly, taking it to provide a metavariable decorating an unfixed node, and from there A can nonetheless select a subset of actions to yield an understanding of Bill's clarification request based on the context provided by her own utterance. Her own reply might well thus also involve such a re-start introducing an unfixed node by *Adjunction following exactly the same pattern of actions as established by the immediately previous parse sequence used in processing the utterance of *Jill*. In such a case, with her utterance of *No* indicating her rejection of that established proposition as part of her own context, re-start is indeed a putative option, since she can use it nevertheless to build an unfixed node but also thereafter to recover the very same actions used in the processing of his utterance. However, given her rejection of the tree constructed from the parse of B's CR, as indicated by her utterance of *No*, she might also simply revert to using her own utterance as context with trivial update as in Fig.2. Either option is possible, clearly licensed by the DS perspective of setting out alternative strategies.

5 Discussion

As these displays have indicated, CR and FR generation can be made relative to the immediate parse context, which may be any partial tree along the transition from initial so-called Axiom state to some completed tree. Furthermore, the assumption, as here, that generation of FRs can (but need not) be on the basis of trivial modifications of some complete tree provides a basis for explaining why even young children can answer clarificatory questions without making any hypothesis as to the basis for clarification other than identifying the node in question.[6]

The added significance of this incremental approach to CR, is that no difference in principle needs to be stipulated to distinguish constituent and clausal types of CR/FR. Even the type of which Purver et al call *a reprise gap* falls into the same type of explanation, and is, on this ac-

[6] In principle the account extends to predicate words, if we make assumptions analogous to those made here for linguistic names, but this assumption needs extended justification to be developed elsewhere.

count, no more than the mechanism one would need in cases where the individual speaker repeats the word as in A's third utterance in (3) (Healey et al., 2003):

(3) A: Could I have some toast please?
B: some....?
A Toast.

All that needs to be assumed is that in order for B to utter "Some....", B will have already had to have parsed A's previous utterance via the construction of some metavariable of cn type as formula value. On this scenario, B will not however succeed in fully understanding what A has said without establishing the value for that metavariable. One way of getting help with this is to initiate the term-construction process again, harmlessly over-riding the earlier specification of $\lambda P.\epsilon.P$, but then signalling the leaving of the pointer at the $?Ty(cn)$ node, which A then provides. All that is needed to model this process is the assumption of a meta-variable for any type licensed, and the assumption that repeat actions may trivially update the determiner node (see Cann et al., 2005).

The analysis of CR's and FR's is thus general: for all apparently distinct subtypes, there is simply a cline of possible partial trees from outset parse state to completed tree, any one of which can constitute a point for clarification by generation of the appropriate word, with the goal of providing some minimal update to that interrupted parse sequence in order, once clarification is provided, to be able to proceed. This account has three advantages. First, the characterisation of the lexical content of the fragment remains constant across all construals of its uses, both fragmentary and non-fragmentary. Second, the phenomenon is explained in an integrated way across both CR and FR fragments. But, more generally than this, the mechanisms posited for this account of CR/FR fragments are none other than those posited for the account of ellipsis in general. Fragments in language are those cases in which their construal can be provided directly from the context, whether by taking whatever partial structure that context provides and building on it, or by taking formulae established in context, or by taking a sequence of actions recorded in context. Clarificatory fragments are those where both input and output to the local parsing/production process may be a partial structure. The only constraint put on such a process is that use of context in language construal has to be licensed by the form of input: and in the case of clarificatory fragments, it is precisely such a license, which intonation provides, indicating both the need to use context for construal and the fact that such construal will be essentially local, particular to the sequence of expressions so picked out.

References

R Cann, R Kempson, and L Marten. 2005. *The Dynamics of Language: An Introduction*. Elsevier.

R Cann, R Kempson, and M Purver. 2007. Context and wellformedness: the dynamics of ellipsis. *Research on Language and Computation*, 5.

R Elugardo and R Stainton, editors. 2005. *Ellipsis and Non-Sentential Speech*. Springer.

R Fernández-Rovira. 2006. *Non-Sentential Utterances in Dialogue: Classification, Resolution and Use*. Ph.d., Kings College London.

J Ginzburg and R Cooper. 2004. Clarification, ellipsis, and the nature of contextual updates. *Linguistics and Philosophy*, 27:297–365.

J Ginzburg and I Sag. 2000. *Interrogative Investigations*. CSLI Publications.

P Healey, M Purver, J King, J Ginzburg, and G Mills. 2003. Experimenting with clarification in dialogue. In *Proceedings of the 25th Annual Meeting of the Cognitive Science Society (CogSci 2003)*.

R Kempson, W Meyer-Viol, and D Gabbay. 2001. *Dynamic Syntax: The Flow of Language Understanding*. Blackwell.

M J Pickering and S Garrod. 2004. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–226.

M Purver, R Cann, and R Kempson. 2006. Grammars as parsers: Meeting the dialogue challenge. *Research on Language and Computation*, 4(2–3):289–326.

M Purver. 2004. *The Theory and Use of Clarification Requests in Dialogue*. Ph.d., Kings College London.

M Purver. 2006. Clarie: Handling clarification requests in a dialogue system. *Research on Language and Computation*, 4(2–3):259–288.

K J Rodriguez and D Schlangen. 2004. Form, intonation and function of clarification requests in german task oriented spoken dialogues. In *Proceedings of Catalog '04 (The 8th Workshop on the Semantics and Pragmatics of Dialogue, SemDial04)*.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, page 73.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Embodied Communication with a Virtual Human

Ipke Wachsmuth
Faculty of Technology & Center for Interdisciplinary Research (ZiF)
University of Bielefeld
ipke@techfak.uni-bielefeld.de

The aim of the ZiF research year on Embodied Communication (2005/2006) has been to launch and explore a new integrated and interdisciplinary perspective that accounts for the crucial role of the body in communication. The core claim of the Embodied Communication Perspective is that human communication involves parallel and highly interactive couplings between communication partners. These couplings range from low-level systems for performing and understanding instrumental actions, like the mirror system, to higher systems that interpret symbols in a cultural context. Going beyond the traditional engineering model of signal transmission, the Embodied Communication Perspective envisions a multi-modal, multi- level, dynamic model of communication. Rather than "exchanging meaning" in back-and-forth messaging, contributors co-construct meaning interactively, using all information available about the other's body and its relation to the environment. This perspective hence provides a novel framework for the study of gesture and forms of nonlinguistic interaction in multimodal dialogue and face-to-face conversation.

A particular goal of the research year on Embodied Communication has been to explore how the modeling of communication with artificial agents can advance our understanding of key aspects of cognition, embodiment, and cognitive processes in communication. Creating an artificial system that reproduces certain aspects of a natural system can help us understand the internal mechanisms that have led to particular effects. Virtual humans, i.e. computer-generated entities that look and act like people and engage in conversation and collaborative tasks in simulated environments, have become prominent in the study of communication. The idea of virtual humans acknowledges that natural communication is largely social and envisions future computer systems that are social actors rather than tools. Taking the virtual human "Max" as an example, the talk will outline some ideas how virtual humans can provide explanatory models in the form of behavior and process simulations and how they can help identify primitives and central mechanisms of embodied communication from a machine modeling perspective.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 75–82.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Annotating Continuous Understanding in a Multimodal Dialogue Corpus

Carlos Gómez Gallo[†], Gregory Aist[°], James Allen[†], William de Beaumont[⋆]
Sergio Coria[‡], Whitney Gegg-Harrison[◇], Joana Paulo Pardal[▽], Mary Swift[†]

Department of Computer Science, University of Rochester, Rochester, NY, USA[†]
Department of Brain and Cognitive Science, University of Rochester, Rochester, NY, USA[◇]
Institute for Human and Machine Cognition, Pensacola, FL, USA[⋆]
Computer Science and Engineering Department, Arizona State University, Tempe, AZ, USA[°]
Spoken Language Systems Lab, Lisbon, Portugal[▽]
Universidad Autónoma de México, Mexico City[‡]
cgomez@cs.rochester.edu

Abstract

We describe an annotation scheme aimed at capturing continuous understanding behavior in a multimodal dialogue corpus involving referential description tasks. By using multilayer annotation at the word level as opposed to sentence level, we can better understand the role of continuous understanding in dialogue. To this end, we annotate referring expressions, spatial relations, and speech acts at the earliest word that clarifies the speaker's intentions. Word-level annotation allows us to trace how referential expressions and actions are understood incrementally. Our corpus has intertwined language and actions which help identify the relationships between language usage, intention recognition, and contextual changes which in turn can be used to develop conversational agents that understand language in a continuous manner.

1 Introduction

In this paper we describe an annotation scheme aimed at capturing continuous understanding interaction in the Fruit Carts corpus (Aist et al., 2006). This corpus is a collection of multimodal dialogue interaction between two humans, where the first (the speaker) gives spoken language instructions to the second (the actor), who responds by manipulating objects in a graphical interface. The Fruit Carts domain was designed to elicit referring expressions from the speaker that are ambiguous in various ways, including prepositional phrase attachment and definiteness. The point at which the actor resolves the ambiguity can be observed through their actions in response to the spoken instructions. While the long-term goal of this corpus collection is to model incremental language processing in a spoken dialogue system, in this paper we concentrate on the highly interactive nature of the human dialogue in the corpus and how to represent it in an annotation scheme.

Previous research in psycholinguistics has shown that continuous understanding plays a major role in language understanding by humans e.g., (Tanenhaus et al., 1995; Altmann and Kamide, 1999; Traxler et al., 1997). Various researchers have proposed software methods for continuous understanding of natural language adapting a wide variety of techniques including finite state machines (Ait-Mokhtar and Chanod, 1997), perceptrons (Collins and Roark, 2004), neural networks (Jain and Waibel, 1990), categorial grammar (Milward, 1992), tree-adjoining grammar (Poller, 1994), and chart parsing (Wiren, 1989). Recently, dialogue agent architectures have been improved by different strategies that adhere to continuous understanding processing (Stoness et al., 2004; Aist et al., 2006). Therefore the work we present here will be a great help to understanding relationships between language and action, and the further development of dialogue agents.

Our annotation scheme for these interactions is centered around the idea of marking the roles, referential expressions, spatial relations and actions in the speaker's speech acts at the word level, as soon as they can be unambiguously identified. This contrasts with traditional utterance-level annotation, since our scheme requires us to break acts down into smaller constituents labeled at the word level.

We are basing our scheme on well developed speech act tagging hierarchies such as DAMSL (Core and Allen, 1997) and DIME-DAMSL

(Pineda et al., 2006). There is a limited amount of previous work related to the current paper. One example is (Reitter and Stede, 2003) which discusses markup allowing for underspecification of the meaning of contributions, but the work in their paper was done at a sentence-by-sentence level or higher (vs. at a word-by-word level in the current paper.) Some authors use the term *incremental annotation* to refer to the human-computer interaction process of successively annotating the same text with additional details (Molla, 2001), (van Halteren, 1998). This process is related to our work in that not all of the text is annotated at the same time. They focus on multiple passes over the same text, while we focus on a left-to-right continuous annotation done (in principle) in a single pass.

2 The Data

The Fruit Carts corpus consists of digital videos of 104 dialogues. Each of the 13 participants, recruited from the university community, directed the human actor in 8 different referential description tasks. Each of these task scenarios ranged from 4 to 8 minutes in duration. The number of utterances in each scenario ranges from 20 to more than 100. There are approximately 4000 utterances total in the corpus, with an average length of 11 words per utterance.

The corpus experiments involve referential description tasks in which the speaker is given a map showing a specific configuration of fruits and geometric shapes in different regions (see map on upper middle panel in Figure 1). The speaker's task is to instruct the actor to reorganize the objects so the final state of the world matches the map first given. The speaker gives spontaneous spoken instructions to the actor on how to go about manipulating the objects. The actor responds to the instructions by moving the objects, but does not speak. As a result the corpus captures a two way human-human dialogue. Thus we have a complex interaction of language and real world actions through a visual and auditory interface.

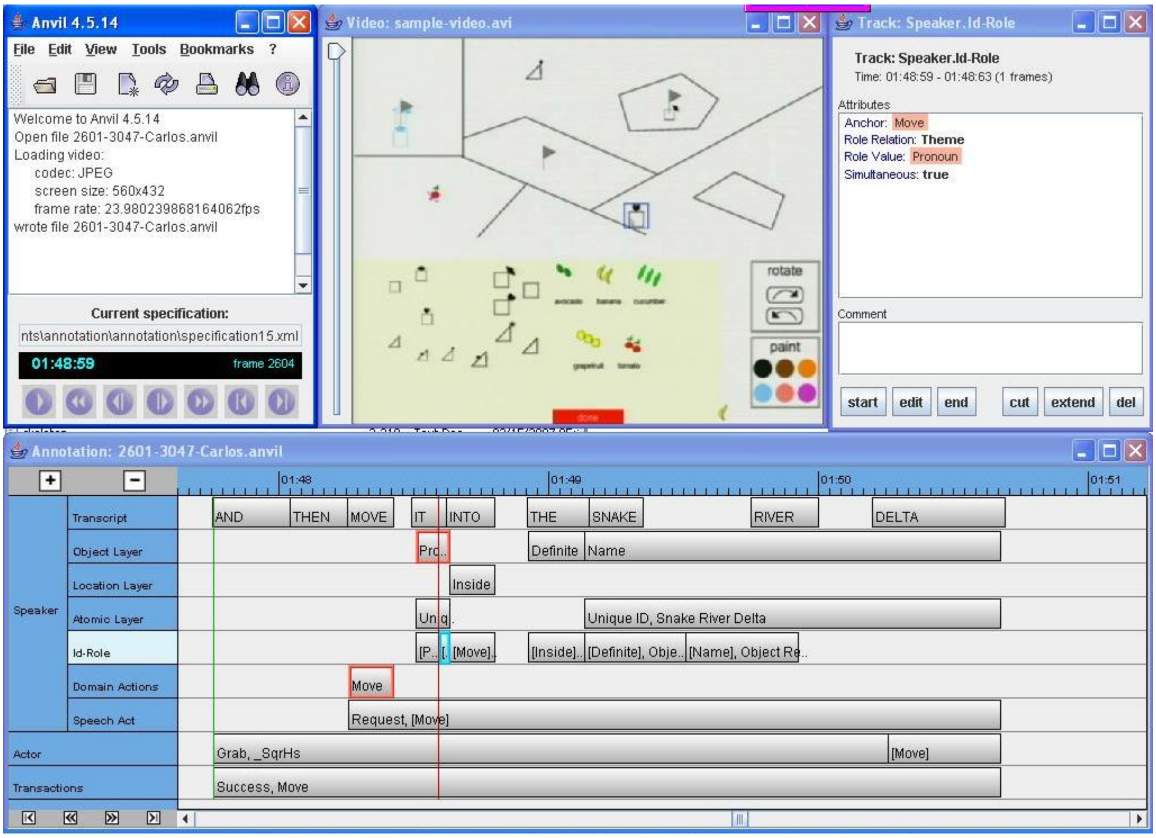
The Fruit Carts domain was devised in order to facilitate the study of continuous understanding of natural language by machines. As such, it contains various points of disambiguation based on factors including object size, color, shape, and decoration; presence or absence of a landmark; and phonetic similarity of geographically close regions of the map (e.g., "Morningside" and "Morningside Heights" are close together.) For example, the objects were designed such that describing the entire shape required a complex description rather than a prenominal modifier. For example, a square with stripes could also be referred to as "the stripey square", but a square with diamonds on the corner cannot be referred to as *"the corner-diamonded square". We thus chose a set of shapes such as "a small square with a diamond on the edge", "a large triangle with a star on the corner", "a small triangle with a circle on the edge", and so forth. Table 1 shows an excerpt of a dialogue in the corpus.

The main operations in the Fruit Carts domain are choosing, placing, painting, rotating an object. The order in which these operations are performed is up to the speaker and the actor. All of the operations are fully reversible in the domain. For example, an object can be returned to the default color by painting it black. This eliminates the need to handle "undo" which is in general a substantial complicating factor for dialogue systems.

The dialogue excerpt in Table 1 illustrates the interaction between the speaker's commands and the actor's actions. Sentences take several interactions to be completed in a combination of visual and auditive interaction. When the speaker utters a command, the actor executes it as soon as he/she has gathered enough information about what to do. During execution, the speaker may give feedback by confirming, correcting, or elaborating as he/she feels appropiate.

SPK> In Morningside there needs to be a triangle with a star on its hypotenuse
ACTR> (*actor moves triangle*)
SPK> Right there and then it needs to be rotated um
ACTR> (*actor waits*)
SPK> to the left
ACTR> (*actor rotates triangle*)
SPK> keep going
ACTR> (*actor keeps rotating*)
SPK> right there
ACTR> (*actor stops*)

Table 1: Example of a Fruit Carts Dialogue



```
Welcome to Anvil 4.5.14
Open file 2601-3047-Carlos.anvil
Loading video:
    codec: JPEG
    screen size: 560x432
    frame rate: 23.980239868164062fps
wrote file 2601-3047-Carlos.anvil
```

Current specification: nts\annotation\annotation\specification15.xml

01:48:59 frame 2604

Track: Speaker.Id-Role
Time: 01:48:59 - 01:48:63 (1 frames)

Attributes
Anchor: Move
Role Relation: **Theme**
Role Value: Pronoun
Simultaneous: **true**

Comment

Figure 1: Annotation of utterance "*and then move it to the snake river delta*"

3 The Tool

Since the corpus we are using has speech and visual modalities on top of speech transcripts, we chose the annotation tool Anvil (Kipp, 2004) for its capabilities to show all these modes concurrently in four different panels (see Figure 1). The lower panel contains the transcript and labels all time aligned with the playable video. The upper middle panel shows the video for a particular session. The upper right panel contains the attributes and the attribute values of the highlighted green box in the Id-role Layer. The upper left panel provides the play, stop, forward buttons to control the playing video.

The multilayer annotation will be described in detail in the following sections. For now, let us briefly present how we represents continuous understanding for a simple annotated utterance "and then move it to the snake river delta" depicted in Figure 1. The speaker is requesting the actor to *Move* a *Square with a Diamond on the Side* to a region called *Snake River Delta*. On the Object Layer we can see the two main entries corresponding to the referring expressions in the utterances (i.e. pronoun "it" and name "snake river delta"). One layer down, the Location Layer, specifies the spatial relation, namely that speaker wants the theme object (i.e. theme) to be *inside of* the *Snake River Delta* region. The Id-role Layer identifies the "it" as the instance of the theme role and "into the snake river delta" as the instance of the location role, both of the Move action.

Figure 1 shows two links by highlighting the boxes with certain colors. The highlighted box on the Id-role Layer identifies the Theme relation of "it" (highlighted box in the Object Layer) with the Move action (highlighted box in the Domain Action Layer). The Speech Act Layer contains the Request act performed by the speaker which links to the domain action Move. On the Actor Layer, there is a label for the action of holding the previously introduced (and selected) object without moving it. The actor then proceeds to move it to the target region as utterance is interpreted. The Transaction Layer shows the committed actions between the speaker and actor finished successfully. In the next section, we explain each of these layers in detail.

Time	Word	Annotation
1	“The”	anchor(A1), definite(A1)
2	“small”	size(A1, small)
3	“box”	objectType(A1, square)
4	“in”	Anchor(A2), spatialRelation(A2, inside), location(A1,A2)
5	“morningside”	anchor(A3), Name(A3), ObjectReferent(A3,MorningsideRegion3), Ground(A2, A3)

Table 2: Detail annotation of “The small box in morningside”

4 The Scheme

Important characteristics of our scheme include the fact that we annotate the speaker’s intentions. This implies that even when certain domain actions, objects or locations are not fully specified in the speech input, the annotation includes the necessary information to execute the commands. For example if speaker says “an avocado in central park”, we construct a Move action even though the verb or command to trigger the action was omitted.

Marking up labels at the word level has a strong implication as well. We are constructing an incremental interpretation of actions, referential expressions, and spatial relationships. Traditionally speech acts have been the smallest unit of annotation. However, in this project we break them down into finer constituents. For instance, with referring expressions, we annotate the object attributes (e.g., size, color, decoration) and break down actions into their semantic roles (e.g., theme, location, angle).

We now present the four principles guiding our annotation scheme in Table 3. Though it is certainly possible and useful to mark labels at the phoneme level, we chose the word level for annotation as a good approximation to incremental annotation as principle 1 states. Principle 2 is applied by reducing speech acts to their minimal units. In our scheme we have object anchors, locations, relation types, core action labels, and all arguments types (e.g., color, angle).

To ensure incremental annotation, labels should be marked exactly at the point where they become unambiguous. The appropiate place to do this is at the point when enough information has been gathered to know the label semantics. Also, even though the transcript contains future sentences, they should not be used for labelling as principle 3 describes. Last, when the speaker uses vocabulary outside the domain, as principle 4 states, we annotate the intended meaning of the word. For instance the speaker may say “tomato” or “apple” both to refer to the same object, or use “move” or “put” both to refer to the same action.

1. Annotation is done at the word level (e.g., not the phonological or sentence level).
2. Annotation is done in minimal semantic increments (e.g., identifying anchors, relation types, arguments).
3. Semantic content is marked at the point it is disambiguated without looking ahead.
4. Reference is annotated according to speaker’s intention.

Table 3: Principles of Annotation.

.

To exemplify how the annotation principles work, let us examine the annotation of a simple NP “The small box in Morningside” in Table 2. The first word that the annotator considers, “the”, introduces a noun phrase. However, we do not yet know the type, color, or size of the object. At this point, the annotator can only introduce an anchor for the object. Later in the speech, the annotator will label object features and link them back to the anchor. In this manner, principle 1 is followed by having the anchor be aligned to the word “the”. Principle 2 is observed when the minimal unit at this point is simply the anchor. In order to follow principle 3, object features are not annotated by using later information (i.e. linking to an entity in the upcoming stream by looking ahead in the transcript or video).

In time step 2, the word “small” is under consideration. The word elaborates one feature of the object which is introduced with anchor A1. The annotator marks the role type (e.g., *size*), role value (e.g., *small*), and role anchor (e.g., *A1*). At time step 3, the object type is introduced by identifying the role type and value in relation to the anchor *A1*.

However, the word “box” was marked as *square* in order to follow principle 4.

5 Description of Domain Actions

The speaker can request the actor to perform certain actions on an object or objects. Domain objects can be selected, moved, rotated, and painted. In addition to these, there are actions that involve mouse movement. For example a Grab action requires the actor to point to an object, select it, and yet not move it. Table 4 shows some of the actions in the Fruit Carts domain along with their semantic roles.

Action	Semantic Roles
Select	obj
Move	obj, location, distance, heading
Rotate	obj, angular distance, heading
Paint	obj, color

Table 4: Actions in the Fruit Carts Domain.

6 Annotation Layers

The speaker utters actions to be performed, domain objects, locations in the map, distances, etc, while the actor is acting in response to these utterances. The speaker may then correct, refine, reject or accept such actions. To annotate this rich amount of information we developed eight layers of annotation (see bottom panel in Figure 1) that convey the dialogue underway focusing on the incremental interpretations of both referential expressions, spatial relations and actions. These layers are the Object, Location, Atomic, Speach Acts (Id-role, Domain Action and Speech Act), Actor, and Transaction layer.

The first three layers encode values for the action semantic roles. In this way noun phrases (Object Layer), spatial relations (Location Layer) and atomic values (Atomic Layer) are ready for the second three layers to refer to. The other layers (see Figure 1) encode the interaction between the speaker language and the actor execution.

6.1 Object Layer

The first layer of annotation is the Object Layer. An object is fully described when its type (e.g., triangle, square, flag, etc), size (e.g., small, big), color, decoration type (e.g., heart, diamond), and location (e.g., corner, side) attributes are all instantiated. Our approach is to annotate NP’s incrementally by identifying an anchor to which each object attribute is linked. The first word of an NP will be marked as an anchor (usually “the” or “a”. To relate attributes to the anchor we use a construct named *Id-role* in order to provide an exact trace of incremental interpretations.

[*Id-role*]: Id-role is a speech act that identifies a particular relationship (the role) between an object (the anchor) and an attribute (the value). It is used for incrementally defining the content of referring expressions and action descriptions

Table 5: Annotation of incremental interpretations with Id-role.

Anchor labels are assigned semantic roles of object features. Anchor types include pronouns, definites, indefinites, names, and demonstratives. If the speaker uses a pronoun, an anchor of type *pronoun* will be marked. Then an Id-Role entry creates the Object Referent relationship between the pronoun (i.e. the anchor) and the domain unique-id (i.e. the value). If on the other hand, the speaker uses a complex NP such as that one in example 2, an anchor is entered at the first word (e.g., “the”, “a”). All other object features are marked and linked to the anchor as they are elaborated by the speaker.

For example, the NP “the triangle with a star on the hypotenuse” has an anchor at “the” of type definite. At the time we hear the word “triangle” we do not know certain semantic roles such as decoration type (whose value is “star”) nor the decoration location (whose value is “on the hypotenuse”). Furthermore, even though the speaker is thinking of a particular object, it is not clear if they are referring to a small or big triangle.

To show this ambiguity and annotate incrementally we mark the anchor which will then be elaborated by identifying role values in later in the speech. Another type of referring expression consists of a group of objects over which an action is distributed, as in “Paint all objects blue”. The annotation of this example follows from the construction of an Id-role which can have a list of values instantiating a role. Thus we would link all relevant objects to the theme role of a Paint action.

This annotation scheme is quite versatile, allowing any objects with partial descriptions be annotated. The interpretation trace of an NP will play an important role in seeing how the action execution is triggered suggesting how early in the command the actor can disambiguate information.

6.2 Location Layer

Entries in this layer encode location descriptions for objects (e.g., “the box in morningside”), and the semantic roles of Move and Rotate actions. Spatial relations contain three attributes: Relation, Relation Modifier and Ground. A relation can be one of the following: *inside of, on top of, right of, left of,* and others. Here again we create an anchor at the first word of the spatial relation (e.g., “in”). An Id-role entry creates the Ground relationship between the anchor and the ground object which serves as frame of reference. Thus an entry in this layer is equivalent to the expression *RELATION (x, ground)* where x is the object holding the relation with the ground.

The Relation Modifier has three values: *a little more*, *a little less* and *touching*. The modifier handles cases where the speaker gives commands incrementally as in “keep going” or “a little more” making heavy use of ellipsis constructions and is particularly used in refinement of the Location semantic role.

As example of this layer, consider the phrase “into the snake river delta” in Figure 1. We create an anchor for “into” with spatial relation type of *inside of.* Since “snake river delta” is a referring expression, it exists in the object layer and it is used as ground object for the spatial relation *inside of.* At this point we can create an Id-role for the Ground relationship between the anchor “into” and the ground object. We also need a second Id-role entry that identifies the anchor “into” as the instance of the location semantic role for the Move action (see Figure 1 and also steps 4 and 5 in Table 2).

Another utterance from the data is the following: “In Morningside Heights, there needs to be a triangle with a star on its hypotenuse”. Notice that the location of the Move action is specified first, before any other argument of the action. Even that we are dealing with a Move action does not follow directly from the copula verb. Other examples such as “the color of the square is blue” also show that the underlying action is not always evident from the verb choice, but rather the argument types.

Our scheme handles these cases nicely due to the versatility of the id-role constructions. For instance, at the time the phrase “In Morningside Heights” is uttered we can not be certain that the speaker is intending a Move action. Thus we are unable to mark it as a location semantic role. This label only happens at a point after the copula verb when the object “a triangle” is specified.

Nevertheless a spatial relation can still be constructed before the location role. The word “in” can be marked as both an anchor for a spatial expression (in the same fashion as NP), and also a *inside of* spatial relation with “Morningside Heights” as *ground.*

6.3 Atomic Layer

The Atomic Layer represents the domain colors, numbers, and the two sizes (small, big) of objects. These are atomic values, as opposed to complex values (i.e. spatial relation). These values instantiate distance, color, and size roles respectively.

As an example, if the speaker utters “rotate it 30 degrees”, we can create an entry for number 30 on this layer. Then the Id-role triplets will relate this number as the angle semantic role for the Rotate action in the Domain Action Layer.

6.4 Speech Act

In this section we describe the Id-role, Domain Action and Speech Act layers. Given that objects, spatial relations and atomic values have been introduced, we can now identify what role these entries have in the action underway using the Id-role construct. Much in the same way of referential expressions, incremental interpretation is an important principle by which we annotate speaker’s actions.

The Id-role construct which has been described in section 6.1 is in the Id-role Layer (see Figure 1). Same as before the Id-role is a triplet that links the semantic roles to its respective value in any of the first three layers (Object, Location or Atomic). Different from before the anchor will not be an *object* being incrementally interpreted but rather an *action* being incrementally interpreted.

The following layer describes the domain actions the speaker can request. These have been explained in section 5. The next layer contains speech acts performed by the speaker. These, described in Table 6, include accept, reject, correct, apology, and others. In this section we are going to focus on the Refine action which is particular to our scheme.

Accept	Speaker can accept or confirm an action performed by the actor.
Request	Speaker can request the actor to perform any of the domain actions.
Correct	A Correct action can be divided into two: a self-correct (speaker) or actor-correct. Such action includes the new information that is being corrected.
Refine	Speaker wants to refine part of the information already given for another action previously stated.

Table 6: Speaker's Speech Acts

We are addressing data that shows incremental instructions to the actor. This occurs largely due to the complex dialogue between speaker and actor that interleaves language and execution. Since speakers see that the actor is interpreting and executing their commands, they feel free to adjust the parameters of their actions. Therefore utterances such as "a little bit more" after a move or rotate command are common (see dialogue 1).

These utterances present elliptical constructions where object, verb and even location are omitted. Usually these sentences will specify arguments given in previous utterances. Notice that the new utterance, either a "a little bit lower" or "keep going" are not contradictory with the previous actions. It is rather an elaboration or refinement of a previous semantic role value (or argument value) of the action. Thus to properly address these types of sentences we have develop an act called Refine that reissues the previous command and refines one of the action arguments. If the new piece of information were contradictory with the already stated actions, the speaker would be uttering a Correct speech act.

6.5 Actor Layer

This layer records the actor's part of the dialogue. It contains all of the domain actions (e.g., select, move) and their possible roles (e.g., object, color, distance). Here we take into account mouse pointing, movements, picking up objects without moving, and releasing objects.

6.6 Transaction Layer

The last layer of annotation is called the Transaction Layer (see Figure 1). It summarizes the speaker-actor interaction by providing the state of the world at the end of all objects manipulations. The Transaction Layer gives us information of what commitments the speaker-actor agree on and whether such commitments finish successfully or unsuccessfully.

At the moment we do not have overlapping transactions. This means that one has to finish before another one starts. Therefore transactions usually contain one domain action with possibly many other speech acts of correction, refinement, rejection, etc. Even though it is certainly possible to have an unfulfilled commitment before acquiring new ones, our current scheme does not allow that.

An utterance such as "move the square to the right and paint it blue" could be thought of a single commitment involving two actions or two overlapping commitments where the first one not yet fullfilled before the second one occurs.

7 Evaluation

An exploratory annotation exercise was performed by two individuals working independently on a same dialogue fragment in order to produce two annotation data sets. Although the annotators were not intensively trained for the task, they were provided with general guidelines.

The inter-annotator agreement, computed as simple percentage and not as kappa statistics (Carletta, 1996), was highest, between 80% and 96%, for labels such as Object Type, Action, Size, Distance, Spatial Relation Modifier, Color, Speech Act and Transaction. Lowest agreement, between 15% and 51%, occurred at labels such as Role Anchors, Role Values, and Speech Act Contents.

These results can be explained as follows: 1) simple values such as color or action types are reliably annotated, well above chance since annotators are choosing from a set of options of around 10 items. 2) linking values that require annotators link to other labels (i.e. linking to different anchors). Since the annotators have not been intensively trained, we are developing a manual annotators can access on line to clarify these issues. Also the annotation scheme is still in a definition and refinement stage and some tagging conventions might be required. This agreement evaluation must be interpreted as a diagnosis tool and not as a final measurement of the scheme reliability. Discrepancies in annotation will be analyzed and discussed to refine the rules and it is expected that the agreement increases when using future improved versions of the scheme.

8 Future Directions

Since referential expressions, spatial relations and speech acts are annotated at the word level as opposed to the sentence level, we have rich information about when objects are brought into discourse, commands are issued by the speaker, actor actions occur, and the state of the world at the end of each transaction. This level of detail allows us to look closely at the relation between actor actions and speaker utterances.

This annotation will allow researchers to evaluate continuous understanding capabilities of conversational agents, develop an intention recognition module that can identify action roles to interpret speech input so that a conversation agent can perform such actions. It may also permit to identify the minimum set of action roles which are required for action recognition, and identify features that correlate a linguistic structure with a particular action role. We can also identify a typical structure of action roles that help recognize which action is underway, and find features that would predict when a transaction is successful and when it is not.

References

G. Aist, J. Allen, E. Campana, L. Galescu, C. Gómez-Gallo, S. Stoness, M. Swift, and M. Tanenhaus. 2006. Software architectures for incremental understanding of human speech. In *Interspeech*.

S. Ait-Mokhtar and J.-P. Chanod. 1997. Incremental finite-state parsing. In *Proc. of the 5th Conf. on Applied Nat. Lang. Proc.*

G. Altmann and Y. Kamide. 1999. Incremental interpretation at verbs: Restricting the domain of subsequent reference. *Cognition*, 73:247–264.

J. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Comp. Ling.*, 22(2):249–254.

M. Collins and B. Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proc. Conf. of ACL.*

M. Core and J. Allen. 1997. Coding dialogues with the DAMSL annotation scheme. In D. Traum, editor, *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines.*

A. Jain and A. Waibel. 1990. Incremental parsing by modular recurrent connectionist networks. In *Proc. of the 2nd Conf. on Advances in Neural Information Proc. (NIPS).*

M. Kipp. 2004. *Gesture Generation by Imitation - From Human Behavior to Computer Character Animation.* Ph.D. thesis, Saarland University.

D. Milward. 1992. Dynamics, dependency grammar and incremental interpretation. In *Proc. of the 14th Conf. on Comp. Ling.*

D. Molla. 2001. Towards incremental semantic annotation. In *Proc. of the 1st MMA.*

L. Pineda, H. Castellanos, S. Coria, V. Estrada, F. López, I. López, I. Meza, I. Moreno, P. Pérez, and C. Rodríguez. 2006. Balancing transactions in practical dialogues. In *CICLing, LNCS.* Springer Verlag.

P. Poller. 1994. Incremental parsing with ld/tlp-tags. *Computational Intelligence*, 10(4).

D. Reitter and M. Stede. 2003. Step by step: underspecified markup in incremental rhetorical analysis. In *Proc. of 4th LINC at EACL*, Budapest.

S. Stoness, J. Tetreault, and J. Allen. 2004. Incremental parsing with reference interaction. In *ACL Workshop on Incremental Parsing.*

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

M. Traxler, M. Bybee, and M. Pickering. 1997. Influence of connectives on language comprehension: Eye- tracking evidence for incremental interpretation. *The Quarterly Journal of Experimental Psychology: Section A*, 50(3).

Hans van Halteren. 1998. The feasibility of incremental linguistic annotation. *Computers and the humanities*, 32(5).

M. Wiren. 1989. Interactive incremental chart parsing. In *Proc. of the 4th meeting of EACL.*

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 83–90.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

A Chatbot-based Interactive Question Answering System

Silvia Quarteroni
The University of York
York, YO10 5DD
United Kingdom
silvia@cs.york.ac.uk

Suresh Manandhar
The University of York
York, YO10 5DD
United Kingdom
suresh@cs.york.ac.uk

Abstract

Interactive question answering (QA) systems, where a dialogue interface enables followup and clarification questions, are a recent field of research. We report our experience on the design, implementation and evaluation of a chatbot-based dialogue interface for our open-domain QA system, showing that chatbots can be effective in supporting interactive QA.

1 Introduction

Question answering (QA) systems can be seen as information retrieval systems which aim at responding to natural language queries by returning answers rather than lists of documents.

Although QA differs from standard information retrieval in the response format, both processes share a lack of interactivity. In the typical information-seeking session the user submits a query and the system returns a result; the session is then concluded and forgotten by the system.

It has been argued (Hobbs, 2002) that providing a QA system with a dialogue interface would encourage and accommodate the submission of multiple related questions and handle the user's requests for clarification. Indeed, information-seeking dialogue applications of QA are still at an early stage and often relate to close domains (Small et al., 2003; Jönsson and Merkel, 2003; Kato et al., 2006).

In this paper, we report on the design, implementation and evaluation of the dialogue interface for our open-domain QA system, YourQA (Quarteroni and Manandhar, 2006). The system is able to provide both factoid and complex answers such as definitions and descriptions. The dialogue interface's role is to enable an information seeking, cooperative, inquiry-oriented conversation to support the question answering component.

Section 2 introduces the design of our interactive QA system; Section 3 describes an exploratory study conducted to confirm our design assumptions. The implementation and evaluation of our prototype are described in Sections 4 and 5. Section 6 briefly concludes on our study.

2 System Design

Our open domain QA system, YourQA, takes the top 20 Google results for a question, retrieves the corresponding Web pages and analyzes them to extract answers and rank them by relevance to the question. A non-interactive interface exists for the system where users enter a question in a text field, and obtain a list of answers in the form of an HTML result page (Quarteroni and Manandhar, 2006).

We now describe the dialogue scenario and management model for the interactive version of the system, where the core QA component is mediated by a dialogue interface.

2.1 Dialogue scenario

In the dialogue scenario we are modelling, a typical QA session consists of the following *dialogue moves*:

1. An initial greeting (*greet* move), or a direct question *q* from the user (*ask(q)* move);

2. *q* is analyzed to detect whether it is related to previous questions;

   (a) If *q* is unrelated to the preceding questions, it is submitted to the QA component;

(b) If *q* is related to the preceding questions (i.e. *q* is a followup question), and is elliptic, i.e. contains no verb (*"Why?"*), the system uses the previous questions to complete *q* with the missing keywords and submits a revised question *q'* to the QA component;

(c) If *q* is a followup question and is anaphoric, i.e. contains references to entities in the previous questions, the system tries to create a revised question *q"* where such references are replaced by their corresponding entities, then checks whether the user actually means *q"* (move *ground(q")*); if the user agrees, query *q"* is issued to the QA component. Otherwise, the system asks the user to reformulate his/her utterance (move *sysReqClarif*) until finding a question which can be submitted to the QA component;

3. As soon as the QA component results are available, an answer *a* is provided (*answer(a)* move);

4. The system enquires whether the user is interested in a *followup* session; if this is the case, the user can enter a query (*ask* move) again. Else, the system acknowledges (*ack*);

5. Whenever the user wants to terminate the interaction, a final greeting is exchanged (*quit* move).

At any time the user can issue a request for clarification (*usrReqClarif(r)*) in case the system's utterance is not understood.

2.2 Dialogue Moves

The dialogue moves with which the interactive QA scenario above is annotated are summarized in Tables 1 and 2. Such moves are a generalization of the dialogue move sets proposed for other information-oriented dialogue models such as GoDiS (Larsson et al., 2000) and Midiki (MITRE Corporation, 2004).

We now discuss the choice of a dialogue management model to implement such moves.

2.3 Choosing a dialogue manager

When designing information-seeking dialogue managers, the simplest approaches appear to be finite-state (FS) models. Here, each phase of the conversation is modelled as a separate state, and each dialogue move encodes a transition to a subsequent state (Sutton, 1998).

User move	Description
greet	conversation opening
quit	conversation closing
ask(q)	user asks question *q*
ack	acknowledgement of previous utterance, e.g. *"Thanks."*
usrReqClarif(r)	clarification request (*r* = reason)

Table 1: User dialogue moves

System move	Description
greet	conversation opening
quit	conversation closing
answer(a)	answer (*a* = answer)
ack	acknowledgement of previous utterance, e.g. *"Ok."*
sysReqClarif	clarification request
ground(q)	grounding (*q* = question)
followup	proposal to continue session

Table 2: System dialogue moves

However, FS models allow very limited freedom in the range of user utterances. Since each dialogue move must be pre-encoded in the models, these are not scalable to open domain dialogue.

A more complex dialogue management model is the Information State (IS) approach inspired by Ginzburg's dialogue gameboard theory (Ginzburg, 1996). The topics under discussion and common ground in the conversation are part of the IS and continually queried and updated by rules fired by participants' dialogue moves. The IS theory, introduced in the TRINDI project (Larsson and Traum, 2000), has been applied to a range of closed-domain dialogue systems (e.g. travel information, route planning).

Although it provides a powerful formalism, the IS infrastructure appears too voluminous for our QA application. We believe that the IS approach is primarily suited to applications requiring a planning component such as in closed-domain dialogue systems and to a lesser extent in our open-domain dialogue system as we currently do not make use of planning. Moreover, in our system the context is only used for question clarification purposes.

2.3.1 The chatbot approach

As a solution joining aspects of both FS and IS approaches, we studied the feasibility of a conversational agent based on an AIML interpreter.

AIML (Artificial Intelligence Markup Language) was designed for the creation of conversational robots ("chatbots") such as ALICE[1]. It is based on pattern matching, which consists in matching the last user utterance against a range of dialogue patterns known to the system ("categories") in order to produce a coherent answer following a range of "template" responses.

Designed for chatting, chatbot dialogue appears more natural than in FS and IS systems. Moreover, since chatbots support a limited notion of context, they seem to offer the means to support followup recognition and other dialogue phenomena not easily covered using standard FS models.

To assess the feasibility of chatbot-based QA dialogue, we conducted an exploratory Wizard-of-Oz experiment, described in Section 3.

3 Wizard-of-Oz experiment

A Wizard-of-Oz (WOz) experiment is usually deployed for natural language systems to obtain initial data when a full-fledged prototype is not yet available (Dahlbaeck et al., 1993). The experiment consists in "hiding" a human operator (the "wizard") behind a computer interface to simulate conversation with the user, who believes to be interacting with a fully automated prototype.

3.1 Assumptions

In addition to the general assumption that a chatbot would be sufficient to successfully conduct a QA conversation, we intended to explore whether a number of further assumptions were founded in the course of our experiment.

First, users would use the system to obtain information, thus most of their utterances would be questions or information requests.

Then, users would easily cope with the system's requests to rephrase their previous utterances should the system fail to understand them.

Finally, the user's clarification requests would be few: as a matter of fact, our answer format provides more information than explicitly required and this has been shown to be an effective way to reduce the occurrence of clarification requests (Kato et al., 2006; Hickl and Harabagiu, 2006).

3.2 Task Design

We designed six tasks, to be proposed in groups of two to six or more subject so that each task was performed by at least two different users. These reflected the intended typical usage of the system (e.g. *"Find out who painted Guernica and ask the system for more information about the artist"*).

Users were invited to test the supposedly completed prototype by interacting with an instant messaging platform, which they were told to be the system interface.

Since our hypothesis was that a conversational agent is sufficient to handle question answering, a set of AIML categories was created to represent the range of utterances and conversational situations handled by a chatbot.

The role of the wizard was to choose the appropriate category and utterance within the available set, and type it into the chat interface to address the user. If none of these appeared appropriate to handle the situation at hand, the wizard would create one to keep the conversation alive and preserve the illusion of interacting with a machine. The wizard asked if the user had any follow-up questions after each answer (*"Can I help you further?"*).

3.3 User Feedback Collection

To collect user feedback, we used two sources:

- the *chat logs*, which provided information about the situations that fell above the assumed requirements of the chat bot interface, the frequency of requests for repetition, etc.;
- a *questionnaire* submitted to the user immediately after the WOz experiment, enquiring about the user's experience.

The questionnaire, inspired by the WOz experiment in (Munteanu and Boldea, 2000) consists of six questions:

Q_1 *Did you get all the information you wanted using the system?*

Q_2 *Do you think the system understood what you asked?*

Q_3 *How easy was it to obtain the information you wanted?*

Q_4 *Was it difficult to reformulate your questions when you were invited to?*

[1] http://www.alicebot.org/

Q_5 *Do you think you would use this system again?*
Q_6 *Overall, are you satisfied with the system?*

Questions Q_1 and Q_2 assess the performance of the system and were ranked on a scale from 1= "Not at all" to 5="Yes, Absolutely". Questions Q_3 and Q_4 focus on interaction difficulties, especially relating to the system's requests to reformulate the user's question. Questions Q_5 and Q_6 relate to the overall satisfaction of the user. The questionnaire also contained a text area for optional comments.

3.4 WOz experiment results

The WOz experiment was run over one week and involved one wizard and seven users. These came from different backgrounds and native languages, were of different ages and were regular users of search engines. All had used chat interfaces before and had an account on the platform used for the experiment, however only one of them doubted that they were confronted to a real system. The average dialogue duration was 11 minutes, with a maximum of 15 minutes (2 cases) and a minimum of 5 minutes (1 case).

From the chat logs, we observed that as predicted all dialogues were information seeking. One unexpected result was that users often asked two things at the same time (e.g. *"Who was Jane Austen and when was she born?"*). To account for this case, we decided to handle double questions in the final prototype, as described in Section 4.

The *sysReqClarif* dialogue move proved very useful, and sentences such as *"Can you please reformulate your question?"* or *"In other words, what are you looking for?"* were widely used. Users seemed to enjoy "testing" the system and accepted the invitation to produce a followup question (*"Can I help you further?"*) around 50% of the time.

Our main observation from the user comments was that users seemed to receive system grounding and clarification requests well, e.g. *" . . . on references to "him/it", pretty natural clarifying questions were asked."*

The values obtained for the user satisfaction questionnaire, reported in Table 3, show that users tended to be particularly satisfied with the system's performances and none of them had difficulties in reformulating their questions (Q_4) when this was requested (mean 3.8, standard deviation .5, where 3 = "Neutral" and 4 = "Easy"). For the remaining questions, satisfaction levels were high, between 4±.8 (Q_3) and 4.5±.5 (Q_5).

Question	judgment	Question	judgment
Q_1	4.3±.5	Q_2	4.0
Q_3	4.0±.8	Q_4	3.8±.5
Q_5	4.1±.6	Q_6	4.5±.5

Table 3: Wizard-of-Oz questionnaire results: mean ± standard deviation

4 System Architecture

The dialogue manager and interface were implemented based on the scenario in Section 2 and the outcome of the WOz experiment.

4.1 Dialogue Manager

Chatbot dialogue follows a pattern-matching approach, and is therefore not constrained by a notion of "state". When a user utterance is issued, the chatbot's strategy is to look for a pattern matching it and fire the corresponding template response.

Our main focus of attention in terms of dialogue manager design was therefore directed to the dialogue moves invoking external modules such as the followup recognition and QA component.

We started from the premise that it is vital in handling QA dialogue to apply an effective algorithm for the recognition of followup requests, as underlined in (De Boni and Manandhar, 2005; Yang et al., 2006). Hence, once a user utterance is recognized as a question by the system, it attempts to clarify it by testing whether it is a double question or a followup question.

4.1.1 Handling double questions

For the detection of *double* questions, the system uses the OpenNLP chunker[2] to look for the presence of "and" which does not occur within a noun phrase. If it is found, the system simply offers to answer one of the two "halves" of the double question (the one containing more tokens) as the QA component is not able to handle multiple questions.

4.1.2 Handling followup questions

The types of followup questions which the system is able to handle are elliptic questions, questions containing third person pronoun/possessive adjective anaphora, or questions containing noun

[2] `http://opennlp.sourceforge.net/`

phrase (NP) anaphora (e.g. "the river" instead of "the word's longest river").

Detection of followup questions For the detection of followup questions, the algorithm in (De Boni and Manandhar, 2005) is used. This is based on the following features: presence of pronouns, absence of verbs, word repetitions and similarity between the current and the n preceding questions. The algorithm is reported below:

```
Followup_question ($q_i$, $q_i..q_{i-n}$)
is true if

 1. $q_i$ has pronoun and possessive
    adjective references to $q_i..q_{i-n}$

 2. $q_i$ contains no verbs

 3. $q_i$ has repetition of common or
    proper nouns in $q_i..q_{i-n}$
    or
    $q_i$ has a strong semantic
    similarity to some $q_j \in q_i..q_{i-n}$
```

Following the authors, we apply the above algorithm using $n = 8$; at the moment the condition on semantic distance is not included for the sake of processing speed.

Resolution of followup questions If a question *q* is identified as a followup question, it is submitted to the QA component; otherwise the following reference resolution strategy is applied:

1. if *q* is *elliptic* (i.e. contains no verbs), its keywords are completed with the keywords extracted by the QA component from the previous question for which there exists an answer. The completed query is submitted to the QA component.

2. if *q* is *anaphoric*:

   (a) in case of *pronoun/adjective anaphora*, the chunker is used to find the first compatible antecedent in the previous questions in order of recency. The latter must be a NP compatible in number with the referent.

   (b) in case of *NP anaphora*, the first NP containing all of the referent words is used to replace the referent in the query.

   In both cases, when no antecedent can be found, a clarification request is issued by the system until a resolved query can be obtained and submitted to the QA component.

Ellipsis and reference resolution is useful not only for question interpretation but also to optimize the retrieval phase: it suggests to extract answers from the same documents collected to answer the antecedent question (De Boni and Manandhar, 2005). Hence, if a clarified followup question is submitted to the QA component, the QA system extracts answers from the documents retrieved for the previous question.

When the QA process is terminated, a message directing the user to the HTML answer page is returned and the followup proposal is issued. We must point out that such solution implies that the clarification and followup abilities of YourQA are limited to the questions. Indeed, it is not possible to handle *answer* clarification at the moment: it would be impossible for the system to conduct a conversation such as:

User*$_n$*: Who was Shakespeare married to?
System*$_n$*: Anne Hathaway.
User*$_{n+1}$*: what was her profession?

Data-driven answer clarification in the open domain is an open issue which we would like to study in the future, in order to make the dialogue component more tied into the structure of the QA system.

4.2 Implementation

Following the typical implementation of a pattern-matching conversational agent, we designed a set of patterns to cover the dialogue scenarios elaborated in the design stage and enriched with the WOz experiment.

4.2.1 AIML interpreter and context

First, we grouped the AIML categories in different .aiml files, each corresponding to one of the dialogue moves in Table 2.

We used the Java-based AIML interpreter Chatterbean[3], which allows to define custom AIML tags and allows a seamless integration between the QA module and the chat interface.

We augmented the Chatterbean tag set with two AIML tags:

- `<query>`, to invoke the YourQA question answering module;
- `<clarify>`, to support the tasks of clarification detection and reference resolution.

The Chatterbean implementation of the conversation context (in a dedicated `Context` class) al-

[3] http://chatterbean.bitoflife.cjb.net/

lows to instantiate and update a set of variables, represented as context *properties*. We defined several of these, including:

- the user's *name*, matched against a list of known user names to select a profile for personalized answer extraction (this feature of YourQA is not discussed here);
- the current *query*, used to dynamically update the stack of recent user questions. The stack is used by the clarification request detection module to perform reference resolution, following the algorithm exposed in Section 4.1;
- the *resolved* question, i.e. the current query as resolved during followup handling (identical to the current query if no followup is detected);
- the *topic* of conversation, i.e. the set of keywords of the last question issued by the user which received an answer (e.g. "What is the world's longest river?" => {`world, longest, river`}). The topic keywords are used to clarify elliptic questions, by augmenting the current query keywords (e.g. "In what country?"=> {`country, world, longest, river`}).

To illustrate a typical case of use of the implementation we start with the AIML category:

```
<category>
<pattern>DO YOU KNOW *</pattern>
<template><srai>CLARIFY *</srai>
</template>
</category>
```

Here, the system attempts to clarify the content of the request following the cue words "Do you know". The template:

```
<template><srai>CLARIFY *</srai>
</template>
```

invokes a category calling the newly introduced AIML tag `<clarify>`:

```
<category>
<pattern>CLARIFY *</pattern>
<template>
<think><set name="clarif">
<clarify></star><clarify>
</set></think>
```

The `<clarify>` tag in turn invokes the Java implementation dedicated to clarification recognition on the text matching the `"*"` expression, which follows the strategy exposed in Section 4.1. Finally, the clarified query (assigned to the context property `"clarif"`) as well as the output of the followup detection algorithm (e.g. `"ELLIPTIC"`) are used in a conditional branch to call the appropriate routine:

```
<condition name="clarif"
value="ELLIPTIC"> ...</condition>
<condition name="clarif"
value="DOUBLE"> ...</condition>
...
</template>
</category>
```

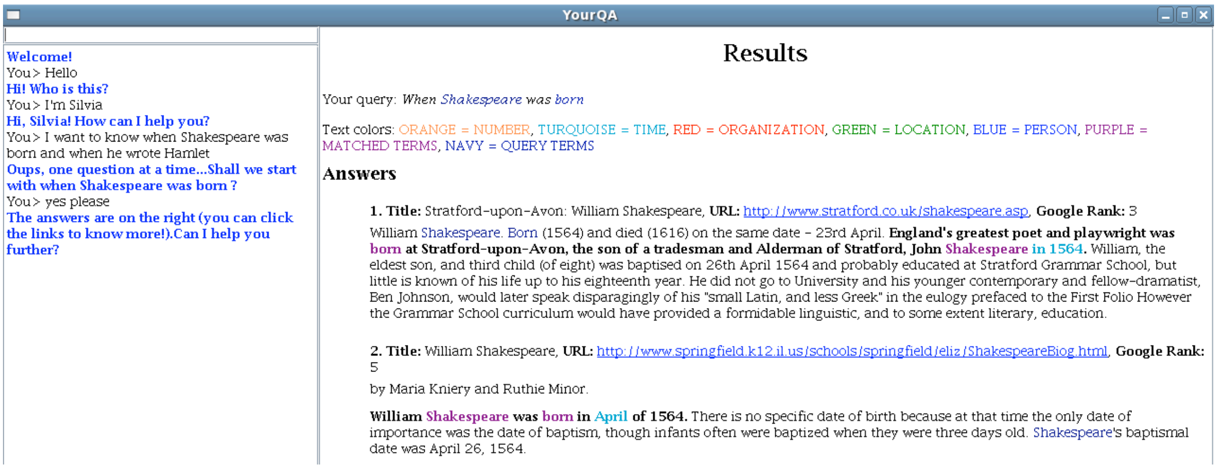
4.2.2 Dialogue Interface

YourQA's interactive interface consists of an applet window with a left panel where the chat takes place and a right panel where results are visualized when ready (see Figure 1). As in a normal chat application, users enter text in a text field and the current session history as well as the system replies are visualized in an adjacent text area.

An example dialogue deriving from a real interaction with the system is the following:
System$_0$*: Welcome!*
User$_1$*: Hi*
S$_1$*: Hi! How can I help you?*
U$_2$*: what is a thermometer?*
S$_2$*: Possible answers are in the panel on the right. Are you happy with these answers?*
U$_3$*: yes!*
S$_3$*: Good!*
U$_4$*: How does it measure the temperature?*
S$_4$*: Do you mean how does a thermometer measure the temperature?*
U$_5$*: yes*
S$_5$*: I found the answers in the panel on the right. Can I help you further?*
U$_6$*: ...*

5 Evaluation

While the accuracy of standard QA systems can be evaluated and compared using quantitative information retrieval metrics (Voorhees, 2003), dialogue interfaces pose complex evaluation challenges as they differ in appearance, intended application and target users.



Welcome!
You> Hello
Hi! Who is this?
You> I'm Silvia
Hi, Silvia! How can I help you?
You> I want to know when Shakespeare was born and when he wrote Hamlet
Oups, one question at a time...Shall we start with when Shakespeare was born ?
You> yes please
The answers are on the right (you can click the links to know more!).Can I help you further?

Results

Your query: *When Shakespeare was born*

Text colors: ORANGE = NUMBER, TURQUOISE = TIME, RED = ORGANIZATION, GREEN = LOCATION, BLUE = PERSON, PURPLE = MATCHED TERMS, NAVY = QUERY TERMS

Answers

1. **Title:** Stratford-upon-Avon: William Shakespeare, **URL:** http://www.stratford.co.uk/shakespeare.asp, **Google Rank:** 3

William Shakespeare. Born (1564) and died (1616) on the same date - 23rd April. **England's greatest poet and playwright was born at Stratford-upon-Avon, the son of a tradesman and Alderman of Stratford, John Shakespeare in 1564.** William, the eldest son, and third child (of eight) was baptised on 26th April 1564 and probably educated at Stratford Grammar School, but little is known of his life up to his eighteenth year. He did not go to University and his younger contemporary and fellow-dramatist, Ben Johnson, would later speak disparagingly of his "small Latin, and less Greek" in the eulogy prefaced to the First Folio However the Grammar School curriculum would have provided a formidable linguistic, and to some extent literary, education.

2. **Title:** William Shakespeare, **URL:** http://www.springfield.k12.il.us/schools/springfield/eliz/ShakespeareBiog.html, **Google Rank:** 5

by Maria Kniery and Ruthie Minor.

William Shakespeare was born in April of 1564. There is no specific date of birth because at that time the only date of importance was the date of baptism, though infants often were baptized when they were three days old. Shakespeare's baptismal date was April 26, 1564.

Figure 1: The chat interface (partial view)

Indeed, these are often evaluated using qualitative metrics such as user satisfaction and perceived time of usage (Walker et al., 2000). Similarly, user satisfaction questionnaires and interaction logs appear to be effective tools to evaluate interactive QA systems (Kelly et al., 2006).

5.1 Experiment design

To quickly conduct a preliminary evaluation of our prototype, we designed three scenarii where users had to look for two different items of infomation relating to the same topic (e.g. Shakespeare's date of birth and when he wrote Hamlet), as in the previous WOz experiment. Users had to choose one or more topics and use first the non-interactive Web interface of the QA prototype (handling questions in a similar way to a search engine) and then the interactive version depicted in Figure 1 to find answers.

After using both versions of the prototype, users filled in a questionnaire about their experience with the *chat version* which comprised the same questions as the WOz questionnaire and the following additional questions:

Q_7 *Was the pace of interaction with the system appropriate?*

Q_8 *How often was the system sluggish in replying to you?*

Q_9 *Did you prefer the chat or the Web interface and why?*

Questions Q_7 and Q_8 could be answered using a scale from 1 to 5 and were taken from the PARADISE evaluation questions (Walker et al., 2000). Q_9 was particularly interesting to assess if and in what terms users perceived a difference between the two prototypes. All the interactions were logged.

5.2 Evaluation results

From the initial evaluation, which involved six volunteers, we gathered the following salient results:

1. in the chat logs, when pronominal anaphora was used by the users, the system was able to resolve it in seven out of nine cases;
2. no elliptic queries were issued, although in two cases verbs were not spotted by the system causing queries to be completed with previous query keywords;
3. due to the limited amount of AIML categories of the system, the latter's requests for reformulation occurred more frequently than expected;
4. Users tended not to reply to the chatbot offers to carry on the interaction explicitly, directly entering a followup question instead.

From the questionnaire (Tab. 4), we collected sightly lower user satisfaction levels (Q_1 to Q_6) than in the WOz experiment (Section 3).

Users felt the system to reply slowly to the questions (Q_7 and Q_8). This is mainly because the system performs document retrieval in real time, hence it heavily depends on the network download speed.

All but one user (i.e. 83.3%) said they preferred the chat interface of the system (Q_9), because of its liveliness and ability to understand pronominal anaphora.

Question	judgment	Question	judgment
Q_1	3.8±.4	Q_2	3.7±.8
Q_3	3.8±.8	Q_4	3.8±.8
Q_5	4.0±.9	Q_6	4.3±.5
Q_7	3.5±.5	Q_8	2.3±1.2

Table 4: Questionnaire results: mean±standard deviation

6 Conclusions

This paper reports the design and implementation of a chatbot-based interface for an open domain, interactive question answering (QA) system. From our preliminary evaluation, we draw optimistic conclusions on the feasibility of chatbot-based interactive QA.

In the future, we will study more advanced strategies for anaphora resolution in questions, e.g. (Poesio et al., 2001) and conduct a more thorough evaluation of our dialogue interface.

As mentioned earlier, we are also interested in data-driven answer clarification approaches for the open domain to further integrate the dialogue component into the QA system.

Acknowledgement We are indebted to the anonymous reviewers for their valuable comments on the initial version of this paper.

References

N. Dahlbaeck, A. Jonsson, and L. Ahrenberg. 1993. Wizard of Oz studies: why and how. In *Proceedings of IUI '93*, pages 193–200, New York, NY, USA. ACM Press.

M. De Boni and S. Manandhar. 2005. Implementing clarification dialogue in open-domain question answering. *Nat. Lang. Eng.*, 11.

J. Ginzburg, 1996. *Interrogatives: Questions, Facts and Dialogue*. Blackwell, Oxford.

A. Hickl and S. Harabagiu. 2006. Enhanced interactive question answering with conditional random fields. In *Proceedings of IQA*.

J. R. Hobbs. 2002. From question-answering to information-seeking dialogs.

A. Jönsson and M. Merkel. 2003. Some issues in dialogue-based question-answering. In *Working Notes from AAAI Spring Symposium*, Stanford.

T. Kato, J. Fukumoto, F.Masui, and N. Kando. 2006. Woz simulation of interactive question answering. In *Proceedings of IQA*.

D. Kelly, P. Kantor, E. Morse, J. Scholtz, and Y. Sun. 2006. User-centered evaluation of interactive question answering systems. In *Proceedings of IQA*.

S. Larsson and D. R. Traum. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Nat. Lang. Eng.*, 6(3-4):323–340.

S. Larsson, P. Ljunglöf, R. Cooper, E. Engdahl, and S. Ericsson. 2000. GoDiS—an accommodating dialogue system. In C. Sidner, editor, *ANLP/NAACL Workshop on Conversational Systems*, pages 7–10, Somerset, New Jersey. ACL.

MITRE Corporation, 2004. *MIDIKI User's manual*.

C. Munteanu and M. Boldea. 2000. MDWOZ: A Wizard of Oz Environment for Dialog Systems Development. In *Proceedings of LREC*.

M. Poesio, U. Reyle, and R. Stevenson, 2001. *Justified sloppiness in anaphoric reference – Computing meaning*, chapter 3. Kluwer.

S. Quarteroni and S. Manandhar. 2006. User modelling for adaptive question answering and Information Retrieval. In *Proceedings of FLAIRS*.

S. Small, T. Liu, N. Shimizu, and T. Strzalkowski. 2003. HITIQA: an interactive question answering system- a preliminary report. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and QA*, pages 46–53, Morristown, NJ, USA. ACL.

S. Sutton. 1998. Universal speech tools: the CSLU toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

E. M. Voorhees. 2003. Overview of the TREC 2003 question answering track. In *Text REtrieval Conference*.

M. A. Walker, C. Kamm, and D. Litman. 2000. Towards Developing General Models of Usability with PARADISE. *Nat. Lang. Eng. Special Issue on Best Practice in Spoken Dialogue Systems*.

F. Yang, Z. Feng, and G. Di Fabbrizio. 2006. A data driven approach to relevancy recognition for contextual question answering. In *Proceedings of IQA*.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 91–98.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Towards Flexible, Domain-Independent Dialogue Management using Collaborative Problem Solving

Nate Blaylock
40 South Alcaniz Street
Institute for Human and Machine Cognition (IHMC)
Pensacola, Florida, USA
blaylock@ihmc.us

Abstract

In this paper, we describe our first efforts at building a domain-independent dialogue manager based on a theory of collaborative problems solving. We describe the implemented dialogue manager and look critically at what level of domain independence was achieved and what remains to be done.

1 Introduction

We are interested in building *conversational agents*—autonomous agents which can communicate with humans through natural language dialogue. In order to support dialogue with autonomous agents, we need to be able to model dialogue about the range of activities an agent may engage in, including such things as goal evaluation, goal selection, planning, execution, monitoring, replanning, and so forth.

Current models of dialogue are only able to support a small subset of these sorts of agent activities. Plan-based dialogue models, for example, typically model either planning dialogue (e.g., (Grosz and Sidner, 1990)) or execution dialogue (e.g., (Cohen et al., 1991)), but not both. Also, most plan-based dialogue models make the assumption that agents already have a high-level goal which they are pursuing.

In our previous work (Blaylock and Allen, 2005), we presented a model of dialogue based on collaborative problem solving (CPS), which includes the set of agent activities mentioned above. This CPS-based model of dialogue allows us to model a much wider range of dialogue types and phenomena than previous models.

Besides allowing us to model more complex types of dialogue, it is the hope that CPS dialogue will help with two other important aspects of dialogue: *flexibility* and *portability*. By flexibility, we mean the ability of the system to cover all natural dialogues (i.e., dialogues that humans would naturally engage in) for a given domain. Flexibility is important for naturalness and ease of use, as well as making sure we can understand and incorporate anything the user might say to the system.

Portability refers to to the ease with which the system can be modified to work in new domains. Portability is especially important to the commercial viability of dialogue systems. For dialogue management, our goal is to create a domain-independent dialogue manager that supports "instantiation" to a particular domain through the use of a small amount of domain-specific knowledge. Several recent dialogue managers approach this level of portability ((Larsson, 2002; Bohus and Rudnicky, 2003), inter alia), however, these are based on models of dialogue which do not cover the range of agent activity that we need (see (Blaylock, 2005) for arguments), and they sacrifice some flexibility. Flexibility is lost, as these dialogue managers require a dialogue designer to specify so-called dialogue plans, as part of the domain-specific information fed to the domain-independent dialogue manager. However, these dialogue plans contain not only domain-dependent task knowledge (e.g., the process for making a travel reservation), but also knowledge about how to *interact* with a user about this knowledge (e.g., greet the user, find out travel destination). This essentially puts the onus of dialogue flexibility in the hands of the dialogue system designer, limiting flexibility to the set of dialogues "described" or "encoded" by the dialogue plan. It is our hope that CPS-based dialogue will result in more flexibility and better portability than previous systems by factoring this interaction knowledge out from domain-dependent task knowledge.

In this paper, we report the progress of our first efforts in building a CPS-based dialogue manager within the SAMMIE-05 dialogue system. We will first briefly describe the CPS dialogue model, and then the SAMMIE-05 dialogue system. We then discuss the implementation SAMMIE-05 dialogue manager and then comment on general progress towards domain independence. We then mention related work and talk about future directions.

2 Modeling Dialogue as Collaborative Problem Solving

In this section, we very briefly describe our CPS model of dialogue. Details of the model can be found in (Blaylock and Allen, 2005; Blaylock, 2005). We first describe our model of collaborative problem solving, and then how that is used to model dialogue.

2.1 A Model of Collaborative Problem Solving

We see problem solving (PS) as the process by which a (single) agent chooses and pursues *objectives* (i.e., goals). Specifically, we model it as consisting of the following three general phases:

- *Determining Objectives*: In this phase, an agent manages objectives, deciding to which it is committed, which will drive its current behavior, etc.

- *Determining and Instantiating Recipes for Objectives*: In this phase, an agent determines and instantiates a recipe to use to work towards an objective. An agent may either choose a recipe from its recipe library, or it may choose to *create* a new recipe via planning.

- *Executing Recipes and Monitoring Success*: In this phase, an agent executes a recipe and monitors the execution to check for success.

There are several things to note about this general description. First, we do not impose any strict ordering on the phases above. For example, an agent may begin executing a partially-instantiated recipe and do more instantiation later as necessary. An agent may also adopt and pursue an objective in order to help it in deciding what recipe to use for another objective.

It is also important to note that our purpose here is not to specify a specific *problem-solving strategy* or prescriptive model of how an agent *should* perform problem solving. Instead, we want to provide a general descriptive model that enables agents with different PS strategies to still communicate.

Collaborative problem solving (CPS) follows a similar process to single-agent problem solving. Here two agents jointly choose and pursue objectives in the same stages (listed above) as single agents.

There are several things to note here. First, the level of collaboration in the problem solving may vary greatly. In some cases, for example, the collaboration may be primarily in the planning phase, but one agent will actually execute the plan alone. In other cases, the collaboration may be active in all stages, including the planning and execution of a joint plan, where both agents execute actions in a coordinated fashion. Again, we want a model that will cover the range of possible levels of collaboration.

Examples of Problem-Solving Behavior In order to better illustrate the problem solving behavior we want to cover in our model, we give several simple examples.

- *Prototypical*: Agent Q decides to go to the park (objective). It decides to take the 10:00 bus (recipe). It goes to the bus stop, gets on the bus and then gets off at the park (execution). It notices that it has accomplished its objective, and stops pursuing it (monitoring).

- *Interleaved Planning and Execution*: Agent Q decides to to go to the park. It decides to take a bus (partial recipe) and starts walking to the bus stop (partial execution) as it decides which bus it should talk (continues to instantiate recipe)....

- *Replanning*: Agent Q decides to go to the park. It decides to walk (objective) and goes outside of the house (begins execution). It notices that it is raining and that doesn't want to walk to the park (monitoring). It decides instead to take the 10:00 bus (replanning)....

- *Abandoning Objective*: Agent Q decides to go to the park by taking the 10:00 bus. As it walks outside, it notices that it is snowing

> and decides it doesn't want to go to the park (abandons objective). It decides to watch TV instead (new objective)....

2.1.1 Problem-Solving Objects

The CPS model operates on problem-solving (PS) objects which are represented as typed feature structures. We define an upper-level ontology of such objects, and define the CPS model around them (which helps keep it domain independent). The ontology can then be extended to concrete domains through inheritance and instantiation of the types defined here.

The ontology defines six *abstract PS objects*, from which all other PS objects descend: *objective, recipe, constraint, resource, evaluation*, and *situation*. Types in the model are defines as typed feature structures, and domain knowledge is connected to the ontology by both inheritance in new classes, as well as creating instances of ontological objects.

2.1.2 Collaborative Problem-Solving Acts

We also define a set of actions which operate on these PS objects. Some of these include `identifying` and object for use in problem solving, `adopting` an object for some specific role (e.g., committing to use a particular resource in the plan), `selecting` an objective for execution.

Collaboration cannot be forced by a single-agent, so we define on top of the CPS acts, a model of negotiation, in which agents can negotiate changes to the current CPS state (i.e., the set of PS objects and the agents' joint commitments to them).

2.2 Integrating CPS into a Dialogue Model

So far, we have described a model of CPS for any agent-agent collaboration. In order to use CPS to model dialogue, we add an additional layer of grounding based on Traum's grounding model (Traum, 1994), which gives the model coverage of grounding phenomena in language as well.

In modeling dialogue with CPS, we use the CPS state as part of the information state of the dialogue, and the meaning of each utterance (from both parties), can be described as a move in the negotiation of change to the current CPS state (augmented with grounding information). Incidentally, this also allows us to model the intentions of individual utterances in a dialogue.

3 The SAMMIE-05 System

The SAMMIE-05 system (Becker et al., 2006)[1] is a multimodal, mixed-initiative dialogue system for controlling an MP3 player. The system can be used to provide typical MP3 services such as playback control, selection of songs/albums/playlists for playback, creation of playlists, and so forth.

The architecture of the SAMMIE-05 system is roughly based on that of the TRIPS system (Allen et al., 2001), in that it separates functionality between subsystems for interpretation, behavior, and generation. Note that this TRIPS-type architecture pushes many tasks typically included in a dialogue manager (e.g., reference resolution) to the interpretation or generation subsystems. The interface in SAMMIE-05 between interpretation, behavior, and generation is, in fact, the CPS-act intentions described in the last section. The intuition behind the TRIPS architecture is to allow a generic behavioral agent to be built, which can drive the dialogue system's behavior by reasoning at a collaborative task level, and not a linguistic level. The dialogue manager we describe in the next section corresponds to what was called the behavioral component in the TRIPS architecture.

4 The SAMMIE-05 Dialogue Manager

The SAMMIE-05 dialogue manager supports a subset of the CPS model discussed above. It is implemented as a set of production rules in PATE (Pfleger, 2004). In this section, we report our work towards creating a domain-independent dialogue manager based on our model of collaborative problem solving. It is our hope that the CPS model of dialogue sufficiently abstracts dialogue in such a way that the same set of CPS-based update rules could be used for different domains. We do not yet claim to have a domain-independent CPS-based dialogue manager, although we believe we have made progress towards this end.

Because of the limits of the SAMMIE domain (MP3 player control), many parts of the CPS model have not been encoded into the SAMMIE-05 dialogue manager, and consequently, the dialogue manager cannot be shown to be even a "proof of concept" of the value of the CPS model

[1] Although the SAMMIE system was updated in 2006, in this paper, we describe the SAMMIE system as it existed in December 2005, which we will refer to as the SAMMIE-05 system. It is roughly equivalent to the system described in (Becker et al., 2006).

itself. Our purpose here is, rather, to discuss the progress of CPS-based dialogue management and the insights we gained in encoding a dialogue manager in this (relatively) simple domain.

Important parts of the CPS model which are not supported by the SAMMIE-05 dialogue manager include: collaborative planning and replanning, hierarchical plans, recipe selection, goal abandonment, and most evaluation. Support for these has been left for future work. Phenomena that are covered by the system include: goal selection (albeit not collaborative), collaborative slot-filling, plan execution, and limited evaluation (in the form of feasibility and error checking). As MP3 player control consists of relatively fixed tasks, these phenomena were sufficient to model the kinds of dialogues that SAMMIE-05 handled.

In the rest of this section, we will first describe the dialogue manager, and how we attempt to make it domain independent using abstraction in the PS object hierarchy. In the process of building this dialogue manager, we also discovered some types of domain-specific knowledge *outside* the CPS model proper, which are also necessary for the dialogue manager. This is described as well, and then we describe parts of the dialogue manager which are still domain specific.

4.1 High-level Dialogue Management

As with other information state update-based systems, dialogue management in the CPS model can be grouped into three separate processes:

Integrating Utterance Information Here the system integrates CPS negotiation acts (augmented with grounding information)—by both user and system—as they are executed. This is a fairly circumscribed process, and is mostly specified in the details of the CPS model itself. This includes such rules as treating an negotiation action as executed when it has been grounded, or marking an object as committed for a certain role when an *adopt* CPS act is successfully executed. These integration rules are detailed in (Blaylock and Allen, 2005).

Agent-based Control Once utterance content (and its ensuing higher-level action generation) has been integrated into the dialogue model, the system must decide what to do and what to say next. One of the advantages of the CPS model is that it shields such a process from the linguistic details of the exchange. Instead, we attempt to build such behavior on general collaborative problem-solving principles, regardless of what the communication medium is. We describe this phase in more detail below.

Package and Output Communicative Intentions During the first two phases, communicative intentions (i.e., CPS negotiation acts augmented with grounding information) are generated, which the system wants to execute. In this last phase, these communicative intentions are packaged and sent to the generation subsystem for realization. When realization is successfully accomplished, the information state is updated using the rules from the first phase.

The real gain in flexibility and portability from the model comes in the second phase, where the dialogue manager acts more like an autonomous agent in deciding what to do and say next. The information state encodes the agent's commitments (in terms of adopted objectives, etc.), and the current state in the collaborative decision-making process (e.g., which possible objects have been discussed for a certain role). If behavior at this level can be defined on general collaborative problem-solving principles, this would make a precomputed dialogue plan unnecessary. This is a win for both flexibility as well as domain portability.

Most dialogue systems (e.g., GoDiS (Larsson, 2002)) use precomputed *dialogue plans* which define a set of dialogue and domain actions which need to be performed by the system during the dialogue. The need for such an explicit dialogue plan not only adds cost to porting systems, it can also affect the flexibility of the system by restricting the ways it can interact with the user during the dialogue.

4.2 Agent-based Control

The agent-based control phase of dialogue management can be divided into three parts. First, the agent tries to fulfill its obligations in the current dialogue (cf. (Traum and Allen, 1994)). This includes principles like attempting to complete any outstanding negotiations on any outstanding CPS acts or at least further them.

Second, the agent looks over its collaborative commitments (as recorded in the CPS state) and attempts to further them. This includes such principles as trying to execute any actions which have been selected for execution. In the case that an objective cannot be executed because vital information is missing (like a value for a parameter), the system will attempt to further the decision making process at that slot (i.e., try to collaboratively find a value for it).

Lastly, the agent uses its own private agenda to determine its actions.[2]

In the SAMMIE-05 dialogue manager, the first and last phases are handled entirely by rules that refer to only the upper-level of the CPS ontology (e.g., *objectives*, *resources*, and so forth), and thus are not dependent on any domain-specific information. Rules in these phases handle the integration semantics of the CPS acts themselves.

The middle level (*agent-based control*) is, however, where portability can become an issue. It is here where the dialogue manager makes decisions about what to do and say next. In our dialogue manager, we were able to formulate many of these rules such that they only access information at the upper ontology level, and do not directly access domain-specific information. As an example, we illustrate a few of these here.

System Identifies a Resource by Request The following rule is used identify a resource in response to a request by the user: **if** an identify resource is being negotiated, and the resource has not been stated by the user (i.e., this is a request that the system identify the resource), and the system can uniquely identify such a resource given the constraints used to describe it; **then** add the found resource to the object and create a new continue negotiation of the identify resource CPS act, and add this to the queue of responses to be generated.

As can be seen, this rule relies only on the (abstract) information from the CPS model. In the MP3 domain, this rule is used to provide user-requested sets of information from the database (e.g., in response to "Which Beatles albums do you have?"). No domain-specific knowledge is encoded in this rule.

System Prepares an Objective to be Executed The following rule is used when the system marks a top-level objective to be executed next. Note that the current version of the system does not support hierarchical plans, thus the assumption is that this is an atomic action. Also, the system currently assumes that atomic action execution is instantaneous: **if** an objective is in the selected slot (i.e., has been selected for execution) **then** put the objective on a system-internal stack to signal that execution should begin.

This is an example of a simple rule which prepares an *objective* for execution. Similar to the rule just described, no domain-specific information is necessary here—all *objectives* are handled the same, no matter from which domain.

Although we were able to formulate many rules with information available in the CPS model, we encountered some which needed additional information from the domain—including the case where the atomic action execution should actually take place. We now turn our attention to these cases.

4.3 Abstracting Additional Domain Information

In the rules discussed above, simple knowledge implicit in the use of abstract PS objects was sufficient for encoding rules. However, there were a few cases which required more information. In this section, we discuss those cases for which we were able to find a solution in order to keep the rules domain-independent. In the next section, we discuss rules which needed to remain domain-specific, and the reasons for that.

Just because domain information is needed for rules does not mean that we cannot write domain-independent rules to handle them. What is required, however, is the specification of an abstraction for this information, which every new domain is then required to provide.

In the MP3 domain, we have identified two general types of this kind of knowledge. We do not consider this to be a closed list:

Execution Knowledge One of the example rules above showed how the decision to begin execution of an atomic action is made. However, the

[2] Note that this would prototypically be beliefs, desires and intentions, although the CPS model does not require this. The model itself does not place requirements on single agents and how they are modeled, as long as the agents represent the CPS state and are able to interact using it. The agent we are using for the SAMMIE-05 dialogue manager is not an explicitly represented BDI agent, but rather encodes some simple rules about helpful behavior.

actual execution requires knowledge about the domain which is not present in the CPS model (as currently formulated).

In the current system, a domain encodes this information in what we call a *grounded-recipe*, which we have provisionally added as a subtype of *recipe*. A *grounded-recipe* contains a reference to the *objective* it fulfills as well as a pointer to object code (a Java class) which implements an interface for a grounded recipe.

This allows us to write, for example, the following domain-independent rule for atomic action execution in the dialogue manager: **if** an *objective* has been chosen for execution; **then** look up a matching *grounded-recipe* for the *objective* and invoke it (i.e., call the `execute` method of the Java class pointed to in the *grounded-recipe* (passing in the *objective* itself as a parameter)).

Evaluation of PS Objects A more general issue which surfaced was the need to make evaluations of various PS objects in order to decide the system's acceptance/rejection of them within a certain context. Although we believe there is a need to specify some sort of general classification for these, only one such evaluation came up in the MP3 domain.

In deciding whether or not to accept the identification of a fully-specified *objective*, the system needed a way of checking the preconditions of the *objective* in order to detect potential errors. For example, the SAMMIE-05 system supports the deletion of a song from a playlist. Now, grounding-level rules (not detailed here) take care of definite reference errors (e.g., mention of a playlist that does not exist). However, if reference to both objects (the song to be deleted and the playlist) is properly resolved, it is still possible, for example, that the user has asked to delete a song from a playlist when that song is not actually on the playlist. Thus, we needed a way of checking this precondition (i.e., does the song exist on the playlist). Similarly, we needed a way of checking to see if the user has requested playback of an empty playlist (i.e., a playlist that does not contain any songs).

As a simple solution, the dialogue manager uses an abstract interface to allow rules to check conditions of any objective: **if** an identify objective is pending for a fully-specified *objective*, and `CheckPreconditions` fails for the *objective*; **then** add a reject of the identify-resource to the output queue.

4.4 Domain-specific Rules in the System

Despite our best efforts, a few domain-specific update rules are still present in the dialogue manager. We describe one of these here which was used to cover holes which the CPS model did not adequately address. We hope to expand the model in the future so that this rule can also be generalized.

In the MP3 domain, we support the creation of (regular) playlists as well as so-called auto-playlists (playlists created randomly given constraints). Both of these services correspond to atomic actions in our domain and would be theoretically handled by some of the rules for execution described above. However, these are both actions which actually return a value (i.e., the newly-created playlist). This kind of return value is not currently supported by the CPS model. For this reason, we support the execution of both of these actions with special domain-specific rules.

5 Related Work

The work in (Cohen et al., 1991) motivates dialogue as the result of the intentions of rational agents executing joint plans. Whereas their focus was the formal representation of single and joint intentions, we focus on describing and formalizing the interaction itself. We also extend coverage to the entire problem-solving process, including goal selection, planning, and so forth.

Our work is also similar in spirit to work on SharedPlans (Grosz and Sidner, 1990), which describes the necessary intentions for agents to build and hold a joint plan, as well as a high-level sketch of how such joint planning occurs. It defines four operators which describe the planning process: *Select_Rec*, *Elaborate_Individual*, *Select_Rec_GR*, and *Elaborate_Group*. Our CPS acts describe the joint planning process at a more fine-grained level in order to be able to describe contributions of individual utterances. The CPS acts could possibly be seen as a further refinement of the SharedPlans operators. Our model also describes other problem-solving stages, such as joint execution and monitoring.

Collagen (Rich et al., 2001) is a framework for building intelligent interactive systems based on Grosz and Sidner's tripartite model of discourse (Grosz and Sidner, 1986). It provides middleware for creating agents which act as collaborative partners in executing plans using a shared artifact (e.g., a software application). In this sense, it is similar to the work of Cohen and Levesque described above.

Collagen uses a subset of Sidner's artificial negotiation language (Sidner, 1994) to model individual contributions of utterances to the discourse state. The language defines operators with an outer layer of negotiation (e.g., *ProposeForAccept* (*PFA*) and *AcceptProposal* (*AP*)) which take arguments such as *SHOULD(action)* and *RECIPE*. Our interaction and collaborative problem-solving acts are similar in spirit to Sidner's negotiation language, covering a wider range of phenomena in more detail (including evaluations of goals and recipes, solution constraining, and a layer of grounding).

Perhaps the closest dialogue manager to ours is the TRAINS-93 dialogue manager (Traum, 1996), which was based on some very early notions of collaborative problem solving. Its agentive component, the Discourse Actor, was a reactive controller which acted based on prioritized classes of dialogue states (including discourse obligations, user intentions, grounding, and discourse goals). Our rules were not explicitly prioritized, and, although similar in spirit, the dialogue states in TRAINS-93 were represented quite differently from our CPS model.

6 Conclusion and Future Work

We have presented the SAMMIE-05 dialogue manager, which is a first attempt at building a dialogue manager based on collaborative problem solving. Although many parts of collaborative problem solving were not handled by the model, we discussed the extent to which the parts covered were encoded using domain-independent rules based on general principles of collaboration.

There is much future work still to be done. The MP3 player control domain did not exercise large parts of the CPS model, and thus much work remains to be done to fill in the rest of the model. In addition, we have really only scratched the surface in terms of specifying true domain-independent collaborative behavior, including many behaviors which have been detailed in the literature (e.g., (Cohen et al., 1991)). We would like to continue to flesh out this kind of general behavior and add it to the dialogue management rules.

Acknowledgements

We would like to thank the SAMMIE group at Saarland University and DFKI, especially Jan Schehl, Ciprian Gerstenberger, Ivana Kruijff Korbayová and Tilman Becker, for many helpful discussions in the planning and implementation of this work. This work was funded under the EU-funded TALK project (No. IST-507802).

References

James Allen, George Ferguson, and Amanda Stent. 2001. An architecture for more realistic conversational systems. In *Proceedings of Intelligent User Interfaces 2001 (IUI-01)*, pages 1–8, Santa Fe, NM, January.

Tilman Becker, Nate Blaylock, Ciprian Gerstenberger, Ivana Kruijff-Korbayová, Andreas Korthauer, Manfred Pinkal, Michael Pitz, Peter Poller, and Jan Schehl. 2006. Natural and intuitive multimodal dialogue for in-car applications: The SAMMIE system. In *Proceedings of the ECAI Sub-Conference on Prestigious Applications of Intelligent Systems (PAIS 2006)*, Riva del Garda, Italy, August 28–September 1.

Nate Blaylock and James Allen. 2005. A collaborative problem-solving model of dialogue. In Laila Dybkjær and Wolfgang Minker, editors, *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*, pages 200–211, Lisbon, September 2–3.

Nathan J. Blaylock. 2005. Towards tractable agent-based dialogue. Technical Report 880, University of Rochester, Department of Computer Science, August. PhD thesis.

Dan Bohus and Alexander I. Rudnicky. 2003. RavenClaw: Dialog management using hierarchical task decomposition and an expectation agenda. In *Proceedings of Eurospeech-2003*, Geneva, Switzerland.

Philip R. Cohen, Hector J. Levesque, José H. T. Nunes, and Sharon L. Oviatt. 1991. Task-oriented dialogue as a consequence of joint activity. In Hozumi Tanaka, editor, *Artificial Intelligence in the Pacific Rim*, pages 203–208. IOS Press, Amsterdam.

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

Barbara J. Grosz and Candace L. Sidner. 1990. Plans for discourse. In P. R. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication*, pages 417–444. MIT Press, Cambridge, MA.

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University.

Norbert Pfleger. 2004. Context based multimodal fusion. In *Sixth International Conference on Multimodal Interfaces (ICMI'04)*, State College, Pennsylvania.

Charles Rich, Candace L. Sidner, and Neal Lesh. 2001. COLLAGEN: Applying collaborative discourse theory to human-computer interaction. *AI Magazine*, 22(4):15–25. Also available as MERL Tech Report TR-2000-38.

Candace L. Sidner. 1994. An artificial discourse language for collaborative negotiation. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 814–819, Seattle, WA. Also available as Lotus Technical Report 94-09.

David R. Traum and James F. Allen. 1994. Discourse obligations in dialogue processing. In *Proceedings of the 32nd Annual Meeting of the Association for Computational linguistics (ACL-94)*, pages 1–8, Las Cruces, New Mexico.

David R. Traum. 1994. A computational theory of grounding in natural language conversation. Technical Report 545, University of Rochester, Department of Computer Science, December. PhD Thesis.

David R. Traum. 1996. Conversational agency: The TRAINS-93 dialogue manager. In *Proceedings of the Twente Workshop on Language Technology 11: Dialogue Management in Natural Language Systems*, pages 1–11, June.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 99–106.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Implementing the Information-State Update Approach to Dialogue Management in a Slightly Extended SCXML

Fredrik Kronlid
Department of Linguistics and GSLT
Göteborg University
S-405 30 Göteborg
kronlid@ling.gu.se

Torbjörn Lager
Department of Linguistics
Göteborg University
S-405 30 Göteborg
lager@ling.gu.se

Abstract

The W3C has selected Harel statecharts, under the name of SCXML, as the basis for future standards in the area of (multimodal) dialogue systems. The purpose of the present paper is to show that a moderately extended version of SCXML can be used to implement the well-known Information-State Update (ISU) approach to dialogue management. The paper also presents an experimental implementation of Extended SCXML, accessible from a user-friendly web-interface.

1 Introduction

The W3C has selected Harel statecharts (Harel, 1987), under the name of SCXML (Barnett et al., 2007), as the basis for future standards in the area of (multimodal) dialogue systems – replacing a simple and fairly uninteresting "theory of dialogue" (the form-based dialogue modelling approach of VoiceXML) with a theory neutral framework in which different approaches to dialogue modelling could potentially be implemented.[1]

One interesting and influential framework for dialogue management that has evolved over the past years is the so called Information-State Update (ISU) approach, based on the notion of an information state and its update via rules. The purpose of the present paper is to show that, if properly extended, SCXML can be used to implement the ISU approach to dialogue management.

2 SCXML = State Chart XML

SCXML can be described as an attempt to render Harel statecharts in XML. In its simplest form, a statechart is just a deterministic finite automaton, where state transitions are triggered by events appearing in a global event queue.

Just like ordinary finite-state automata, statecharts have a graphical notation. Figure 1 depicts a very simple example.

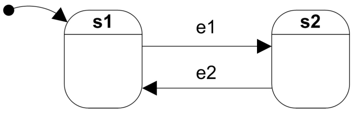


Figure 1: Simple statechart

Any statechart can be translated into a document written in the linear XML-based syntax of SCXML. Here, for example, is the SCXML document capturing the statechart in Figure 1:

```
<scxml initialstate="s1">
  <state id="s1">
    <transition event="e1" target="s2"/>
  </state>
  <state id="s2">
    <transition event="e2" target="s1"/>
  </state>
</scxml>
```

The document can be executed by an SCXML conforming interpreter, an approach aiming at greatly simplifying the step from specification into running dialogue system application.

Harel (1987) also introduced a number of (at the time) novel extensions to finite-state automata, which are also present in SCXML, including:

Hierarchy Statecharts may be hierarchical, i.e. a state may contain another statechart down to an arbitrary depth.

[1] The present paper is based on the February 2007 SCXML working draft.

Concurrency Two or more statecharts may be run in parallel, which basically means that their parent statechart is in two or more states at the same time.

Broadcast communication One statechart S_1 may communicate with another statechart S_2 (running in parallel with S_1) by placing an event in the global event queue that triggers a transition in S_2.

Datamodel SCXML gives authors the ability to define a data model as part of an SCXML document. A data model consists of a `<datamodel>` element containing one or more `<data>` elements, each of which may contain an XML description of data.

For our ISU implementations, we will find uses for all of these features, but will sometimes find is necessary to add a few novel ones as well. Fortunately, SCXML is designed with extensibility in mind (Barnett et al., 2007), and our own investigations suggest that there is indeed room for simple extensions that would increase the expressivity of SCXML even further.

3 The Information State Update Approach to Dialogue Modelling

Simplifying somewhat, the ISU approach to dialogue modelling can be characterized by the following components:

1. An *information state* representing aspects of common context as well as internal motivating factors
2. A set of *dialogue moves* that will trigger the update of the information state
3. A set of declaratively stated *update rules* governing the updating of the information state

The idea of information state update for dialogue modelling is centred around the information state (IS). Within the IS, the current state of the dialogue is explicitly represented. "The term Information State of a dialogue represents the information necessary to distinguish it from other dialogues, representing the cumulative additions from previous actions in the dialogue, and motivating future action" (Larsson and Traum, 2000).

Dialogue moves are meant to serve as an abstraction between the large number of different messages that can be sent (especially in natural language) and the types of updates to be made on the basis of performed utterances (Larsson and Traum, 2000, p. 5). Dialogue moves trigger non-monotonic updates of the IS. Thus, user utterances (or other kinds of user input) are matched against a set of possible update rules that change the IS in the appropriate places (e.g. a new value is entered into a slot). A single user utterance may unleash a whole chain of updates, allowing for generalisations beyond monolithic utterance updates.

The ISU approach should be seen as a rather abstract and relatively "empty" framework that needs to be filled with theoretical content to become a full-fledged theory of dialogue. For example, Larsson (2002) develops and implements a theory of Issue-Based Dialogue Management, taking Ginzburg's (1996) concept of Questions Under Discussion (QUD) as a starting point. QUD is used to model raising and addressing issues in dialogue (including the resolution of elliptical answers). Issues can also be raised by addressing them, e.g. by giving an answer to a question that has not be explicitly asked (question accommodation).

Two well-known implementations of the ISU approach to dialogue management are TrindiKit (Larsson and Traum, 2000) and DIPPER (Bos et al., 2003). Implemented/embedded in Prolog and relying to a large extent on properties of its host language, TrindiKit was the first implementation of the ISU approach. DIPPER is built on top of the Open Agent Architecture (OAA), supports many off-the-shelf components useful for spoken dialogue systems, and comes with a dialogue management component that borrows many of the core ideas of the TrindiKit, but is "stripped down to the essentials, uses a revised update language (independent of Prolog), and is more tightly integrated with OAA" (Bos et al., 2003). Other implementations exist, but TrindiKit and DIPPER are probably the most important ones.

4 Implementing ISU in SCXML

We suggest that most systems implementing the ISU approach to dialogue management can be reimplemented in (Extended) SCXML, exploiting the mapping between the ISU components and SCXML elements depicted in Table 1.

Of course, we cannot really prove this claim, but by taking a simple example system and reim-

The ISU Approach	SCXML
Information state	Datamodel
Dialogue move	Event
Update rule	Transition

Table 1: From ISU into Extended SCXML

plement it in SCXML we hope to be able to convince the reader of the viability of our approach. We choose to target the IBiS1 system from (Larsson, 2002), and thus most of our discussion will be comparing TrindiKit with SCXML, but we also hint at how DIPPER compares with SCXML. As we shall see, our conclusion is that SCXML could potentially replace them both.

4.1 Information states as datamodels

The expressivity of the SCXML `<datamodel>` is perfectly adequate for representing the required kind of information structures. A typical IBiS1 information state may for example be represented (and initialised) as follows:

```
<datamodel>
  <data name="IS">
    <private>
      <agenda>{New Stack init}</agenda>
      <plan>{New Stack init}</plan>
      <bel>{New Set init}</bel>
    </private>
    <shared>
      <com>{New Set init}</com>
      <qud>{New Stack init([q])}</qud>
      <lu>
        <speaker>usr</speaker>
        <move>ask(q)</move>
      </lu>
    </shared>
  </data>
</datamodel>
```

Here, the datamodel reflects the distinction between what is private to the agent that 'owns' the information state, and what is shared between the agents engaged in conversation. Note that `IS.shared.qud` points to a stack with `q` on top, indicating that it is known by both parties that the question `q` is "under discussion".[2]

4.2 Dialogue moves as SCXML events

The closest SCXML correlate to a dialogue move is the notion of an *event.* An SCXML event has a *name*, and an optional *data payload*. The (current) SCXML draft does not represent events formally, but for the purpose of the present paper we will represent them as records with a label (for representing their name) and a set of feature-value pairs (for representing the data payload). An ASK move where a speaker `a` is asking a question `q` may thus be represented as: `says(speaker:a move:ask(q))`

4.3 Update rules as transitions

A TrindiKit ISU-style update rule consists of a set of *applicability conditions* and a set of *effects* (Larsson and Traum, 2000, p. 5), and a collection of such rules forms what is essentially a system of condition-action rules – a *production system.* While SCXML is easily powerful enough to implement such a system, the expressivity of the language for stating the conditions is not adequate for our purpose, since there is no mechanism in place for carrying information (i.e. information dug up from the IS) from the conditions over to the actions. This is where we are suggesting a small extension. We propose that a `pcond` attribute be added to the `<transition>` element, the value of which is a Prolog style query rather than an ordinary boolean expression, i.e. a query that evaluates to true of false (just like an ordinary boolean expression) but which will possibly also bind variables if evaluated to true. We suggest that the names of these variables be declared in a new attribute `vars`, and that the values of them are made available in the actions of the `<transition>`.

For example, an update rule written in the following way in the Prolog-based TrindiKit notation

```
rule( integrateSysAsk,
    [ $/shared/lu/speaker = sys,
      $/shared/lu/move = ask(Q)],
    [ push( /shared/qud, Q ) ] ).
```

may be written as follows in Extended SCXML:

```
<transition vars="Q"
            pcond="IS.shared.lu.speaker=sys
                   IS.shared.lu.move=ask(Q)"
            target="downdateQUD">
  <script>{IS.shared.qud push(Q)}</script>
</transition>
```

4.4 The update algorithm as a statechart

Dialogue management involves more than one rule, and the application of the rules needs to be controlled, so that the right rules are tried and applied at the right stage in the processing of a dialogue. Furthermore, we require three *kinds* of rules: 1) rules that perform unconditional maintenance operations on the datamodel (representing

[2] We use `q` and `r` here as placeholders for a question and a response, respectively.

the information state), 2) rules that enable events (representing dialogue moves) to update the datamodel, and 3) rules that when triggered by certain configurations of the datamodel updates it, i.e. changes its configuration. (The above example is of the third kind.)

Here is an example of the first kind of rule, responsible for first clearing the agenda, and then transferring to the grounding state:

```
<state id="init">
  <transition target="grounding">
    <script>
      {IS.private.agenda clear}
    </script>
  </transition>
</state>
```

(We shall return to the significance of the enclosing state further down.) For an example of the second kind of rule we offer:

```
<state id="grounding">
  <transition event="says"
              target="integrate">
    <assign location="IS.shared.lu.move"
            expr="Eventdata.move"/>
    <assign location="IS.shared.lu.speaker"
            expr="Eventdata.speaker"/>
  </transition>
</state>
```

This rule provides a bridge between the events representing dialogue moves and the datamodel representing the IS. If an event of the form says(speaker:sys move:answer(r)) appears first in the event queue when the statechart is in state grounding, the rule will set IS.shared.lu.move to the value answer(r) and IS.shared.lu.speaker to sys, and then a transfer to the state integrate will take place. In this state, three transitions representing update rules of the third kind are available:

```
<state id="integrate">
  <transition vars="Q"
              pcond="IS.shared.lu.speaker=sys
                     IS.shared.lu.move=ask(Q)"
              target="downdateQUD">
    <script>{IS.shared.qud push(Q)}</script>
  </transition>
  <transition vars="Q"
              pcond="IS.shared.lu.speaker=usr
                     IS.shared.lu.move=ask(Q)"
              target="downdateQUD">
    <script>
      {IS.shared.qud push(Q)}
      {IS.private.agenda push(respond(Q))}
    </script>
  </transition>
  <transition vars="Q R"
              pcond="IS.shared.lu.move=answer(R)
                     {IS.shared.qud top(Q)}
                     {Domain.relevantAnswer Q R}"
              target="downdateQUD">
    <script>{IS.shared.com add(Q#R)}</script>
  </transition>
</state>
```

The transitions are tried in document order and given the current datamodel the last one will be the one chosen for execution. Its effect is that q#r (i.e. the pair of q and r, representing a proposition) will be added to the set at IS.shared.com i.e. the set of beliefs that the user and system shares (or "the common ground"). Thereafter a transition to the state downdateQUD will take place:

```
<state id="downdateQUD">
  <transition vars="Q R"
              pcond="{IS.shared.qud top(Q)}
                     {Domain.relevantAnswer Q R}
                     {IS.shared.com member(Q#R)}"
              target="load_plan">
    <script>{IS.shared.qud pop}</script>
  </transition>
  <transition target="load_plan"/>
</state>
```

In this state, either the first of its transitions will trigger, first popping the QUD and then leading to the load_plan state, or else the second transition will trigger, also leading to load_plan, but this time without popping the QUD. That is, the state will *try* to downdate the QUD. Given the current configuration of the datamodel in our example, the first rule will trigger, the element on top of the stack at IS.shared.qud will be popped, and (the relevant part of) the datamodel end up as follows:[3]

```
<datamodel>
  <data name="IS">
    <private>
      <agenda>[]</agenda>
      <plan>[]</plan>
      <bel>{}</bel>
    </private>
    <shared>
      <com>{q#r}</com>
      <qud>[]</qud>
      <lu>
        <speaker>sys</speaker>
        <move>answer(r)</move>
      </lu>
    </shared>
  </data>
</datamodel>
```

Note how the underlying DFA 'backbone' controls when certain classes of rules are eligible for execution. In statechart notation, the relevant statechart can be depicted as in Figure 2.[4] By comparison, in TrindiKit the control of the application of update rules is handled by an *update algorithm* written in a procedural language designed for this purpose.

[3] Here, [] and {} indicate the empty stack and the empty set, respectively.

[4] The details of the load_plan and exec_plan states may be found in our web-based demo.

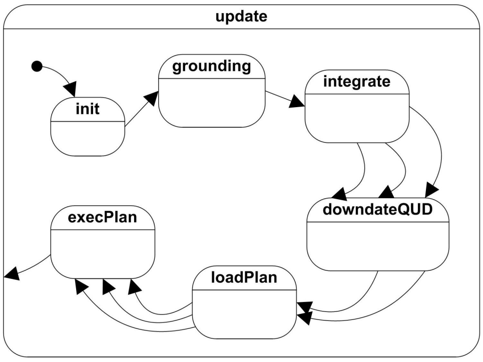


Figure 2: Update statechart

The update algorithm (or a version of it) used by IBiS1 is shown here:

```
if not LATEST_MOVES == failed
then ( init,
       grounding,
       integrate,
       try downdate_qud,
       try load_plan,
       repeat exec_plan )
```

Note that the statechart in Figure 2 does basically the job of this algorithm. Terms like "init", "grounding", "integrate", "downdate_qud", etc. refer to TrindiKit *rule classes*. In our statechart, they correspond to states.

4.5 Implementing modules as statecharts

The update statechart in Figure 2 basically corresponds to the *update module* in IBiS1, responsible for updating the information state based on observed dialogue moves. There is also a *select module* in IBiS1, responsible for selecting moves to be performed, which space does not allow us to go into detail about here (but see our web-based demo).

Together, the update module and the select module forms the *Dialogue Move Engine* (DME) – the dialogue manager proper. As can be seen in Figure 3, DME processing starts in the select state and then alternates between update and select.

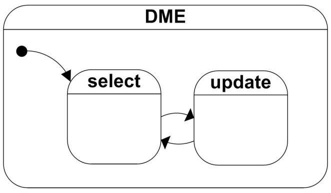


Figure 3: The Dialogue Move Engine

4.6 Interpretation and generation

SCXML is not supposed to directly interact with the user. Rather, it requests user interaction by invoking a *presentation component* running in parallel with the SCXML process, and communicating with this component through asynchronous events. Presentation components may support modalities of different kinds, including graphics, voice or gestures. Concentrating on presentation components for spoken language dialogue (a.k.a. "voice widgets") we may assume that they include things like a TTS component for presenting the user with spoken information and an ASR component to collect spoken information from the user.

For example, our interpretation module may invoke an ASR component, like so:[5]

```
<state id="interpret">
  <invoke targettype="vxml"
          src="grammar.vxml#main"/>
</state>
```

and our generation module may invoke a TTS component as follows:

```
<state id="generate">
  <invoke targettype="vxml"
          src="generate.vxml#prompt"/>
</state>
```

4.7 The dialogue system statechart

The TrindiKit architecture also features a *controller*, wiring together the other modules necessary for assembling a complete dialogue system, either in sequence or through some asynchronous (i.e. concurrent) mechanism (Larsson, 2002). We choose here to exemplify an asynchronous architecture, taking advantage of the concurrency offered by SCXML. The statechart corresponding to a full dialogue system might look like in Figure 4.

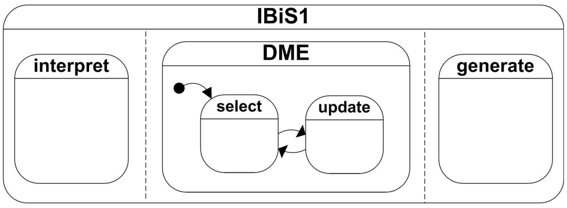


Figure 4: Parallel control

The dashed lines show – using standard statechart graphical notation – that the interpretation mod-

[5] Here we use VoiceXML for our example presentation components. This is not optimal, but we take comfort in the fact that the next major version of VoiceXML (known as V3) will be redesigned from the bottom and up with uses like these in mind.

ule, the DME and the generation module are run in parallel. In SCXML the full dialogue system may be sketched as follows:

```
<parallel id="IBiS1">
   <state id="interpret" .../>
   <state id="DME">
     <initial>
       <transition target="select"/>
     </initial>
     <state id="select" .../>
     <state id="update" .../>
   </state>
   <state id="generate" .../>
</parallel>
```

Communication between the modules of the system – between the interpreter, generator and DME – is performed in the broadcast style supported by SCXML, by letting one module place events in the global event queue – events to be picked up by another module. Comparing SCXML and TrindiKit, we note that the SCXML notion of an event queue seems to do the job of TrindiKit's *module interface variables* (MIVs), which is exactly this – to enable modules to interact with each other.

4.8 From TrindiKit to SCXML: a summary

In Table 2, we summarize the relevant correspondences between TrindiKit and our SCXML formalization of the ISU approach to dialogue management.

TrindiKit	**SCXML**
Information state	Datamodel
Dialogue move	Event
Module interface vars	Event queue
Update rule	Transition
Rule class	State (simple)
Update algorithm	State (complex)
Module	State (complex)
Control algorithm	State (complex)

Table 2: From TrindiKit into Extended SCXML

We note that SCXML is considerably more simple than TrindiKit, in that rule classes, update algorithms, modules and control algorithms are all represented as (simple or complex) states/statecharts.

4.9 From DIPPER to SCXML

(Bos et al., 2003) illustrate the DIPPER architecture and information state update language with an example which implements a "parrot", where the system simply repeats what the user says. These are the information state and the relevant update rules, in DIPPER notation:

```
is:record([input:queue(basic),
           listening:basic,
           output:queue(basic)]).

urule(timeout,
      [first(is^input)=timeout],
      [dequeue(is^input)]).

urule(process,
      [non_empty(is^input)],
      [enqueue(is^output,first(is^input)),
       dequeue(is^input)]).

urule(synthesise,
      [non_empty(is^output)],
      [solve(text2speech(first(is^output)),[]),
       dequeue(is^output)]).

urule(recognise,
      [is^listening=no],
      [solve(X,recognise('.Simple',10),
              [enqueue(is^input,X),
               assign(is^listening,no)]),
       assign(is^listening,yes)]).
```

Here is our translation into SCXML:

```
<scxml initialstate="process">
  <datamodel>
    <data name="IS">
      <input>{New Queue init}</input>
      <output>{New Queue init}</output>
    </data>
  </datamodel>
  <state id="process">
    <transition cond="{IS.input first($)}==timeout">
      <script>
        {IS.input dequeue}
      </script>
    </transition>
    <transition cond="{Not {IS.input isEmpty($)}}">
      <script>
        {IS.output enqueue({IS.input first($)})}
        {IS.input dequeue}
      </script>
    </transition>
    <transition cond="{Not {IS.output isEmpty($)}}">
      <send event="speak"
            expr="{IS.output first($)}"/>
      <script>
        {IS.output dequeue}
      </script>
    </transition>
    <transition target="listening"/>
  </state>
  <state id="listening">
    <onentry>
      <send event="recognise"/>
    </onentry>
    <transition event="recResult" target="process">
      <script>
        {IS.input enqueue(Eventdata)}
      </script>
    </transition>
  </state>
</scxml>
```

We shall use this example as our point of departure when comparing DIPPER, SCXML and TrindiKit. First, we note that DIPPER uses the *solveables* of OAA for the purpose of enabling modules to interact with each other. In the case of the fourth rule above, a solvable is sent to the OAA agent responsible for speech recognition, which within 10 seconds will bind the variable `X` to either the recognition result or to the atom `timeout`. This value of `X` will then be added to the input queue. Our SCXML version works in a similar fashion. An event `recognise` is sent in order to activate the speech recognition module, and a transition is triggered by the `recResult` event returned by this module. The `Eventdata` variable will be bound to the recognition result.

Secondly, in the DIPPER rule set, an information state field 'listening' is used (as we see it) to simulate a finite state automaton with two states `listening=yes` and `listening=no`. The idea is to control the application of the fourth rule – it is meant to be applicable only in the 'state' `listening=no`. The general strategy here appears to be to take advantage of the fact that a production system can easily simulate a finite state automaton. DIPPER can thus eliminate the need for an update algorithm in the style of TrindiKit, but at the expense of complicating the rules.

Note that the 'listening' field is not required in the SCXML version, since we can use two "real" states instead. Indeed, looking at TrindiKit, DIPPER and SCXML side by side, comparing TrindiKit's use of an update algorithm, DIPPER's DFA simulation 'trick', and SCXML's use of real states, we think that SCXML provides the neatest and most intuitive solution to the problem of controlling the application of update rules.

Finally, few (if any) extensions of SCXML appear to be needed in order to reconstruct DIPPER style dialogue managers in SCXML. This is mainly due to the fact that DIPPER does not make use of Prolog style conditions the way TrindiKit does. Whether the availability of Prolog style conditions in this context is crucial or not is, in our opinion, still an open question.

5 A More Abstract Point of View

In a recent and very interesting paper Fernándes and Endriss (2007) present an hierarchy of abstract models for dialogue protocols that takes as a starting point protocols based on deterministic finite automata (DFAs) and enhances them by adding a 'memory' in the form of an instance of an abstract datatype (ADT) such as a stack, a set or a list to the model. They show that whereas a DFA alone can handle only simple dialogue protocols and conversational games, a DFA plus a set can handle also for example the representation of a set of beliefs forming the common ground in a dialogue, a DFA plus a stack is required if we want to account for embedded subdialogues, questions under discussion á la Ginzburg, etc., and a DFA plus a list is needed to maintain an explicit representation of dialogue history.

Space does not allow us to give full justice to the paper by Fernándes and Endriss here. We only wish to make the point that since an SCXML state machine at its core can be seen as just a fancy form of a DFA, and since SCXML does indeed allow us to populate the datamodel with instances of ADTs such as stacks, sets and list, it seems like SCXML can be regarded as a concrete realization very "true to the spirit" of the more abstract view put forward in the paper (and more true to this spirit than TrindiKit or DIPPER). Having said this, we hasten to add that while we think that the DFA core of SCXML is well-designed and almost ready for release, the datamodel definitely needs more work, and more standardization.

6 An SCXML Implementation

We have built one of the first implementations of SCXML (in the Oz programming language, using Oz as a scripting language). A web interface to a version of our software – called Synergy SCXML – is available at <www.ling.gu.se/˜lager/Labs/SCXML-Lab/>. Visitors are able to try out a number of small examples (including a full version of the SCXML-IBiS1 version described in the present paper) and are also able to write their own examples, either from scratch, or by modifying the given ones.[6]

7 Summary and Conclusions

We summarize by highlighting what we think are the strong points of SCXML. It is:

- **Intuitive**. Statecharts and thus SCXML are based on the very intuitive yet highly abstract notions of *states* and *events.*

- **Expressive**. It is reasonable to view SCXML as a multi-paradigm programming language, built around a declarative DFA core, and extended to handle also imperative, event-based and concurrent programming.

- **Extensible**. SCXML is designed with extensibility in mind (Barnett et al., 2007), and our own investigations suggest that there is indeed room for simple extensions that will

[6] Our implementation is not the only one. *Commons* SCXML is an implementation aimed at creating and maintaining an open-source Java SCXML engine, available from <http://jakarta.apache.org/commons/scxml/>. There are most likely other implementations in the works.

increase the expressivity of SCXML considerably.

- **Theory neutral**. Although it is clear that the framework is suitable for implementing both simple DFA-based as well as form-based dialogue management, the framework as such is fairly theory neutral.

- **Visual**. Just like ordinary finite-state automata, statecharts have a graphical notation – for "tapping the potential of high bandwidth spatial intelligence, as opposed to lexical intelligence used with textual information" (Samek, 2002).

- **Methodologically sound.** The importance of support for refinement and clustering should not be underestimated. In addition, the fact that SCXML is closely aligned to statechart theory and UML2 will help those using model driven development methodologies.

- **XML enabled**. Thus, documents may be validated with respect to a DTD or an XML Schema, and there are plenty of powerful and user friendly editors to support the authoring of such documents.

- **Part of a bigger picture**. SCXML is designed to be part of a framework not just for building spoken dialogue systems, but also for controlling telephony – a framework in which technologies for voice recognition, voice-based web pages, touch-tone control, capture of phone call audio, outbound calling (i.e. initiate a call to another phone) all come together.

- **Endorsed by the W3C**. The fact that SCXML is endorsed by the W3C may translate to better support in tooling, number of implementations and various runtime environments.

We conclude by noting that despite the fact that SCXML was not (as far as we know) designed for the purpose of implementing the ISU approach to dialogue management, it is nevertheless possible to do that, in the style of TrindiKit (provided the proposed rather moderate extensions are made) or in the style of DIPPER. Indeed, we believe that SCXML could potentially replace both TrindiKit and DIPPER.

All in all, this should be good news for academic researchers in the field, as well as for the industry. Good news for researchers since they will get access to an infrastructure of plug-and-play platforms and modules once such platforms and modules have been built (assuming they *will* be built), good news for the industry since a lot of academic research suddenly becomes very relevant, and good news for the field as a whole since SCXML appears to be able to help bridging the gap between academia and industry.

8 Acknowledgements

A preliminary version of this paper has been presented in a seminar at the Centre for Language Technology (CLT) in Göteborg. We are grateful to the members of this seminar for reading and discussing the paper, and for proposing useful additions and improvements.

References

Jim Barnett et al. 2006. State Chart XML (SCXML): State Machine Notation for Control Abstraction. http://www.w3.org/TR/2007/WD-scxml-20070221/.

Johan Bos, Ewan Klein, Oliver Lemon and Tetsushi Oka. 2003. DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture. In *4th SIGdial Workshop on Discourse and Dialogue*, ACL, Sapporo.

Raquel Fernández and Ulle Endriss. 2007. Abstract Models for Dialogue Protocols. In *Journal of Logic, Language and Information* vol. 16, no. 2, pp. 121-140, 2007.

Jonathan Ginzburg. 1996. Interrogatives: Questions, Facts and Dialogue. In S. Lappin (ed.): *Handbook of Contemporary Semantic Theory*, Blackwell.

David Harel. 1987. Statecharts: A Visual Formalism for Complex Systems. In *Science of Computer Programming 8*, North-Holland.

Staffan Larsson and David Traum. 2000. Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. In *Natural Language Engineering*, vol. 6, no. 3-4, pp. 323-340, 2000.

Staffan Larsson. 2002. *Issue-Based Dialogue Management.* Ph.D. thesis, Göteborg University.

Miro Samek. 2002. *Practical Statecharts in C/C++*. CMPBooks.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, page 107.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Recent advances in spoken language understanding

Renato De Mori
School of Computer Science, Mc Gill University, Canada
and
Laboratoire d'Informatique, Université d'Avignon, France

This presentation will review the state of the art in spoken language understanding.

After a brief introduction on conceptual structures, early approaches to spoken language understanding (SLU) followed in the seventies are described. They are based on augmented grammars and non stochastic parsers for interpretation.

In the late eighties, the Air Travel Information System (ATIS) project made evident problems peculiar to SLU, namely, frequent use of ungrammatical sentences, hesitations, corrections and errors due to Automatic Speech Recognition (ASR) systems. Solutions involving statistical models, limited syntactic analysis, shallow parsing, were introduced.

Automatic learning of interpretation models, use of finite state models and classifiers were also proposed; Interesting results were found in such areas as concept tags detection for filling slots in frame systems, conceptual language models, semantic syntax-directed translation, stochastic grammars and parsers for interpretation, dialog event tagging.

More recent approaches combine parsers and classifiers and reconsider the use of probabilistic logics. Others propose connectionist models and latent semantic analysis.

As interpretation is affected by various degrees of imprecision, decision about actions should depend on information states characterized by the possibility of having competing hypotheses scored by confidence indicators. Proposed confidence measures at the acoustic, linguistic and semantic level will be briefly reviewed.

Applications, portability issues and the research agenda of the European project LUNA will be described.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 109–116.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Coordinating on ad-hoc semantic systems in dialogue

Staffan Larsson
Dept. of linguistics
Göteborg University, Sweden
sl@ling.gu.se

Abstract

An exploratory study of a Map Task dialogue indicates that dialogue participants coordinate on an ad-hoc vocabulary and associated concepts (meanings) to enable information exchange, and that ad-hoc vocabularies can be cobbled together from a heterogeneous mix of "micro-vocabularies" borrowed from various other (a priori unrelated) domains. To account for these observations, we sketch a basic framework for formalising the process of coordination of semantic systems in dialogue, and relate this framework to some interactional processes of semantic coordination in dialogue, such as feedback, negotiation and accommodation.

1 Vocabulary in a Map Task dialogue

In the Map Task corpus[1], a GIVER explains a route, provided on the giver's map, to a FOLLOWER who has a similar (but slightly different) map but with no route marked. A map contains landmarks portrayed as labelled line drawings. In a route-giving task like that recorded in the Map Task corpus, expressions referring to landmarks, compass directions etc. can be a priori expected as a kind of "prototype" devices for talking about maps. A typical utterance may look as follows[2]:

GIVER: right **a camera shop**, right, head due **south** ... from that just ... **down** for about **twelve centimetres**, have you got **a parked van** at the bottom ?

Here, we may note two constructions expressing direction ("south", "down"), one expressing a distance ("twelve centimetres") and two referring to landmarks ("a camera shop", "a parked van"). A further example:

GIVER: go round the left hand side of the camera shop ... in between **the edge of the page** and the camera shop.

Whereas the previous expressions were completely expected given the general direction-giving task, the reference to an absolute position using "the edge of the page" is perhaps less expected. Clearly, this is a consequence of the dialogue participants (DPs) talking about a (paper) map rather than e.g. about some actual terrain.

GIVER: so you're ... you're going diagonally sort of north ... northeast ... it's not it's it's a sort of **two o'clock** almost **three o'clock** ... from the allotments ... over

Here, we have GIVER referring to map directions using the expressions "two o'clock" and "three o'clock". This is most likely an everyday variant of the practice of English-speaking pilots of using "o'clock" for directions[3]. Let's look at a final excerpt:

GIVER: right, you go ... down the side of the camera shop right for about twelve centimetres ... and do a sort of **a "u" shape** ... for and **the bottom of the "u" shape** should be about three centimetres long, right do you know what i'm meaning

[1] http://www.hcrc.ed.ac.uk/maptask/maptask-description.html

[2] The following excerpts are taken from Map Task dialogue q4nc4, available at the Map Task web site.

[3] Note the use of a hedging "sort of" before "two o'clock", which seems to indicate that the speaker is slightly unsure as to whether the following expression is quite appropriate. A similar observation is made by Brennan (To appear) (p. 11): "[h]edges seem to be one way of marking that a referring expression is provisional."

...

GIVER: you've worked it out already , eh we're doing **a "u" shape** round the parked van but it's a sort of three cent– see if you imagine a 'u' right ... **the stems of the "u" the ... vertical bits** are sort of three centimetres between

First, a trajectory is referred to using the expression "a 'u' shape". This trajectory is (or so we argue) then reified as an imagined 'u'-shape on the map, now acting more akin to a landmark with a concrete (if invisible) shape, size and even component parts ("the ... vertical bits"; "the stems of the 'u' ").

2 Micro-vocabularies used in Map Task dialogue

Based on the above excerpts (and others from the same dialogue), we are now able to provide a very tentative inventory of referring expressions used by GIVER and FOLLOWER in the Map Task dialogue. DPs refer to distances, absolute and relative locations, directions, and trajectories. Below, we list the sub-types of expressions used for each basic class.

- distances on page, in centimetres ("about twelve centimetres")
- absolute locations
  - landmarks ("the camera shop")
  - page edges ("the edge of the page"; "at the bottom"; "the far right-hand side")
  - typography on page ("the words 'yacht club'")
  - (imagined) letter shapes ("the bottom of the 'u' shape"; "the stems of the 'u' the ... vertical bits")
- relative locations
  - relative to landmark ("left hand side of (landmark)")
  - relative to sheet of paper ("the other side of the page")
- directions
  - compass directions ("head due south")
  - left, right, up, down, diagonally, etc.
  - clock directions ("sort of two o'clock")
- trajectories
  - imagined/drawn lines ("a straight line up the ...")
  - letter shapes as trajectories ("do sort of a 'u' shape")

3 Interleaving resource registers

How can we account for this diversity in the range of linguistic expressions used in a simple direction-giving dialogue? In this section, we will propose a basic terminology intended to form a basis for a formal account of what we see happening in dialogues such as the one quoted above.

3.1 Perspectives

In the Map Task dialogue, the DPs need to coordinate on a way of talking about the map. What the above excerpts show is that there are several ways of talking about a map; this is also shown in the Maze Game experiments (Garrod and Anderson, 1987; Healey, 1997) where DPs alternative between an abstract "coordinate system" perspective on a maze ("Go to the fourth row down and the second from the right"; "Six three"), and more concrete perspectives involving e.g. corridors ("Go forward, then turn left at the junction") or shapes ("the bit sticking out on the right"). In our view, a way of talking about X involves *taking a perspective*[4] on X and selecting a vocabulary associated with that perspective. Taking a perspective P on subject matter X in dialogue involves an analogue - "talking about X as P" - e.g. talking about directions on a map as clock arms. Different perspectives have different advantages and disadvantages; for example, an abstract perspective is compact but error-prone; a clock perspective on directions may e.g. enable shorter utterances. One plausible reason for interleaving and switching several perspectives and associated vocabularies thus seems to be that it increases the efficiency of communication.

3.2 Resource and ad-hoc registers

On a fundamental level, we believe that a language can be regarded as consisting of a multitude of activity-specific "language games" involving activity-specific *registers*. A register is a an *activity-specific semantic system* (a "micro-language"), consisting minimally of a set of linguistic signs, i.e., linguistic expressions and associated concepts (meanings)[5]. In dialogue, registers may be used as *resources* which can be borrowed or appropriated into a new activity and

[4] Garrod and Anderson (1987) and Healey (1997) instead talk about adopting "description types".

[5] A *compositional* register will more generally contain *mappings* between expressions and meanings.

adapted to the domain at hand. Putting it differently, an *ad-hoc register* is assembled to be able to talk about some subject matter from one or more perspectives. In the map-task dialogue, several different resource registers are introduced and accepted[6]; often, both introduction and acceptance are implicit, but sometimes verbal signals (including feedback) are used to manage semantic coordination. For example, one could imagine “sort of” being used to signal introduction of new register.

As mentioned, in the Map Task dialogue we find some resource registers that can be regarded as “standard” or “default” ways of talking about maps, whereas others are more unexpected. First, the standard map registers subsumes (1) a *landmarks* register provided to DPs as pictures and text on map, (2) a *compass directions* register, and (3) a *(metric) distance* register. The non-standard parts of the ad-hoc register are:

- *clock* register: map directions as clock hands ”two o’clock” etc.
- *sheet-of-paper* register perspective: map as a sheet of paper edges of page distances on page relations between pages (e.g. ”opposing page”)
- *letter shape* register perspective: Viewing map as a piece of paper where letter shapes can be drawn letter shapes (”a ’u’ shape”) parts of letter shapes (”stems”)

3.3 Appropriating and interleaving registers

To describe the dynamics of registers in the above dialogue, we can say that the clock, sheet-of-paper and letter-shape registers are *appropriated* into the map task activity, where it is *interleaved* with landmark, compass direction, and metric distance registers to form an ad-hoc register[7]. This involves adapting the meanings associated with resource register vocabularies to the current situation.

4 Meaning potentials

To describe how linguistic expressions can be interactively (in dialogue) appropriated into a new activity, we need an account of semantics which (1) allows several activity-specific meanings for a single expression, and (2) allows open and dynamic meanings which can be modified as a consequence of language use. The received view in formal semantics (Kaplan, 1979) assumes that there are abstract and context-independent “literal” meanings (utterance-type meaning; Kaplan’s “character”) which can be regarded formally as functions from context to content; on each occasion of use, the context determines a specific content (utterance-token meaning). Abstract meanings are assumed to be static and are not affected by language use in specific contexts. Traditional formal semantics is thus ill-equipped to deal with semantic coordination, because of its static view of meaning.

We believe that the idea of “meaning potentials” may offer a more dynamic view of meaning. The term originates from “dialogical” approaches to meaning (Recanati, 2003). On the “dialogical” view, language is essentially dynamic; meaning is negotiated, extended, modified both in concrete situations and historically. Interaction and context are essential for describing language, and there is a general focus on the context-dependent nature of meaning. Linguistic expressions have meaning potentials, which are not a fixed and static set of semantic features, but a dynamic potential which can give rise to different situated interpretations. Different contexts exploit different parts of the meaning potential of a word.

We refer to the dynamic aspect meaning potentials as *semantic plasticity*. Semantic plasticity will be central to our account of how activity-specific abstract[8] meanings are updated and gradually change as a consequence of use.

5 Towards a formalisation of semantic plasticity and meaning potentials

To describe in more detail how DPs coordinate on registers (e.g. when adapting a resource register to a new domain), we need a dynamic account of meanings and registers allowing incremental modifications (updates) to semantic systems. We also need a description of possible dialogue strategies for register coordination. Describing this process *formally* requires formalising the dynamics of registers and meaning potentials, and the dia-

[6] Often, several resource registers are used in a single phrase, as e.g. in ”in between the edge of the page and the camera shop”.

[7] This “interleaving strategy” can be compared with the “switching strategies” evident in maze game experiments (Healey, Garrod), where speakers switch between perspectives (description types). Presumably, both interleaving and switching are possible.

[8] We use “abstract meaning” to refer to utterance-type meanings, either activity-specific or activity-independent.

logue protocols involved in negotiating semantic systems. In this section, we will take some initial steps towards this goal by sketching a formal account of semantic plasticity.

We propose to regard the meaning of a linguistic construction or word[9] to depend on previous uses of that word. This makes it possible to model how meanings change as a result of using language in dialogue. The basic idea is that speakers have internalised (potentially complex) dispositions, or *usage patterns*, governing the use of specific words. These dispositions depend, among other things, on observations of previous situations where the word in question has been used, and on specific generalisations over these situations.

Semantic plasticity is described in terms of updates to individual usage patterns associated with words (in general, linguistic constructions) triggered by observations of their use in dialogue. When a usage pattern $[c]$ is sufficiently coordinated[10] (shared) within a community, we will talk about $[c]$ as the meaning potential of a word c. By modelling plasticity of usage patterns of individuals, we thus indirectly model semantic plasticity in a linguistic community.

5.1 Usage sets and usage patterns

To get a handle on semantic plasticity, we will start by positing for each language user A and word c a *usage-set*[11] S_c^A containing all situations where A has observed a use (token) of c. Formally, $S_c^A = \{s \mid A$ has observed a use of c in situation $s\}$. This should be regarded merely as an abstract theoretical entity. .

We assume that A generalises over S_c^A; this generalisation we call the usage pattern (or usage disposition) $[c]^A$. In cognitive terms one can think of the usage pattern as the "memory trace" of observed uses of c.

That c has been used in a situation simply means that someone has uttered a token of c in that situation[12].

5.2 Situated meanings and interpretations

On each occasion of use of c in situation s, c has a specific situated utterance-token meaning which derives partly from the shared abstract utterance-type meaning (meaning potential) $[c]$ and partly from s. We write this meaning formally as $[c]_s$. The subjective counterpart of a situated meaning is a *situated interpretation*, written as $[c]_s^A$ for an agent A; this is the interpretation that A makes of c in s based on A's usage pattern $[c]^A$. A situated meaning $[c]_s$ arises in a situation when the DPs in s make sufficiently similar situated interpretations of c in s.

5.3 Appropriate and non-appropriate uses

We will assume that new uses of a word c can be classified as appropriate or inappropriate given an existing usage pattern[13] for c[14]. The formal notation we will use to express that a use of c in situation s is appropriate with regard to A's usage pattern for c is $[c]^A \vdash s$. Correspondingly, $[c]^A \nvdash s$ means that s is not an appropriate situation in which to use c given $[c]^A$[15].

On the whole, if a token of c uttered in a situa-

[9] Although we intend this account to cover not only words but also other constructs phrases, syntactic categories, and other linguistic elements, we will henceforth (for simplicity) use "word" instead of "linguistic construction".

[10] Roughly, a usage pattern connected to an expression is sufficiently coordinated in a community when speakers and hearers are able to use that expression to exchange information sufficiently to enable them to achieve their shared and private goals. For example, in the Map Task dialogues an expression is sufficiently coordinated when DPs are able to make use of it in carrying out the route-giving tasks assigned to them.

[11] An alternative term is *situation-collocation*.

[12] It is important to point out that the notion of "situation" we are using here is an abstract one; the reason is that we want to keep the framework general. In more concrete instantiations of this abstract framework, the notion of a situation will be specified based on the activity in which an agent acts and the requirements on the agent in this activity, as well as the representations and sensory-motor machinery of the agent. As a simple example, in the work of Steels and Belpaeme (2005) the situation is limited to a colour sample, perceived by a robot through a camera and processed into a representations of colours in the form of three real-valued numbers.

[13] It may be thought that appropriateness should be defined in terms of collective meaning potentials rather than individual usage pattens, to make sense of talk of "incorrect use of words." However, we believe that such talk is better regarded as one of many strategies for explicit negotiation of meanings, which always occurs in concrete situations and between individual DPs with their respective usage patterns. A theoretical notion of correct or incorrect use of words (independent of individual usage patterns) runs into several problems, such as defining how many DPs must share a usage pattern in order for it to be deemed "correct." This does not mean we cannot make sense of talk of incorrect and correct use of words; it only means that regard such notions primarily as devices in negotiations of shared meanings.

[14] In general, appropriateness is not necessarily a Boolean property, but rather a matter of degree. This is a simplification in the current theory.

[15] The exact method of deciding whether a new token is appropriate or not will depend on the specific kinds of representations, learning algorithms, and measures of similarity that are assumed (or, in an artificial agent, implemented).

tion s is consistent with $[c]^A$, A is likely to understand c and to judge s to be an appropriate situation of use of c. However, it is important to leave open the possibility that a DP may not understand, or understand but reject, a token of c even if this token of c in the current situation is appropriate with respect to A's usage pattern for c. Similarly, a DP may choose to use a word in a situation where she judges it inappropriate given previous uses; we call this a *creative use* (in contrast to conservative uses which are appropriate given previous uses).

5.4 Usage-pattern updates

It follows from the definition of $[c]^A$ that whenever A observes or performs a use of c, S_c^A will be extended, and so the usage pattern $[c]^A$ may change. This is a *usage pattern update*. *Prima facie*, there are many different possible kinds of ways that a usage pattern may be modified, depending on assumptions regarding semantic representation.

Usage-pattern updates can be distinguished according to several dimensions; we will start by making a rough distinction between *reinforcements* and *revisions*.

If a use of c in situation s is consistent with A's usage pattern for c, i.e., c is appropriate in s ($[c]^A \vdash s$), there is no drastic change; the previous disposition is reinforced by extending $[c]^A$ with A's situated interpretation of c in s, $[c]_s^A$. We will write this formally as $[c]^A \circ_= [c]_s^A$). However, if the current use of c is not consistent with usage disposition ($[c]^A \nvdash s$), there will be a relatively drastic revision of the disposition (formally, $[c]^A \circ_* [c]_s^A$).

5.5 Situation-types and structured meaning potentials

To account for how registers can be appropriated (borrowed) from one activity (e.g. telling the time) to another (e.g. direction-giving) we need a formalisation which allows new meanings of existing words to be created as a result of observed novel (at least subjectively) language use. Meaning potentials, which in addition to being dynamic can also be *structured*, and thus allow for different contexts to exploit different meaning potential *components*, seem useful.

We will use *situation-type* as a general term for contexts, activities, institutions etc. where words take on specific meanings. A register, or "microlanguage", is the lexicon used in a situation-type, pairing the words used (vocabulary) with meanings (what can be talked about; ontologies; coordinated usage patterns) in the situation-type[16]

In general, a situation-type may be associated with several registers (corresponding to different perspectives on the situation-type), each providing a mapping from a vocabulary to (abstract) meanings specific to the situation-type. Conversely, the meaning potential for a word is often structured into several situation-type-specific components.

We have established that $[c]^A$ is agent A's usage pattern for word c, and that $[c]_s^A$ is the interpretation that agent A makes of c in s; this interpretation is a function of s and $[c]^A$. We will now extend our notation with $[c]_\alpha^A$ - an agent A's situation-type-specific usage pattern for c in situation-type α. In general, any aspect of the utterance situation-type may activate usage pattern components. A structured meaning potential exists in a linguistic community with coordinated structured usage patterns. A component of structured meaning potential for c in situation-type α is written as $[c]_\alpha$[17].

As a simple example inspired by the Map Task dialogue above, the meaning potential ["two o'clock"] can be described as structured into

- ["two o'clock"]$_{clock}$, where *clock* stands for an activity type involving telling the time; this meaning potential component can be paraphrased "02:00 AM or PM"
- ["two o'clock"]$_{direction-giving}$, where α has been assigned a situation type index corresponding to direction-giving activities; this meaning potential component is paraphraseable as "east-northeast direction"

5.6 Interpretation and update involving structured usage patterns

A token c_s of a word c in situation s is interpreted by B as $[c]_s^B$. If $[c]^B$ is a complex usage pattern, some component of $[c]^B$ must be selected as the abstract meaning to be used for contextual interpretation. Now, assume that situation s is classified by B as being of situation-type α. This triggers a component of $[c]^B$ - the *activated usage pattern component* $[c]_\alpha^B$.

[16] This terminology builds on (and modifies slightly) that of Halliday (1978).

[17] An obvious extension to this formalism, which we will not develop further here, would be to index meaning potentials (and their components) by the linguistic community in which they exist.

In this case, $[c]^B_\alpha$ is a likely candidate for which part of $[c]$ gets updated. (If B is not able to find a relevant usage pattern component, B may create a new ad-hoc component, which can be updated during the dialogue. This pattern may or may not be retained afterwards; it may be assimilated into some existing component of $[c]$, or the start of a new usage pattern component.)

Let’s take an example. Assume [“two o’clock”] is structured into [“two o’clock”]$_{clock}$ and [“two o’clock”]$_{direction-giving}$, as above. Now assume we get the following utterance:

GIVER: ”sort of two o’clock”

Because the activity is direction-giving, FOLLOWER activates [”two o’clock”]$^{follower}_{direction-giving}$. FOLLOWER then instantiates [”two o’clock”]$^{follower}_{direction-giving}$ to arrive at a contextual interpretation [”two o’clock”]$^{follower}_{s}$ (roughly, a 60 degree angle on FOLLOWER’s map). Insofar as [”two o’clock”]$^{follower}_{direction-giving}$ $\vdash$ s, we get a reinforcing update [”two o’clock”]$^{follower}_{direction-giving}$ $\circ_{=}$ [”two o’clock”]$^{follower}_{s}$.

6 Semantic coordination

This section sketches a framework for modelling *negotiation of meaning in dialogue*, i.e. the social processes (dialogue games) involved in the explicit and implicit negotiation of meaning in dialogue, and their relation to the cognitive processes (semantic updates).

After discussing the basic devices available to speakers for conducting semantic negotiation, we will give examples of how the theory sketched above can be used to analyse short dialogue excerpts in terms of semantic updates. As yet, the theory does not include a taxonomy of dialogue moves involved in semantic negotiation, and therefore the analysis does not include dialogue moves; instead, utterances are analysed directly in terms of their associated semantic updates. Coming up with a general taxonomy of such moves and their associated updates is a major future research goal.

6.1 Basic devices for coordination in dialogue

We assume (provisionally) three basic devices available to dialogue participants for negotiating (and, typically, achieving coordination of) linguistic resources: feedback, explicit negotiation, and accommodation. “Negotiation” is used here in a weak sense of “interactive achievement of coordination”.

Feedback (Allwood, 1995; Clark, 1996) involves signals indicating perception, understanding, and acceptance of utterances in dialogue, as well as failure to perceive or understand; clarification requests; and rejections. It is well known that feedback governs that coordination of the dialogue gameboard (“informational coordination”); however, it also guides coordination of language use (“language coordination”).

For example, *corrective* feedback is common in adult-child interaction. Below is an example; A is the child, B the adult, and as part of the common ground there is a topical object in the situation s visible to both A and B. We also assume that A is not familiar with the word “panda”.[18]

A: Nice bear

B: Yes, it’s a nice panda

Here, B rejects this use of “bear” by providing negative feedback in the form of a correction (and in addition, B gives positive feedback accepting the assertion that the focused object (animal) “is nice”). For an account of this example in terms of semantic plasticity and coordination, see Larsson (2007).

Explicit negotiation is the overt meta-linguistic negotiation of the proper usage of words, including e.g. cases where explicit verbal or ostensive definitions are proposed (and possibly discussed). Although semantic negotiation typically has the goal of coordinating language use, it may in general be both antagonistic and cooperative. In Steels and Belpaeme (2005), robot agents play a language game of referring to and pointing to colour samples. The colour-language system of an individual agent is modelled as a set of categories in the form of neural nets that respond to sensory data from colour samples, and a lexicon connecting words to categories. This is clearly a case of explicit semantic plasticity and semantic negotiation, as categories are updated as a result of language use. Semantic negotiation here takes the form of explicit and cooperative negotiation. For an account of a dialogue taken from the this exper-

[18] This example from Herb Clark, p.c.; similar examples can be found in Clark (2003)

iment in terms of semantic plasticity, see Larsson (2007).

By *accommodation* we refer to adaptations to the behaviour of other DPs. For example, one may adapt to the presuppositions of an utterance of “The King of France is bald” by modifying the dialogue gameboard to include the existence of a king of France. We want to extend the notion of accommodation beyond the dialogue gameboard, to include changes in the language system.

For each word used in an utterance u, the addressee (here, B) in a dialogue is (usually) expected to react if he thinks a word in u was inappropriately used. If B is able to construct a situated interpretation $[c]^B_s$ (which may involve more or less effort) but finds this use inappropriate ($[c]^B \nvdash s$), this may be due to a mismatch between s (as perceived by B) and $[c]^B$. B may now reject this use of c explicitly using negative feedback, or quietly alter $[c]^B$ ($[c]_B \circ_* [c]^B_s$) so that this use of c can be counted as appropriate after all.

6.2 Coordination through accommodation

We will now give an example of semantic coordination in dialogue, where meaning accommodation leads to updates to complex usage patterns.

Assume we get the following utterance in a Map Task dialogue in a situation s:

GIVER: ”sort of two o’clock”

Assume[19] also that FOLLOWER is not familiar with the “direction-giving” use of “two o’clock”. More precisely, [“two o’clock”]^{fol} only contains [“two o’clock”]$_{clock}$, so [“two o’clock”]$^{fol} \nvdash s$.

By analogical reasoning using contextual features, FOLLOWER is nevertheless able to correctly understand A’s utterance and arrives at a contextual interpretation [“two o’clock”]$^{fol}_s$. Now, since [“two o’clock”]$^{fol} \nvdash s$, FOLLOWER needs to revise [“two o’clock”]^{fol} by creating a new activity-specific component [“two o’clock”]$^{fol}_{d-g}$. We get an overall update [“two o’clock”]^{fol} $\circ_=$ [“two o’clock”]$^{fol}_s$ which can be decomposed as two updates, (1) creation of [“two o’clock”]$^{fol}_{d-gg}$, followed by [“two o’clock”]$^{fol}_{d-g}$ $\circ_=$ [“two o’clock”]$^{fol}_s$. After this update, [“two o’clock”]$^{fol} \vdash s$, i.e., the novel (for FOLLOWER) use of “two o’clock” by GIVER has been accommodated.

7 Kinds of coordination in dialogue

On our view, two kinds of coordination happen in everyday human-human dialogue. *Informational coordination* has successfully been studied using the concepts of dialogue games and updates to a shared dialogue gameboard. One of the goals of the research presented here is to extend this approach to describing *language coordination* (and more specifically, semantic coordination) in terms of the dynamics of updates to language systems.

The framework sketched here aims at describing all kinds of semantic coordination[20]. In the “two o’clock” example given above, coordination is essentially a matter of mapping an expression (“two o’clock”) to a pre-existing meaning (denoted in the compass directions register as “east-northeast”). For this kind of coordination, some version of traditional formal semantics may suffice, provided it is extended with a dynamic mapping between linguistic expressions and their meanings[21].

However, in other cases the dynamics go beyond word-meaning mappings. Specifically, to account for cases where an expression is used to denote a *new* concept, such as “the u-shape” above, we need to describe the dynamics of *concept creation*. Similarly, existing concepts may be affected by their use in dialogue, e.g., by subtly modifying values of usage-governing conceptual features by small increments. For example, in Steels and Belpaeme (2005), concepts are represented as neural nets which are updated by small adjustments to network weights, according to a standard backpropagation algorithm.

These dynamics, which we refer to as *concept-level* dynamics, are an important motivation for the introduction of meaning potentials. They are also our main reason for believing that traditional formal semantics will not suffice to account for semantic plasticity coordination.

To deal with concept-level dynamics in a general way, one will probably need to keep track of of semantic features connected to expressions in

[19] In this example, we will use the following abbreviations: fol = follower, d-g = direction-giving.

[20] A typology of variants of semantic coordination is a future research goal.

[21] Note that “dynamic semantics” (Groenendijk and Stokhof, 1988) is not dynamic in this sense, as it follows traditional formal semantics in assuming a static mapping between words and meanings.

the lexicon (Pustejovsky, 1991) and allow these feature matrices to be updated as a result of semantic negotiation and coordination subdialogues. Work in this direction may benefit from ideas put forward by Gärdenfors (2000), as well as in work on machine learning (Mitchell, 1997) and Latent Semantic Analysis (Landauer and Dumais, 1997). One version of formal semantics which seems promising for the illumination of concept-level dynamics is the record-type theoretic approach which Cooper has been developing (Cooper, 2005a; Cooper, 2005b). This formal approach allows for both underspecification or uncertainty of meaning by the use of types of meaning and also a structured approach to meaning analysis which allows for modification of meaning in a way which is not possible, for example, in the classical formal semantics analysis of meaning as functions from contexts to intensions.

8 Conclusion

To account for the observed dynamics of semantic systems in dialogue, we have sketched a formalisation of the notion of meaning potential, in the form of dynamic structured usage patterns which are shared within a linguistic community through a process of semantic coordination in dialogue. This process can be described as updates to structured usage patterns resulting from language use. We have also outlined some basic mechanisms of coordination: feedback, explicit negotiation, and accommodation.

This paper presents preliminary work aiming towards a unified theoretical account of semantic coordination. Apart from developing the theory and the formal framework further, we want to extend the coverage of this theory by further empirical studies, and to start implementing strategies for semantic coordination in practical dialogue systems.

References

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

S. E. Brennan. To appear. The vocabulary problem in spoken language systems. In S. Luperfoy, editor, *Automated spoken dialog systems*. Cambridge, MA: MIT Press.

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

E. V. Clark. 2003. *First language acquisition*. Cambridge: Cambridge University Press.

Robin Cooper. 2005a. Austinian truth, attitudes and type theory. *Research on Language and Computation*, 3(4):333–362, December.

Robin Cooper. 2005b. Records and record types in semantic theory. *J. Log. and Comput.*, 15(2):99–112.

Peter Gärdenfors. 2000. *Conceptual Spaces: The Geometry of Thought*. MIT Press, Cambridge, MA, USA.

Simon C. Garrod and Anthony Anderson. 1987. Saying what you mean in dialogue: a study in conceptual and semantic co-ordination. *Cognition*, 27:181–218.

J. A. G. Groenendijk and M. J. B. Stokhof. 1988. Context and information in dynamic semantics. In *Working models of human perception*. Academic Press.

M.A.K Halliday. 1978. *Language as Social Semiotic: The Social Interpretation of Language and Meaning*. Baltimore: University Park Press.

P.G.T. Healey. 1997. Expertise or expertese?: The emergence of task-oriented sub-languages. In M.G. Shafto and P. Langley, editors, *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 301–306.

D. Kaplan. 1979. Dthat. In P. Cole, editor, *Syntax and Semantics v. 9, Pragmatics*, pages 221–243. Academic Press, New York.

Thomas K Landauer and Susan T. Dumais. 1997. A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

Staffan Larsson. 2007. A general framework for semantic plasticity and negotiation. In H. C. Bunt, editor, *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS-7).*

Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill, New York.

J. Pustejovsky. 1991. The generative lexicon. *Computational Linguistics*, 17(4):409–441.

Francois Recanati. 2003. *Literal Meaning - The Very Idea*. Cambridge University Press.

Luc Steels and Tony Belpaeme. 2005. Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, 28(4):469–89, August. Target Paper, discussion 489-529.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 117–124.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Dialogue Games for Crosslingual Communication

Paul Piwek
NLG group
Computing Department
The Open University
Walton Hall
Milton Keynes, UK
P.Piwek@open.ac.uk

David Hardcastle
NLG group
Computing Department
The Open University
Walton Hall
Milton Keynes, UK
D.W.Hardcastle@open.ac.uk

Richard Power
NLG group
Computing Department
The Open University
Walton Hall
Milton Keynes, UK
R.Power@open.ac.uk

Abstract

We describe a novel approach to *crosslingual dialogue* that supports *highly accurate* communication of *semantically complex* content between people who do not speak the same language. The approach is introduced through an implemented application that covers the same ground as the chapter of a conventional phrase book for food shopping. We position the approach with respect to dialogue systems and Machine Translation-based approaches to crosslingual dialogue. The current work is offered as a first step towards the innovative use of dialogue theories for the enhancement of human–human dialogue.

1 Introduction

Original Dutch text: Daar achter staat een doos met appels. Kan ik daar een een halve kilo van hebben?

Translation into English by human: *Back there, there is a box with apples. Can I have half a kilo of those?*

Translation into English by Altavista Babelfish (April 17, 2007): *There behind state a box with apples. Am I possible of it a half kilo have?*

The example above illustrates some of the shortcomings of Machine Translation (MT). Apart from many other errors in the translation, note that Babel Fish incorrectly uses singular 'it' to refer to the plural 'apples'. Babel Fish does not model how sentences both change the context and depend on it for their interpretation; consequently 'apples' does not lead to the introduction of a representation for a (plural) set of apples that can subsequently be referred to. This is a symptom of a more general issue: Much of MT is still grounded in the classical transmission model in which a speaker communicates a message *m* by encoding *m* in a natural language sentence and the hearer subsequently decodes it. MT typically maps sentences from source to target *one at a time*, treating each sentence as separate problem. In this paper, we will put forward an approach to crosslingual dialogue that fits better with contemporary semantic theory, in which meanings of natural language expressions are conceived of as 'programs' that change information states, rather than static representations (of the world or what is in the mind of the speaker).

From a practical point of view, it is worthwhile to compare MT-based crosslingual dialogue systems with spoken dialogue systems. Even for relatively simple domains, such as travel planning, large and extremely large-scale research projects such as the Spoken Language Translator (Rayner et al., 2000) and Verbmobil[1] have, despite making substantial contributions to various areas of speech and language processing, not yet delivered systems for practical deployment. In contrast, spoken dialogue systems are nowadays deployed in many countries for tasks ranging from providing travel information to call routing. The apparent intractability of human–human crosslingual dialogue, as opposed to human–machine dialogue, is partly a result of the fact that whereas in the latter it is straightforward to influence the human dialogue participant's contributions, through system

[1] See http://verbmobil.dfki.de/

initiative, it is less obvious how to do so in human–human dialogue. When a system tracks human–human dialogue, it cannot influence the utterances of the human interlocutors (e.g., by asking questions such as 'From where to where would you like to travel?').

In short, both in theoretical and practical terms, the current state-of-the-art of tools for supporting crosslingual human–human dialogue lags behind other areas of dialogue research. The current work is an attempt to close the gap. We will present an approach to crosslingual dialogue that allows both for better transfer of knowledge from contemporary theories of semantics and dialogue to crosslingual dialogue technology, and has potential for practical applications.

The basic idea is to take the conception of dialogue as a game in which contributors take turns that result in updates on their information states (Traum and Larsson, 2003) quite literally. Although we aim to leverage insights regarding the foundations of human–human dialogue, we will not direcly mimick it in all its details. The aim is to exploit contemporary insights (from speech act theory, theories of common ground in dialogue, conversational sequencing, etc.) to build computational artifacts that *enhance* human–human dialogue.

In the next section, we introduce the underlying technology, Conceptual Authoring, and describe how it can be adapted to facilitate crosslingual dialogue. Details of the underlying system architecture are described in Section 3. Section 4 summarizes the benefits of the proposed approach and compares it with Machine Translation-based approaches. Finally, in Section 5 we provide a number of research goals that we intend to pursue in future, using the current work as a starting point.

2 From Conceptual Authoring to Crosslingual Dialogue

Conceptual Authoring (CA) was pioneered by Power and Scott (1998). A number of CA-applications were developed at the University of Brighton and, subsequently, the Open University[2]. At Harvard, Nickerson (2005) investigated CA for reference specification, and Xerox Research Centre Europe has explored a similar approach, which they call Multilingual Document Authoring (Dymetman et al., 2000).

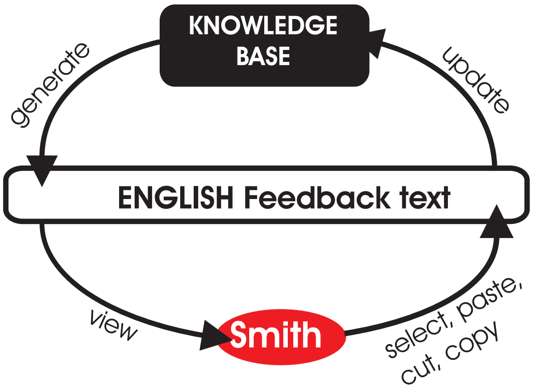


Figure 1: Conceptual Authoring (CA) editing cycle

The key principle underpinning CA is presented by the editing cycle in Figure 1: Given a Knowledge Base (KB), the system generates a description of the KB in the form of a feedback text containing anchors (coloured spans of text) representing places where the content in the KB can be extended. In Figure 1, the user is Mr. Smith and he interacts with an English feedback text. Each anchor is associated with pop-up menus, which present the possible extensions of the KB at that point. These are computed by consulting an ontology that underlies the KB. More precisely, the KB consists of two components:

1. an ontology, also known as the terminological box (*T-box*) which specifies the set of available concepts and their attributes, and

2. an assertion box (*A-box*) in which instances of concepts/classes are introduced. It is the A-box that is updated, and the T-box which specifies the set of possible updates.

On the basis of the user's selection, the KB is updated and a new feedback text (reflecting the updated content) is generated. Additionally, spans of feedback text representing an object in the KB can be selected using the mouse to move or remove the object to or from a location in the KB. After each action, a new feedback text is generated representing the updated KB.

The potential of this approach is evidenced by its successful deployment in query formulation (Piwek et al., 2000; Evans et al., 2006; Hallett et al., 2007). For example, Hallett et al. (2007) showed that the method enables untrained users

[2] See http://mcs.open.ac.uk/nlg/research/Conceptual_Authoring.html

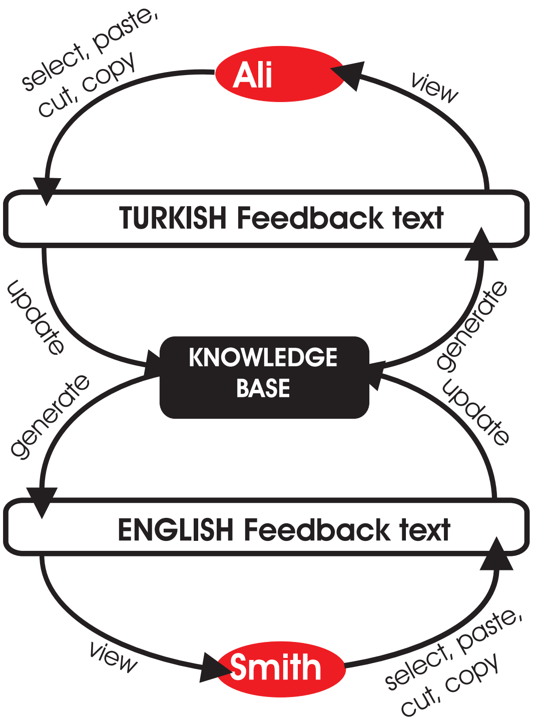


Figure 2: CROCODIAL Multi-person Conceptual Authoring (CA) editing cycle

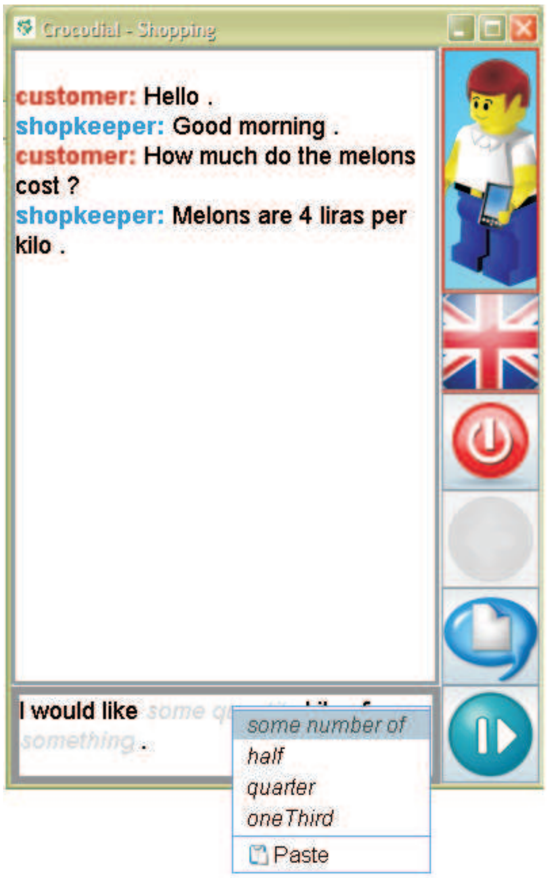
to successfully and reliably formulate complex queries, while avoiding the standard pitfalls of free text queries.

Here, we discuss an extension – CROCODIAL (for Crosslingual Computer-mediated Dialogue) – of CA to dialogue that was first proposed in Piwek & Power (2006). The extension rests on the idea of taking CA from single-person authoring to multi-person authoring. This is visualized in Figure 2. Here, we have a second editor (Ms. Ali) with access to the same underlying KB as Mr. Smith. Crosslingual dialogue is made possible because although each editor has access to the same KB, their views of it are different: Ali looks at it through 'Turkish glasses' (a language generator for Turkish) and Smith through English ones. Of course such a multi-person editing does not necessarily lead to interactions that qualify as dialogues. To approximate dialogue behaviour we introduce some constraints:

1. The jointly edited structure has to be interpreted as representing the *dialogue history*, progressively built up.

2. Only the most recent turn in the history can be modified, although material can be *copied from preceding turns to establish anaphoric links*.

3. Interlocutors construct turns one at a time.



customer: Hello .
shopkeeper: Good morning .
customer: How much do the melons cost ?
shopkeeper: Melons are 4 liras per kilo .

I would like some q... something .

Figure 3: Screen capture of Conceptual Authoring (CA) Interface for English-speaking Customer. Construction of contribution is in progress in feedback pane.

Figures 3, 4 and 5 are screen captures of our implemented CROCODIAL prototype system. The system supports for conversations between between a shopkeeper and a customer. In our examples, we have a Turkish-speaking shopkeeper and an English-speaking customer.

CROCODIAL allows both for use of the system similar to chatroom internet applications, and on a single portable device (see section 3). For this particular scenario, the scenario is running on a single portable device (e.g., PDA or Tablet PC). The interface consists of three panes:

1. a history pane (top left) that shows a record of the conversation so far,

2. a feedback editing pane (bottom left) where

the current ‘speaker’ can edit his or her turn, and

3. a pane (right-hand side) with several icons and buttons running from the top to the bottom of the pane:

   (a) an icon representing the current role of the speaker (either shopkeeper or customer),

   (b) an icon representing the language of the current speaker (the icon is clickable and allows the current user to change their language, i.e., the language in which the KB is depicted in the history and feedback panes),

   (c) a button to exit from the application,

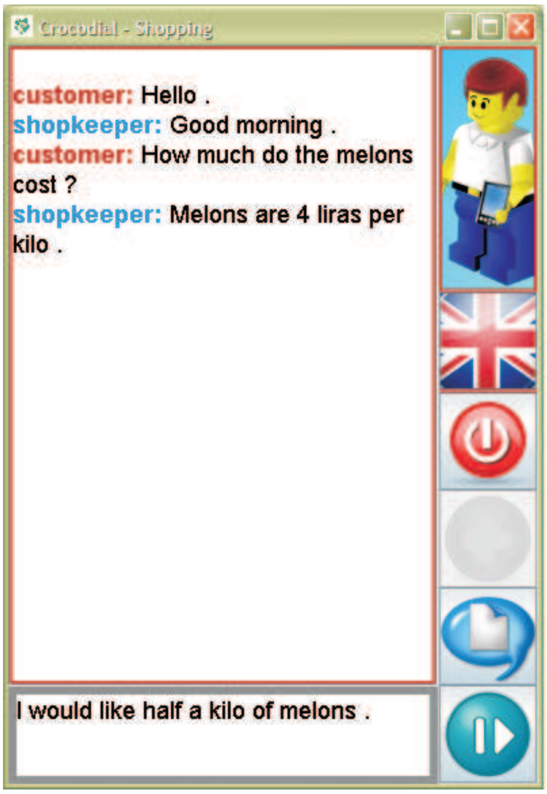
   (d) an ‘undo button’,

   (e) a button that allows the current speaker to add further speech acts/messages to their turn, and

   (f) a button that allows the current speaker to release the turn to the other speaker. When this button is pressed, a number things happen: Firstly, in the KB the representation underlying the feedback text is added to the history. Secondly, fresh underlying representation is created for the feedback text that allows formulation of a new turn. Thirdly, the language of the history and feedback panes are changed to that of the next ‘speaker’. Finally, the righhand side pane is changed accordingly, i.e., the icon of the current role is changed, and the icon for the current language is changed also to that of the next speaker.

In Figure 3, it is the English-speaking customer’s turn. The history pane shows the preceding conversation. In the feedback pane, the state of the turn that is under construction is represented by the text ‘I would like *some quantity* of *something*’. The anchors are in grey italicized text. They indicate options for extending the current turn. The user has selected the anchor ‘*some quantity*’ and is presented on a menu with several options (‘half’, ‘quarter’ and ‘one third’).

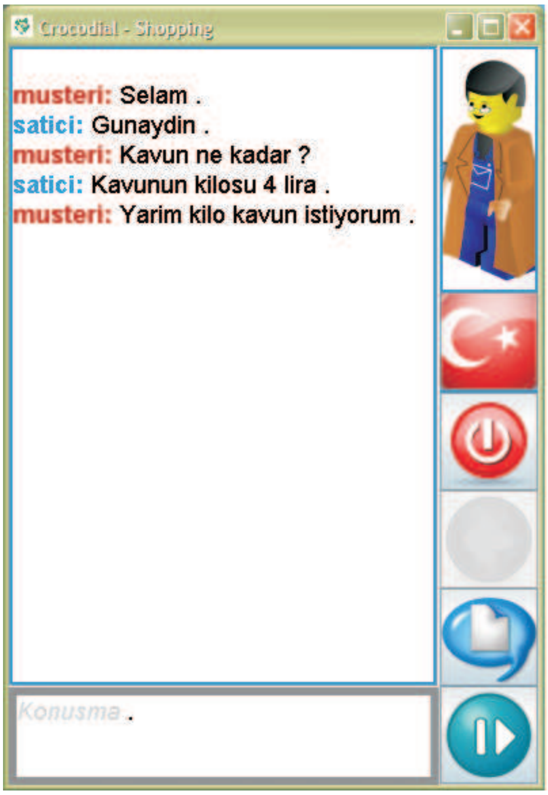
The next figure (Figure 4) shows the state of the feedback text after the user has made selections for both anchors, resulting in the text ‘I would like half a kilo of melons’.



customer: Hello .
shopkeeper: Good morning .
customer: How much do the melons cost ?
shopkeeper: Melons are 4 liras per kilo .

I would like half a kilo of melons .

Figure 4: Screen capture of Conceptual Authoring (CA) Interface for English-speaking Customer. Contribution has been completed.



musteri: Selam .
satici: Gunaydin .
musteri: Kavun ne kadar ?
satici: Kavunun kilosu 4 lira .
musteri: Yarim kilo kavun istiyorum .

Figure 5: Screen capture of Conceptual Authoring (CA) Interface for Turkish-speaking shopkeeper. State of the Interface after the customer has released the turn to the shopkeeper.

The result of the current user yielding the turn is depicted in Figure 5. Now, it is the Turkish shopkeeper's turn. The language of the history and feedback panes and the right-hand pane icons have been changed accordingly. The feedback pane provides the starting point for constructing a new turn; the anchor '*Konusma*' is equivalent to the English anchor '*Some speech act*'.

The current prototype implements one chapter (food shopping) from a traditional English-Turkish phrase book. Further work will include extending this to further chapters (such as traveling and restaurants/bars). Each chapter is viewed as a self-contained 'dialogue game' that allows the users to construct certain locutions that are appropriate in the corresponding setting.

It took the first author approximately three days to develop the ontology and English resources for the food shopping dialogue game. It took another two days to add the language sources for Turkish. This involved consulting a native speaker of Turkish.

3 System Architecture and Implementation

CROCODIAL is implemented as a chat room: users log in and are authenticated against a server, and each user can see who else is logged in and initiate a dialogue with them. The difference is that the users must agree a dialogue game (such as a shopping encounter) and decide the role within that dialogue game that each is to play (for example the customer and the shop assistant).

The chat window that each user sees is similar in layout to most chat interfaces. It contains a history of the conversation with each entry labelled and colour-coded to identify the speaker, some navigation controls and an input pane to receive text input. This input pane is a CA feedback text interface, allowing the user to interact with the underlying Knowledge Base to develop each utterance that they wish to contribute to the conversation. In the current implementation, the CA application which deals with operations on the KB and generation of feedback texts is implemented in Prolog, running as a shared process on the server.

The chat application is implemented in Java and sits on top of a newly developed framework that makes it easy to develop user interfaces to our CA applications. The architecture – see Figure 6 – is broadly MVC (Model-View-Controller) with the task of updating the model delegated by the controller to the Prolog CA core system. Since the interlocutors are both extending the same underlying KB, the Prolog system is single-threaded, with each new utterance extending the same A-Box. To turn this single-threaded application into a CA chat room the View component is replaced with a multi-threaded session object that allows each chat window to send commands to Prolog and receive updates to its current model as appropriate. To ensure that users do not simultaneously extend the A-Box in mutually inconsistent ways the users are forced to take turns.

Bandwidth requirements are kept down by transmitting only the most recent turn as the model - this means that the history of the conversation shown to each user must be stitched back together by the Java session from the sequence of partial models returned by Prolog.

At any point in the dialogue each user can switch to a different language, choosing from any of the languages supported by the system. Because the text in the chat window is a conceptually

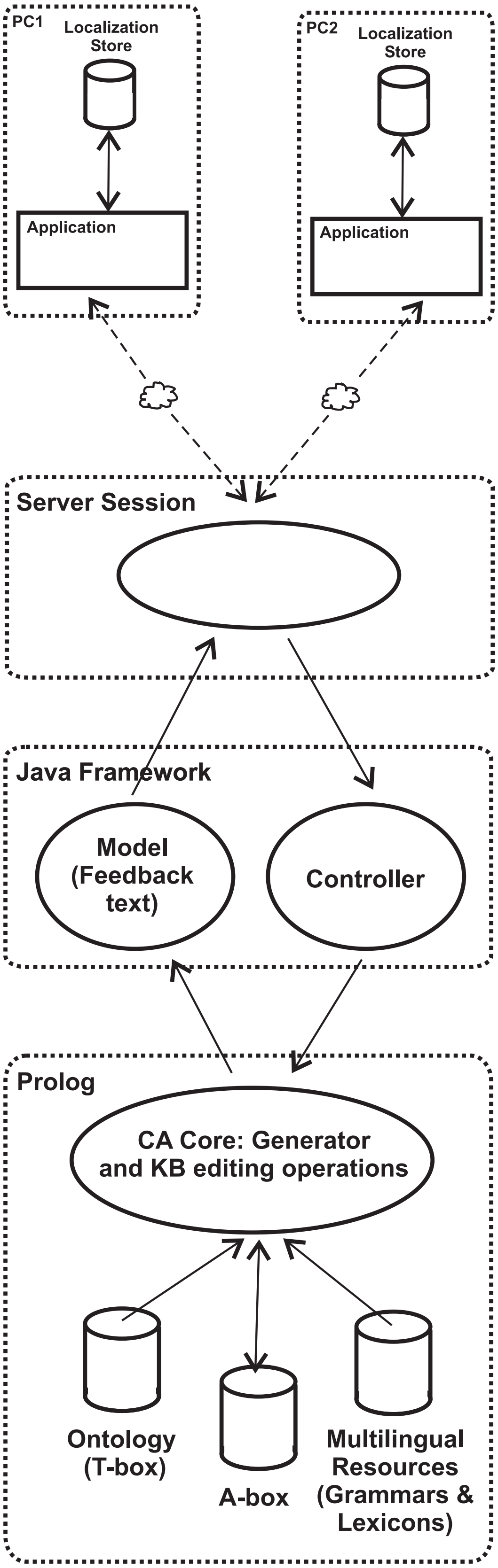


Figure 6: CROCODIAL architecture diagram

authored feedback text this action is functionally equivalent to any other interaction with the text: a command is sent from the window to the session on the server, it delegates the request to a Prolog executable and receives a new CA model which it then returns to the chat window which made the request.

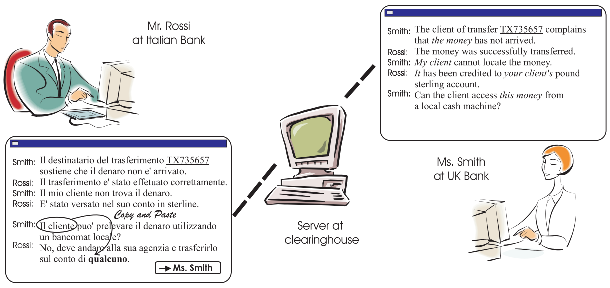
The server itself is multi-sessional, and so a single server can support many simultaneous dialogues, and a single user logged onto the server can participate in many different dialogues at the same time. Each of these dialogues is conceptually logged: by which we mean that each state change made to the A-Box is recorded in a log file. This allows a record of the conversation to be regenerated in any language supported by the system, even if the language was not used in the original dialogue, and also allows us to analyse the use of the system, for example by analysing the time taken to formulate particular questions or responses, or analysing how often speakers back track and correct their utterances.

We have also implemented CROCODIAL as a single-user standalone system which launches a single chat window that switches role and language each time the user completes an utterance. We are planning to migrate the Prolog generation system to Java and produce a single-user Java-only version for use in mobile computing contexts, such as on a PDA in an actual shop in a foreign country.

4 Discussion

In our introduction, we attributed the success of dialogue systems for tasks such as travel planning partly to the fact that the tasks that are involved allow the initiative to reside with the system. The system determines what the main topic will be of the user's next turn, and can thus build up fairly reliable expectations concerning what the user is about to say next. The difficulties facing Machine Translation-based crosslingual dialogue systems for facilitating human–human dialogue can be traced back to the absence of opportunities for such system initiative: each interlocutor is free to say whatever they want at any given time in the conversation. The system has no reference point such as its own utterances, with respect to which it can 'anchor' the users's utterances.

The CROCODIAL system can be viewed as addressing this problem. The multi-person CA technology forces each interlocutor to construct their



Smith: Il destinatario del trasferimento TX735657 sostiene che il denaro non e' arrivato.
Rossi: Il trasferimento e' stato effetuato correttamente.
Smith: Il mio cliente non trova il denaro.
Rossi: E' stato versato nel suo conto in sterline.
Smith: Il cliente puo' prelevare il denaro utilizzando un bancomat locale?
Rossi: No, deve andare alla sua agenzia e trasferirlo sul conto di **qualcuno**.

Smith: The client of transfer TX735657 complains that *the money* has not arrived.
Rossi: The money was successfully transferred.
Smith: *My client* cannot locate the money.
Rossi: *It* has been credited to *your client's* pound sterling account.
Smith: Can the client access *this money* from a local cash machine?

Figure 7: Mock-up of CROCODIAL application in the banking domain from Piwek & Power (2006).

dialogue contributions using concepts that are available to the system. The users are, however, not directly confronted with the system's concepts/ontology; rather this is mediated to them via feedback texts that are automatically generated.

The fact that this approach constrains the interlocutors in what they can say (just like in spoken dialogue systems, the interlocutors are constrained in what they can say through system initiative), has to be weighed up against a number of clear benefits. Firstly, as opposed to Machine Translation-based crosslingual dialogue, CROCODIAL supports highly accurate exchange of information, since no interpretation of the user's utterances is required; users manipulate the underlying content of messages directly. Secondly, the CA technology allows for formulation of utterances with complex semantic content that is even beyond the capabilities of most current state-of-the-art dialogue systems, including (plural) co-reference (Piwek, 2000) and logical connectives (Power, 1999). The technology underlying CROCODIAL, drawing on insights from dynamic semantics, fits in better with contemporary semantic theory than Machine Translation-based approaches. Finally, in addition to the accuracy and coverage of complexity supported by our approach, it also allows us to benefit from the fact that the interlocutors construct a formal representation of the content of the interaction.

In Piwek & Power (2006) we discuss the use of CROCODIAL for exchanges between employees of international banks regarding financial transactions.

Figure 7 shows a dialogue in the banking domain that is discussed Piwek and Power (2006). We describe how the formal representation of the content underlying the dialogue can be exploited for automatic summarization of the dialogue. The interaction in Figure 7 could lead a summarizer to produce the following summary which integrates contextual information regarding the transaction (date, banks involved, etc.).

> On 15-1-2003 Ms Smith (Citibank) called Mr Rossi (Banca di Roma) about the transfer of 100.000 GBP to the account of Count Roberto da Silva (654012). It was established that the money had been transferred to the pound sterling account of Da Silva. This account can only be accessed via a local branch of the Banca di Roma.

Similar summaries could be generated on demand in other languages when the need for this arises; the basis for such summaries is the formal representation of the dialogue which the interlocutors unwittingly construct.

5 Conclusions and further work

We have described a prototype for supporting human–human crosslingual dialogue. Apart from the practical benefits of this system (allowing accurate transfer of complex semantic content, with a formal record of the dialogue as a by-product) we would like to argue that this prototype also provides us with a workbench for deploying contemporary theories of dialogue and gaining a better understanding of these theories.

In the current prototype we chose to use the shared KB to represent the dialogue history. There are, however, alternatives. We could, for example use the KB to represent the commitments (Hamblin, 1971; Walton and Krabbe, 1995) of the interlocutors, and investigate what kind of feedback texts this would require. This would also allow us to empirically compare prototypes based on dialogue history versus commitment store KB's.

Finally, in the current prototype at each stage in the dialogue, options for constructing a turn are presented unfiltered and in alphabetical order. Theories of dialogue provide us, however, with many rules that constrain/predict the content of turn given the preceding turns. We would like investigate whether using such information to filter and re-order the CA editing option, allows for quicker/more efficient dialogues.

Acknowledgements

We would like to thank Gökçen Piwek for her help with building the language resources for the Turkish feedback texts.

References

M. Dymetman, V. Lux, and A. Ranta. 2000. XML and Multilingual Document Authoring: Convergent Trends. In *Procs of COLING 2000*, pages 243–249, Saarbruecken, Germany.

R. Evans, P. Piwek, L. Cahill, and N. Tipper. 2006. Natural language processing in CLIME: a multilingual legal advisory system. *Journal of Natural Language Engineering*, June.

C. Hallett, D. Scott, and R. Power. 2007. Composing Questions through Conceptual Authoring. *Computational Linguistics*, 33(1):105–133.

C.L. Hamblin. 1971. Mathematical Models of Dialogue. *Theoria*, 37:130–155.

J. Nickerson. 2005. *Reference Specification in Multilingual Document Production*. Ph.D. thesis, Department of Computer Science, Harvard University.

P. Piwek and R. Power. 2006. CROCODIAL: Crosslingual Computer-mediated Dialogue. In *Procs of the 3rd International Workshop on Computer Supported Activity Coordination (CSAC 2006)*, Paphos, Cyprus, May.

P. Piwek, R. Evans, L. Cahill, and R. Evans. 2000. Natural Language Generation in the MILE system. In *Procs of IMPACTS in NLG workshop*, Schloss Dagstuhl, Germany, July.

P. Piwek. 2000. A Formal Semantics for Editing and Generating Plurals. In *Procs of COLING 2000*, pages 607–613, Saarbruecken, Germany.

R. Power and D. Scott. 1998. Multilingual authoring using feedback texts. In *Procs of COLING-ACL 98*, Montreal, Canada.

R. Power. 1999. Controlling logical scope in text generation. In *Procs of the European Workshop on Natural Language Generation*, Toulouse, France.

M. Rayner, D. Carter, P. Bouillon, V. Digilakis, and M. Wiren. 2000. *The Spoken Language Translator*. Cambridge University Press, Cambridge.

D. Traum and S. Larsson. 2003. The Information State Approach to Dialogue Management. In *Current and New Directions in Discourse and Dialogue*, pages 325–353. Kluwer Academic Publishers.

D. Walton and E. Krabbe. 1995. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, New York.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 125–131.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Between “cost” and “default”: a new approach to Scalar Implicature

Francesca Foppolo
University of Milano-Bicocca
francesca.foppolo@unimib.it

Abstract

Scalar Implicatures are pragmatic inferences that are normally derived in conversational exchanges when a scalar term, such as for example “or”, is used. Different theoretical accounts have been proposed to describe how and at which point in the derivation we actually add this inference. Large part of the most recent debate is focused on the question of the “cost” of implicature computation, an aspect that is crucial to choose among alternative accounts. In this perspective, my intent here is to present an experimental study in the ongoing debate centred on the “costly” or “default” nature of implicature computation. The main result of the study presented here is the fact that a “cost” is found only when the implicature is added despite the fact that it leads to a weakening of the overall assertion (namely, in DE contexts): this loss in informativity, and not implicature computation *per se*, is interpreted as the source of this “cost”. The theoretical background for this study is offered by Chierchia (2006) and his new intriguing parallelism between the phenomenon of scalar implicature and negative polarity.

1 The phenomenon

Scalar Implicatures (SIs henceforth) are pragmatic inferences that are normally derived in conversational exchange when a scalar term, such as “or” is used. Consider the example in (1) and (2):

(1) The dwarf is singing *or* dancing
(2) The dwarf is singing *and* dancing

What is normally conveyed by uttering (1) is that (2) doesn’t hold. This amounts to saying that, by uttering (1), the inference that the hearer is allowed to draw is (3), which is actually how a sentence like (1) is normally understood:

(3) The dwarf is singing *or* dancing <u>but not both</u>

The mechanism by which SIs are derived is based on the notion of scale, on the one hand, and on that of informational strength on the other. In our example above, “or” belongs to an informational scale, i.e. <or, and>, in which “and” is the strongest element. By virtue of the fact that (2) constitutes the strongest alternative to (1) (it contains the stronger element “and”), and that (2) is not what was actually reported, then one is entitled to assume that (2) does not hold, hence the inference in (3) in which the negation of the strongest element on the scale is added.

2 The ongoing debate

Different theoretical accounts have been proposed to explain how and when implicatures are derived. We will focus here on one aspect of this debate in particular, namely the question of the “cost” of implicature computation. This has been the centre of the most recent debate between supporters of Relevance Theory (cf., a.o., Sperber and Wilson, 1986) on the one hand and of Default approaches on the other (cf., a.o., Levinson, 2000). The claim that implicatures are added at a cost by our processing system is necessary to differentiate these two approaches. In Levinsonian terms, implicature computation constitutes a default process, i.e. something that our computational/processing system performs automatically, thus it is by definition virtually costless. On the Relevance Theoretical view, instead, every operation imposed to our processing system must be evaluated in terms of “costs and benefits”, ultimately in terms of “relevance” to contextual assumptions: only those stimuli that are relevant enough are worth a processing effort. From this assumption, the claim that implicatures are costly necessarily follows: implicatures are only derived when explicitly required by the context, i.e. when the benefits that one gains from their computation overcome the processing effort required to derive them. If implicatures were costless, then the principle of optimal relevance would lose its foundation. This is the reason why all the experimental works on scalar implicatures within the Relevance Theoretic tradition have been focused on finding evidence of such a "cost".

Between these two approaches, there is a third proposal, recently delineated by Chierchia (2006). This approach seems to combine some features of the two approaches and, in my view, gives a new direction for solving the question of how and when and why scalar implicatures are derived. I will sketch this new proposal in the following section.

3 Chierchia's proposal

In Chierchia's most recent work (cf. Chierchia, 2006 in particular but also Chierchia, 2002), a unified account of negative polarity elements like *any* and scalar implicatures is being considered. In this new formulation, a binary feature σ is introduced as regulating the activation of scalar alternatives associated to scalar and negative polarity items. This feature can be assigned two values: $[\pm\ \sigma]$. Selecting $[+\sigma]$ results in the activation of the scalar alternatives (ALTs henceforth); selecting $[-\sigma]$ results in the selection of the plain meaning in which ALTs are not active. The crucial point is that, whenever the feature $[+\sigma]$ is selected, then the constraint on strengthening applies and an exhaustivization operator **O** (which has a meaning akin to that of *only*) must be used. For our purposes, it suffices saying that the result of this mandatory operation always leads us to the selection of the strongest – most informative – interpretation of the sentence containing the scalar item. With respect to the theoretical debate introduced in section 2, this new formulation leaves place to the notion of a *strategy* on the one hand and to the notion of *default* on the other: if the choice of activating the alternative interpretations of a statement containing a scalar term is in the end a matter of a subjective choice (thus, optional), once the selection has been made and the alternative interpretations activated, then the choice of the stronger alternative is instead mandatory. Very informally, the operator **O** applied to a sentence like (1) above, containing a scalar expression of the form "A or B" in which the ALTs are active will result in the derivation of the scalar implicature associated to *or*: **O** (singing *or*$_{[+\sigma]}$ dancing) = *only* (singing *or* dancing) = *only* (singing *or* dancing) and not (singing *and* dancing), thus excluding sentence (2) and deriving the inference in (3). The choice between activating the set of alternatives or not is considered optional in case of scalar terms while their activation is mandatory in case of NPIs. We won't pursue further the discussion on the parallelism with NPIs (this goes beyond the purposes of the present paper) but it's interesting to report a generalization on SIs already reported in Chierchia, 2002: "(Ordinary) scalar implicatures are suspended in the contexts that license *any* (as a Neg Pol or as Free Choice Item)". Typically, these are the contexts defined as Downward Entailing (or Downward Monotone), i.e. those contexts that licence inferences from sets to their subsets. For example, the antecedent of conditional represents a canonical DE context, in contrast with the consequent of conditional, which represents an Upward Entailing context instead, allowing only inferences from a set to its superset. Crucially, adding an implicature in DE contexts leads to a weakening of the overall assertion (given that informativity is "reversed" in DE contexts), while it leads to a strengthening in case the scalar term appears in a NON-DE context. Considering our tendency to be maximally informative and the monotonicity properties of the context, with respect to sentences (4), representing a DE context, and sentence (5), representing a NON-DE context, the distributional generalizations in (6) can thus be derived:

(4) If the troll is singing or dancing then he's happy (=DE)

(5) If the troll is happy, then he is singing or dancing (=NON-DE)

(6)

(a) The exhaustive interpretation (via application of the operator **O)** of a scalar term is easier in a NON-DE than in a DE context;
→ SI computation is easier in (5) than (4) (increased informativity)

(b) Having an implicature embedded in a DE context is way harder than having it embedded in a NON-DE context
→ SI computation is harder in (4) than (5) (loss of informativity)

(c) The flip between having an implicature and not having it is relatively easy in NON-DE contexts
(activation or de-activation of ALTs)

(d) The flip between having an implicature and not having it is hard in a DE context (loss of informativity)

These predictions have been specifically tested in the experimental study that I'm going to present in the next section.

4 A reaction-time study

As we have seen, Chierchia's proposal makes clear-cut predictions as to when the derivation of SIs is expected, also in relation to the type of syntactic context in which the scalar term operates. In this respect, and in consideration of the debate on the "cost" of SI computation reported in section (2), the experiment I'm going to present addresses the following questions:

(7)
(i) whether one of the candidate interpretations constitutes the preferred one depending on the syntactic environment it appears in (to this purpose, the rate of acceptance/rejection of critical sentences across conditions will be considered);
(ii) whether the derivation of the implicature is a costly process (to this purpose, the analysis of reading times (RTs henceforth) will be analysed).

4.1 Participants

A total of 30 subjects participated in this experiment. Participants were mainly 1st year students at the Psychological Faculty of the University of Milano-Bicocca, and received credits for their participation.

4.2 Procedure

The experiment was realised using E-Prime. Subjects were tested in a quiet room using a laptop and after a training session. Participants' task was to evaluate sentences as "true" or "false" with respect to a scenario constituted by four pictures that appeared on the screen. After an introductory screen in which characters and objects were presented for the first time, critical material was presented as follows (by pressing the space bar on the keyboard): at the top of a black screen a sentence appeared (in white). Participants were instructed to read (silently) the sentence and then press the space bar key to see the four pictures describing the situation against which they had to evaluate the sentence as "true" or "false". By pressing the bar, the four pictures appeared on the screen in the space below the sentence (in a random order). To answer, subjects had to press one of two highlighted keyboard keys: a green key for "true" and a red key for "false". After pressing it, they were either asked to move on by pressing the space bar (whenever their answer was "false") or, in case they answered "true", they had to answer another question that appeared in the middle of the screen (the four pictures remained there): "How much do you think the sentence is a good description of the situation represented in the pictures?" They were given a scale of response varying from 1 (bad) to 5 (good). Only time to answer the True/False question was recorded, starting from the moment they pressed the bar to make the pictures appear on the screen till they pressed the True/False key. Critical conditions were treated as a within subject factor: each subject was shown the complete battery of the material but saw only one occurrence per each critical item-type, for a total of 17 test items, 4 of which were critical test sentences containing *or* and the others were fillers.

4.3 Material

To avoid interferences from extra-linguistic factors on the interpretation of sentences, all the material presented in this experiment contained only fantasy names for characters and objects. Characters were in fact introduced as inhabitants of weird planets with their bizarre objects, unfamiliar to inhabitants on Earth.

The experiment presented a 2×2 condition design, in which two conditions were created as a within subject factor, each displaying 2 different levels. Condition I represents the type of syntactic environment in which the disjunction appears. The monotonicity properties of the context is varied, as summarized in (8): in sentences of type (a) *or* is embedded in the antecedent of conditional, which crucially constitutes a DE environment, like (4); on the contrary, in sentences of type (b) *or* is embedded in the consequent of the conditional, which constitutes a NON-DE environment like (5) above.

(8) Condition I: monotonicity of the context
(a) *If a P has an A or a B, then he also has a C* [= DE context]

(b) *If a P has a C, then he also has an A or a B*
[= NON-DE context]

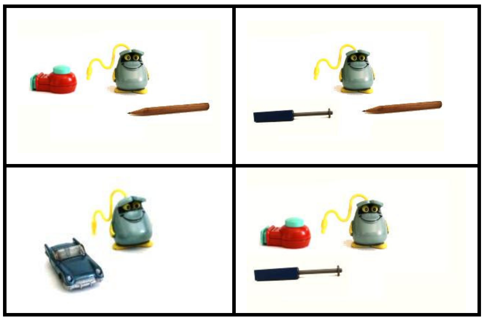
Each sentence was presented to each subject in two different critical situations, corresponding to levels S1 and S2 of Condition II (see (9) below). Each situation modulated the interpretation associated to the scalar term contained in the sentences by means of the scenario represented by the set of four pictures.

(9) Condition II: situations
S1: a situation representing the *exclusive* interpretation of *or*;
S2: a situation representing the *inclusive* interpretation of *or*.

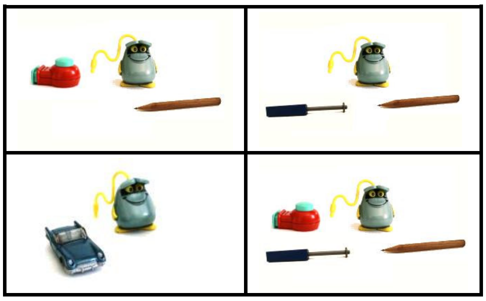
Consider, for example, the following test sentences (recall that fantasy names were used):

(10)
(a) If a Glimp has a curp or a dorf, then he also has a pencil
(b) If a Glimp has a pencil, then he also has a curp or a dorf

These were tested (on different subjects) in the following scenarios, representing conditions S1 and S2 respectively:



S1: only compatible with *exclusive* interpretation of *or* (see last picture: A and B but not C).



S2: only compatible with *inclusive* interpretation of *or* (see last picture: A and B and C)

Please note that the only crucial difference between the two scenarios is represented by the last picture in the sequence (remember that, during the experiment, the order of presentation of the four pictures was completely randomized across items and subjects). Crucially, scenario S1 is only compatible with the *exclusive* interpretation of *or*, which is the most informative in case of sentences of type (b), i.e. in a NON-DE context, but not of sentences of type (a), i.e. in a DE context. On the contrary, scenario S2 is only compatible with the *inclusive* interpretation of *or*, which is the most informative in case of sentences of type (a) but not of sentences of type (b).

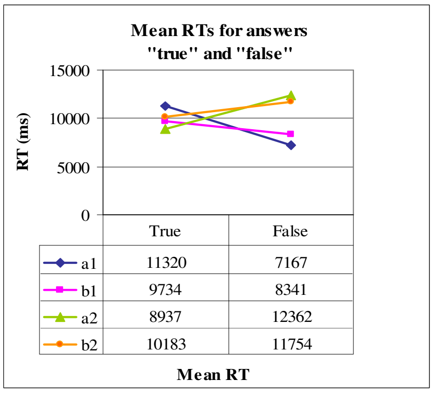
4.4 Results

Results are summarized in the table below, divided per type of sentences which crucially differ in their monotonicity properties: the 2nd column reports the type of scenario in which the sentence is evaluated (recall that S1 corresponds to the *exclusive* interpretation of *or* while S2 corresponds to the *inclusive*); the 3rd column reports the percentage of "true" answers followed by the rate assigned to the scale that appears in the 4th column; the last three columns report respectively: the response times to answer "true", to answer "false" and the mean total response time per condition.

Sent.	**Sit.**	**True**	**Scale rate**	**RTs for True**	**RTs for False**	**Mean RTs**
(a) DE	S1 (exc)	**57%**	3.47	11320	7167	9628
	S2 (inc)	**90%**	3.81	8937	12362	9291
(b) NON -DE	S1 (exc)	**87%**	4.38	9734	8341	9549
	S2 (inc)	**77%**	4.04	10183	11754	10562

Data on critical items can be analyzed with respect to different parameters: percentage of "true" and "false" answers; time taken to make a decision between "true" and "false"; grade assigned to the scale. I will focus here on the main findings. First of all, a large majority of subjects (90%) accept (a) sentence in Condition S2, compatible with the *inclusive* interpretation of *or*, while only half of them (57%) accept it in S1, where *exclusive* interpretation of *or* is represented. This difference is statistically significant (t(29)=-3.34, p<.01). In the second place, the rate of acceptance of the (a)-sentence in Condition S1 (representing the *exclusive* interpretation) is also significantly different from the rate of acceptance of the (b)-sentence (representing a NON-DE context) in the same condition (57% vs. 87%, t(29)=-3.07, p<.01). Moreover, those subjects that accepted the sentences in scenario S1 assigned a significantly lower score to (a) than (b) sentences (t(41)=-2,59, p<.01).

Data reported in the Table above are also interesting in another respect: reaction times to evaluate critical items in different conditions can be compared, considering overall mean RTs per sentence-type or distinguishing between RT to answer "true" and "false" separately, as plotted in the graph below.



A first point worthy of remark is the fact that no significant difference emerges taking context (DE vs. NON-DE) or scenario (*inclusive* vs. *exclusive*) as critical factors. These results seem to indicate that the processing load required to evaluate both types of sentences in both conditions was almost identical, at least if we consider mean RT overall. However, this consideration should be handled with care, given that one needs to integrate the overall picture with the data plotted in the graph, showing RTs for both sentence types and situations but differentiated between "true" and "false" type of answer.

Let's discuss Relevance Theory predictions first. According to this approach no difference due to the monotonicity properties of the two contexts is in principle to be expected. In fact, according to that approach, analysis of RTs should reveal a "cost" of scalar implicature computation. In this respect, the first crucial comparison is the one between RTs for answering "true" between situations S1 and S2 and a comparison on RTs for answering "false" between the same conditions. The second comparison to reveal the "cost" of implicature is the one between the RTs for answering "true" and the RTs for answering "false" within the same condition. None of these comparisons, however, turned out statistically significant.

Most interestingly, among RTs, only one comparison revealed statistically significant. Precisely, this was the time to answer "true" in situation S1 in case of sentence (a) compared to the mean time to answer "false" when evaluating the same sentence in the same condition (t(29)=5.16, p<.001). This reflects the fact that subjects that derived the implicature in case of DE context did it at a "cost". This finding is crucial in two respects: the same presumptive "cost" did not emerge from any other comparison, contrary to the Relevance Theory's expectations; also, this was the only "hard step" predicted by the distributional generalizations outlined in (6) derived from Chierchia's theory.

4.5 Discussion

One of the questions addressed in this experiment was the influence of the syntactic context on SI computation, ultimately the effect of the monotonicity properties of the context on informativity. Considering the acceptance rate first, we can say that the results obtained confirm our predictions. In the first place, subjects treat the two sentence types differently in the two situations; secondly, they derive SIs more when *or* appears in a NON-DE than in a DE context; thirdly, they prefer not to derive the SI when *or* appears in a DE context. The second question raised in (7) above asked whether the process of computing implicatures is costly. According to the framework I am adopting, no cost is to be associated to scalar implicature computation *per se*, contrary to Relevance Theoretic approaches. A cost is instead to be expected when the implicature is derived despite the fact that the scalar term is embedded in a DE context, in which the adding of the implicature would result in a weakening of the overall assertion. This prediction seems largely supported by the results: only those participants that accept the (a)-sentence in S1, thus deriving the implicature in a DE context, took significantly longer than the participants that reject that sentence in the same condition. If the cost were to be attributed to implicature computation in general, then the same contrast should be found in case of sentence (b), but this is not so. To account for the data obtained in this task, the claim that implicature computation *per se* is costly is, in my opinion, to be rejected.

In summary, the claim that the default interpretation of the scalar term depends on the monotonicity properties of the context in which the scalar term is embedded is largely supported by the data obtained in this experiment: without such a claim, it would be difficult to account for the fact that sentence (a) in which *or* appears in a DE context, when evaluated in scenario S1 representing the *exclusive* interpretation, is the hardest condition of all, both in terms of subjects' distribution, scale rate and RTs.

5 Conclusions

The experimental results presented here seem to be in contrast with recent works on SI computation realized within the Relevance Theoretic tradition. In particular, I'm referring to the works by Noveck and Posada (2003), Bott and Noveck (2004), Breheny et al. (2005) and Katsos et al. (2005). By means of different techniques, these authors conducted on-line experiments on adults while evaluating underinformative sentences containing scalar terms such as *some* and *or* in different experimental "situations" (as for, e.g., presence or absence of a preceding biasing context, or different instructions/suggestions given to participants to fulfil the task). Very generally, the results of these studies seem to point to the same direction, namely: whenever subjects compute the implicature associated to a scalar term, they do it at a cost. This is reflected by a slow down in correspondence of the scalar trigger when measuring reading times (like in the studies presented by Breheny and colleagues and Katsos and colleagues), or by an increased time to process the whole sentence (when measuring reaction times, like in the Bott and Noveck's study or in the ERP study conducted by Noveck and Posada). These results were uniformly interpreted by these authors as evidence of the fact that scalar implicature computation is a costly process. Without entering too much in the details of each single study, I would like to make some general considerations about the findings of the works mentioned above. In the first place, let's consider subject's distribution. It's interesting to note that in most (if not all) cases subjects split when they have to judge an underinformative sentence, even when the sentence is given "out of the blue", i.e. in the absence of a preceding context (this finding was also replicated in the experiment presented here). This is a clear indication, according to my view, that subjects are adopting a strategy to which they stick when solving the experimental task: half of the subjects consider the computation of the implicature "relevant enough" (to borrow from Relevance Theory terminology) and thus add the implicature; the other half, instead, keep to the plain meaning of the scalar term, and do not derive the implicature. I believe that the solution proposed by Chierchia (2006) well explains these facts, being feature selection the result of a subjective choice, and also being the flip between having or not having the implicature in NON-DE contexts way easier than in DE contexts. On the contrary, it's more difficult to find a ready explanation of this split in subjects' distribution within the Relevance Theory given that the presumption of optimal relevance of a given stimuli should in principle be the same across all participants.

On the other hand, RT data seems at first glance to be better explained by Relevance Theory. The crucial comparison, according to this approach, is between the RTs of the two groups: subjects that derive the implicature always take longer than the rest. This result is sufficient, according to them, to claim that the process of computing SI is costly and thus subjects only derive SIs when the benefits obtained by the adding of a SI exceed the processing effort required for its derivation. However, it's not that clear that this overload is effectively due to the adding of the implicature *per se*. As the results in the experiment presented here show, the only "cost" is found when the implicature is added despite the fact that it leads to a weakening of the overall assertion. As we said, this "flip" is predicted to be hard in Chierchia's generalization and this loss in informativity, and not implicature computation, seems to be the source of this "cost".

In the end, I believe that the intriguing debate on pragmatic inference, which has very recently attracted the interest of psycholinguists, is far from being solved. To begin with, the majority of the studies have been focused on measuring the "cost" of implicature derivation. Though interesting, I think this is not the *only* question to be solved within a semantic-pragmatic theory of Scalar Implicatures.

References

Bott, L. and I. Noveck (2004). "Some utterances are underinformative: The onset and time course of scalar inferences." *Journal of Memory and Language* **51**: 433-456.

Breheny, R., N. Katsos, et al. (2005). "Are generalized scalar implicatures generated by defaults? An on line investigation into the role of context in generating pragmatic inferences." *Cognition*: 1-30.

Chierchia, G. (2006). "Broaden Your Views: Implicatures of Domain Widening and the "Logicality" of Language." *Linguistic Inquiry* **37**(4): 535-590.

Foppolo, F. (2007). *The logic of pragmatics. An experimental investigation with children and adults.* Unpublished Ph.D. dissertation, University of Milano-Bicocca.

Levinson, S. C. (2000). *Presumptive Meanings - The theory of Generalized Conversational Implicatures.* Cambridge, MA, MIT Press.

Noveck, I. and A. Posada (2003). "Characterizing the time course of an implicature." *Brain and Language* **85**: 203-210.

Sperber, D. and D. Wilson (1986). *Relevance: Communication and Cognition.* Oxford, Blackwell.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 133–140.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Incredulity Questions

Ariel Cohen
Ben-Gurion University, Israel
arikc@bgu.ac.il

Abstract

Incredulity questions have a double nature: on the one hand, they are questions, while, on the other hand, they are statements of incredulity or indignation. Hence, a multidimensional account of their interpretation is attractive. Artstein (2002) proposes a multidimensional account of a similar phenomenon—echo questions. He argues that the expression that is questioned is focused, and, using Rooth's (1985; 1992) alternative semantics, suggests that the interpretation of the echo question is its focus semantic value.

While similar, incredulity questions differ from echo questions in both form and meaning. They have a different intonation pattern, where incredulity is expressed by expanded pitch range, rather than by focus. Incredulity questions also have a different interpretation: they are not used to recover some information that was misheard or misunderstood, but to express incredulity or indignation about a statement that was heard and understood perfectly. Yet, Artstein's approach can be extended to handle incredulity questions, if, instead of the focus semantic value, we use a new semantic value, the *world semantic value*, which considers alternative possible worlds. Thus, an incredulity question expresses the claim that in none of the speaker's belief (or normative) worlds is the echoed statement true—hence the incredulity (or indignation) expressed toward that statement.

1 Introduction

Suppose Bill hears Ann uttering (1.a); in response, Bill utters (1.b) or (1.c) (capitals indicate pitch accent—the interpretation of this pitch accent will be discussed momentarily).

(1) a. John is going to get the job.
b. B: JOHN is going to get the job?!
c. B: WHO is going to get the job?!

How are we to interpret Bill's utterances?

On the one hand, they look like questions—specifically, echo questions. Bill's utterances end with rising intonation, and they can get the same sort of answer that a genuine question would elicit. Thus, "yes" and "John" are possible (though perhaps not very helpful) responses to (1.b) and (1.c), respectively.

On the other hand, however, Bill's utterances are not genuine questions. Bill is not seeking information; we can safely assume that Bill understood what Ann was saying. The point of Bill's utterances is to express incredulity. Bill does not question the fact that John will, indeed, get the job, but expresses surprise—this is not at all what Bill expected, so much so that it is hard for Bill to believe it. For instance, Bill may believe that John is an extremely unsuitable choice, so that his appointment is incredible. Bill may also express indignation: he may be interpreted as saying that John's appointment is bad, unethical, unjust, or the like. For example, Bill may have received a promise to get the job himself, and John's appointment breaks this promise.

Note that when Ann responds, she may, and usually will, relate to the incredulity or indignation aspects of Bill's utterance, rather than treat it as a question. Thus, it would be quite felicitous of her to offer some sort of explanation or justification, as in (2.a) or (2.b):

(2) a. John is actually a good choice, but he never got a chance to show his true ability.

b. I am sorry, I know I promised you the job, but the big boss forced me to appoint John.

This dual aspect of incredulity questions is demonstrated nicely in the following excerpt from *Sointula*, by Bill Gaston:

> "I...want to be rid of this whiskey before I tackle the West Coast Trail."
>
> "YOU'RE doing the trail?"
>
> Gore sees Bob scan his body while asking this and he hears incredulity in the question.
>
> "Yes." He pauses. "Why?" (p. 92)

When Gore says "yes", he is answering the question aspect of Bob's utterance; when he asks "why?", he is questioning why Bob is expressing incredulity.

So, incredulity questions, like (1.b) and (1.c), have aspects of a question, and also aspects of an assertion (or, perhaps, an expressive). An understanding of incredulity questions, therefore, is important from a theoretical point of view, in that it combines with an increasing body of work on constructions that can express more than one meaning simultaneously, and provides clues to their proper treatment.[1]

The study of incredulity questions is also important from a practical-computational point of view. Clearly, a question answering system needs to respond to an incredulity question differently from the way it responds to a genuine question: to provide helpful feedback to the user, the system should supply some justification or explanation (Carberry 1989; Lambert and Carberry 1991; Chin 2000). Consider, for example, the following exchange from a system that helps students register for courses (Lambert and Carberry 1991):

(3) **User:** When does CS400 meet?
System: CS400 meets on Mondays, 7–9p.m.
User: CS400 meets at night?

The user's second utterance is clearly an incredulity question. The user seeks some explanation for why the course is taught at such an unusual time. A simple answer of "yes" would clearly be inappropriate; what the user really wants is some sort of explanation or justification for this surprising fact.

In this paper I provide a semantics of incredulity questions, which is compatible with their double nature. I explain why they look like questions, yet can be interpreted as assertions, and how their interpretation is related to their intonation.

The rest of the paper is organized as follows. In section 2 it is argued that the double nature of incredulity questions calls for a multidimensional approach. In section 3 I discuss a multidimensional approach to a related phenomenon—echo questions. Section 4 argues that the crucial element of a multidimensional theory of incredulity question consists of referring to alternative possible worlds. Section 5 formalizes this idea, and section 6 demonstrates how the formalization accounts for the properties of incredulity questions.

2 A Multidimensional Theory

Since incredulity questions have a dual aspect, it makes sense to account for them with a multidimensional theory: a theory according to which an expression may have more than one semantic value.

Asher and Reese (2005) propose such a theory. They assign to incredulity questions a

[1] See Potts (to appear) for a recent discussion.

complex semantic type: *question* • *assertion.* Taking the standard view (Hamblin 1973) that the meaning of a question is the set of its potential answers C, Asher and Reese take the assertion to be the claim that one of these answers is unexpected:

(4) $\exists p(p \in C \land \textbf{Expect}\neg p)$

Pragmatics then makes sure that the previously mentioned answer is selected as unexpected. Thus, (1.b) and (1.c) are questions, but they are also assertions that one of their potential answers, namely (1.a), is unexpected.[2]

I think this view, according to which the incredulity aspect is related to the question aspect, is essentially correct. In this paper, I suggest a way of deriving this interpretation from more general principles.

3 Echo Questions

Incredulity questions are often treated as a kind of echo questions, because they share many syntactic properties (Authier 1993). Looking at the semantics of echo question may therefore help us figure out the meaning of incredulity questions.

Artstein (2002) proposes a theory of echo questions that is particularly attractive for our purposes, because it is multidimensional at a fundamental level. Specifically, Artstein follows *alternative semantics* (Rooth 1985; 1992). Rooth argues that every expression ϕ has two semantic values: in addition to the ordinary semantic value, $[\![\phi]\!]^O$, ϕ also has a *focus semantic value*, $[\![\phi]\!]^F$, which is a set of alternatives to the focused element(s) of ϕ.

According to Artstein, echo questions have a distinctive contour with a rising pitch accent (L+H* in the notation of Pierrehumbert 1980), and a high-rising boundary (HH%). He argues that this pitch accent is an instance of focus; one of the reasons for this claim is that, just like focus, echo questions can appear on parts of words:

(5) a. She believes in WHAT-jacency?
b. John witnessed a great reve-WHAT-tion?
c. Bill is a WHAT-dontist?

Thus, the focus semantic value of (1.b) will be a set of alternative propositions of the form:

(6) {John is going to get the job,
Mary is going to get the job,
Julie is going to get the job,
...}

This set of alternatives corresponds to a question inquiring which of these alternative propositions was asserted.[3]

Sentence (1.c) looks like a *wh*-question; however, Artstein argues that it is not a genuine question, because such sentences do not obey locality restrictions. Instead, he argues that the *wh*-word is focused, and the interpretation of (1.c) is its focus semantic value; which is the same as that of (1.b). Thus, (1.b) and (1.c) have the same semantics (though they may have different pragmatics).

Artstein can therefore account for the question aspect of an echo question: it is used when one interlocutor failed to understand or hear clearly what the other one is saying. Thus, if we interpret Bill's utterances in (1.b) or (1.c) as echo questions, the implication is that he did not hear clearly, and is seeking confirmation about the identity of the person who will get the job.

[2] To be precise, Asher and Reese only treat incredulity *assertions*, such as:

(i) a. A: I'd like you here tomorrow morning at eleven.
b. B: !Eleven in the morning!

They do, however, treat other complex type questions in a similar way, so I believe the above is a faithful presentation of their view.

[3] Of course, one ought to be more precise, and replace these glosses with whatever one's favorite theory says that the semantic value of propositions is. I will return to this issue in section 6 below.

Artstein acknowledges that, in addition to clarification-seeking echo questions, there are also cases where an echo question is used to express incredulity or indignation about some proposition, usually the previous utterance or an entailment of it. However, he does not explain how these particular aspects of the meaning follow from his system: how does it follow that if Bill is inquiring about the identity of the person who got the job, then Bill knows it is John, but expresses incredulity or indignation about the fact?

Moreover, incredulity questions differ in their intonation from pure echo questions. In fact, they have a tune similar to that of ordinary declaratives, except that, being questions, they have a final rise rather than a final fall (Moulton 1987). The meaning of incredulity is expressed not through the tune, but via an expanded pitch range (Hirschberg and Ward 1992; Herman 1996; Jun and Oh 1996; Lee 2005). Another difference is that incredulity questions cannot apply to parts of words: the utterances in (5) above do not have an incredulity or indignation interpretation.

Thus, while for echo questions a case can be made that the pitch accent is associated with focus, an analogous case for incredulity questions would be hard to make. Nonetheless, I believe that Artstein's insight, namely that incredulity questions, just like echo questions, involve reference to a set of alternatives, is correct. In the next section I propose an extension of his multidimensional approach, which can handle the meaning of incredulity questions.

4 Considering Alternative Worlds

So, when Bill utters (1.b) or (1.c), his utterance invokes a set of alternatives. But alternatives to what? Clearly, the alternatives have something to do with John; but Bill is not considering alternative candidates for the job, because he heard and understood that John is the one.

One may suggest that Bill *is* considering alternative candidates for the job, but his utterance is a rhetorical, rather than a genuine question, since he already knows the answer. But this will not do. Normally, a rhetorical *wh*-question is interpreted as implying that the answer is the empty set.[4] For example:

(7) a. Who believes such nonsense? (Bolinger 1957:158)

b. When has he ever said a word against his mother? (Horn 1978:151)

c. What difference does it make? (Quirk *et al.* 1985:826)

The rhetorical question in (7.a) implies that nobody believes such nonsense, (7.b) implies that he has never spoken against his mother, and (7.c) implies that it makes no difference.[5]

So Bill's question does not involve alternative candidates for the job. Instead, I suggest that Bill is considering alternative worlds. The incredulity or indignation interpretations are then generated as follows.

The identity of the alternative worlds depends on the modal base (which, in turn, is dependent on the context). The modal base can be doxastic, i.e. the alternative worlds are Bill's belief worlds: in each one of these worlds, some candidate is getting the job. Bill is then asking us to find a world among them in which John gets the job. This is a rhetorical question, because Bill already knows the answer—he hardly needs us to tell him what's in his belief worlds!

Therefore, when Bill is asking about his belief worlds, he is implying that the answer is the

[4] I am using the neutral term "implying", since it is not relevant to our discussion here whether this is an entailment, a presupposition, or an implicature.

[5] There are well known exceptions to this generalization, such as (i), as said by a mother to her son, which clearly expects the answer "mother".

(i) Who fed you and gave you a proper education? (Han 2002:218, note 6)

But even in such cases, the implication is that nobody *besides* the addressee's mother is a true answer to the question. But this is not the point of (1.b) or (1.c); Bill is not drawing attention to the fact that nobody else is going to get the job, but expresses his incredulity or outrage at John's getting it.

empty set: i.e. in none of his belief worlds does John get the job. Hence, his getting the job is incredible. Of course, we are concerned here with Bill's belief worlds *before* Ann spoke; after he heard Ann and accepted what she said, Bill's belief worlds will obviously contain the fact that John will get the job.

Alternatively, the modal base may be deontic. In this case, Bill refers to worlds that are permissible, given his norms. Once again, this is interpreted as a rhetorical question, since Bill's norms are obviously known to himself. Therefore, Bill is implying that in none of these worlds does John get the job. This is how the indignation interpretation is generated: the appointment of John to the job constitutes a violation of Bill's norms of conduct.

The focus semantic value is not able to generate this reading. But a different sort of semantic value might.[6] We need a semantic value that can model expectations (and their violation), by taking into account possible worlds.

5 World Semantic Value

At this point, it would be a good idea to consider a different phenomenon where expectation is important. One such case is the interpretation of *many.*

It is well known that *many* is vague: there are no clear criteria for how many is many. Consider (8.a), for example, whose logical form is something like (8.b).

(8) a. Many academics watched the 2006 World Cup.

b. $\mathbf{many}(\mathbf{academic}, \mathbf{watch\text{-}WC})$

Sentence (8.a) would be true iff the proportion of academics who watched the 2006 World Cup is higher than some threshold. That is to say, it would be true iff

$$(9) \qquad \frac{|[\![\mathbf{academic}]\!] \cap [\![\mathbf{watch\text{-}WC}]\!]|}{|[\![\mathbf{academic}]\!]|} > \rho,$$

for some parameter ρ. [7] The question is, then: what is the value of this threshold ρ?

In a well known paper, Fernando and Kamp (1996) propose to solve this problem as follows. They suggest that $\mathbf{many}(\psi, \phi)$ is true iff it could well have been the case that fewer ψs are ϕs. In other words, there are more ψs that are ϕs than expected. For example, (8.a) means that more academics watched the 2006 World Cup than one would expect of academics. Fernando and Kamp formalize this notion of expectation using probabilities over possible worlds.

Cohen (to appear) provides a multidimensional account of Fernando and Kamp's proposal by proposing a new type of semantic value: *world semantic value*, $[\![\phi]\!]^W$, which takes into account alternatives to the world of evaluation of ϕ. $[\![\phi]\!]^W$ is a set: each member of the set is the ordinary semantic value of ϕ in some world. For example, if ϕ is a property, $[\![\phi]\!]^W$ is a set of sets of individuals. Every member of $[\![\phi]\!]^W$ is a set of individuals that are in the extension of ϕ in some world.

Can these sets overlap? There are reasons to believe that the answer is no. Von Fintel (1997, note 2) argues for using Lewis's (1968, 1971, 1986) *counterpart theory* in accounts of natural language quantification. If that is so, then the individuals in different worlds are different. Hence, the sets that are members of $[\![\phi]\!]^W$ are disjoint.

If we then apply union to the world semantic value of the property, $\bigcup [\![\phi]\!]^W$, we get the set of all individuals that are in the extension of ϕ in some world.

In the case of (8), the union of the world semantic value of the restrictor, $\bigcup [\![\mathbf{academic}]\!]^W$, is the set of possible academics, i.e. individuals who are academics in some world. The union of the world semantic value of the scope, $\bigcup [\![\mathbf{watch\text{-}WC}]\!]^W$, is the set of individuals who watched the 2006 World Cup in some world.

[6] For proposals involving other semantic values, in addition to Rooth's focus semantic value, see Büring (1997; 1999) and Cohen (to appear).

[7] As is well known, (8.a) also has a reading where the absolute number of academics who watched the 2006 World Cup, rather than the proportion, is considered, and probably other readings as well. The ambiguity of *many*, however, does not concern us here.

Now consider the probability that something is a ϕ in some world, given that it is a ψ in some world:

(10) $P(\bigcup [\![\phi]\!]^W \mid \bigcup [\![\psi]\!]^W)$

Since individuals in different worlds are different, (10) is the probability that if an individual in some world is a ψ, then it is a ϕ. This is precisely the expectation that a ψ is a ϕ, required by Fernando and Kamp's theory. **many**(ψ, ϕ) is true, then, just in case the proportion of ψs that are ϕs is greater than this expectation:

(11) $$\frac{|[\![\psi]\!] \cap [\![\phi]\!]|}{|[\![\psi]\!]|} > P(\bigcup [\![\phi]\!]^W \mid \bigcup [\![\psi]\!]^W).$$

In the case of (8), the threshold is the probability

(12) $P(\bigcup [\![\textbf{watch-WC}]\!]^W \mid \bigcup [\![\textbf{academic}]\!]^W)$

This is the probability that someone who is an academic in some world, watched the 2006 World Cup in that world.

Thus, (8) is true iff the proportion of academics who watched the 2006 World Cup is higher than the expectation that an academic watched the 2006 World Cup:

(13) $$\frac{|[\![\textbf{academic}]\!] \cap [\![\textbf{watch-WC}]\!]|}{|[\![\textbf{academic}]\!]|} > P(\bigcup [\![\textbf{watch-WC}]\!]^W \mid \bigcup [\![\textbf{academic}]\!]^W)$$

These appear to be the correct truth conditions.

6 Tying It All Together

We now have everything in place to account for incredulity questions. Consider (1.b). The expanded pitch range is used to indicate that the world semantic value of *John* ought to be considered. This is the set of counterparts to John in each one of Bill's belief (or normative) worlds:

(14) $\{\text{John}_{w_1}, \text{John}_{w_2}, \text{John}_{w_3}, \ldots\}$.

How do we combine the world semantic value of *John* with the other elements of the sentence? We can follow the same procedure as the one used for computing the focus semantic value. Rooth (1985; 1992) suggests that the focus semantic value of an expression is computed compositionally, using the ordinary semantic rules to combine the focus semantic values of its parts.[8]

Thus, the world semantic value of (1.b) can be glossed as (15).

(15) $\{\text{John}_{w_1}$ is going to get the job,
John$_{w_2}$ is going to get the job,
John$_{w_3}$ is going to get the job,
$\ldots\}$

This is merely a gloss; to make it precise, one needs to replace the set of English sentences with a set of propositions. But we need to be careful: when we consider the world semantic value of a proposition, it makes little sense to take propositions to be sets of possible worlds, since possible worlds are precisely what we are abstracting away from. In the case of (15), we will get the undesirable result that each member of the world semantic value is either a singleton set $\{w_i\}$, if John$_{w_i}$ is going to get the job, or the empty set, if John$_{w_i}$ is not going to get the job.

But recall that we are using counterpart theory. In this theory, there are independent reasons for not treating propositions as sets of possible worlds (Dorr 2005). Instead, we ought to use some form of *structured propositions*. Indeed, Lewis (1986) himself suggests such a representation. Thus, the proposition expressed by *"a is P"* is an ordered pair $\langle a, P' \rangle$, where P' is the set of all individuals, in all worlds, that have the property P.[9] This proposition is then true iff the first element in the pair is a member of the second: $a \in P'$.

For example, the proposition expressed by (16.a) is (16.b), and is true iff John is one of

[8] Though see Cohen (1999) for some problems with this view.

[9] So, in effect, $P' = \bigcup [\![P]\!]^W$.

the individuals that are going to get the job in some world. Since every individual can occur in one world only, this is equivalent to (16.c), and is true iff John (in the world of evaluation) is going to get the job, as desired.

(16) a. John is going to get the job.
b. $\langle$John,$\{x|x$ is going to get the job in some world$\}\rangle$
c. $\langle$John,$\{x|x$ is going to get the job$\}\rangle$

Applying this view of propositions, the resulting world semantic value of (1.b) is a set of propositions of the form:

(17) $\{\langle \text{John}_{w_1},\{x|x$ is going to get the job$\}\rangle$,
$\langle \text{John}_{w_2},\{x|x$ is going to get the job$\}\rangle$,
$\langle \text{John}_{w_3},\{x|x$ is going to get the job$\}\rangle$,
$\ldots\}$

What Bill is actually asking, then, is this: in which world (among my doxastic/deontic alternatives) is John going to get the job? This is a rhetorical question: Bill knows better than anyone else what happens in his belief or normative worlds. Hence, it is taken to be a question that implies that none of its possible answers is correct. Therefore, Bill is implying that (prior to Ann's utterance) in none of his belief worlds does John get the job (which is why he is incredulous), or that in none of his normative worlds does John get the job (which is why he is indignant). Again, it is important to emphasize that the rhetorical question is about Bill's belief/normative worlds, not about the identity of the person who is going to get the job.

Following Artstein (2002), the meaning of a *wh* incredulity question like (1.c) is the same. The utterance does not receive a normal question meaning; instead, the world semantic value of the *wh*-word is used, just like in the case of a non-*wh* incredulity question. Indeed, the responses in (2) are felicitous answers to both (1.b) and (1.c). Of course, Ann might choose to treat Bill's utterance as a normal question, rather than an incredulity question; in this case, the appropriate responses would differ: "yes" for (1.b), and "John" for (1.c).

Thus, an incredulity question has a dual aspect. On the one hand, it really is a question: Bill is asking in which world John is going to get the job. If Ann answers *yes* to (1.b) or *John* to (1.c), she is indicating such a world—the actual world (though this world may not be among Bill's belief or normative worlds). On the other hand, it is a statement of incredulity or indignation: by being rhetorical, the question implies that none of the possible answers are true, i.e. that the echoed statement is incredible or outrageous.

References

Artstein, R. 2002. A focus semantics for echo questions. In *Proceedings of the Workshop on Information Structure in Context*, ed. by Á. Bende-Farkas and A. Riester 98–107.

Asher, N., and Reese, B. 2005. Negative bias in polar questions. In *Proceedings of Sinn und Bedeutung 9*, ed. by E. Maier, C. Bary, and J. Huitink.

Authier, J. M. 1993. Nonquantificational *wh* and weakest crossover. *Linguistic Inquiry* 24.161–168.

Bolinger, D. 1957. *Interrogative Structures of American English: The Direct Question*. Tuscaloosa: University of Alabama Press.

Büring, D. 1997. *The Meaning of Topic and Focus—The 59th Street Bridge Accent*. London: Routledge.

—— 1999. Topic. In *Focus: Linguistic, Cognitive, and Computational Perspectives*, ed. by P. Bosch and R. van der Sandt 142–165. Cambridge: Cambridge University Press.

Carberry, S. 1989. A pragmatics-based approach to ellipsis resolution. *Computational Linguistics* 15.75–96.

CHIN, D. N. 2000. Planning intelligent responses in a natural language system. *Artificial Intelligence Review* 14.283–331.

COHEN, A. 1999. How are alternatives computed? *Journal of Semantics* 16.43–65.

—— to appear. No alternative to alternatives. *Journal of Semantics*.

DORR, C. 2005. Propositions and counterpart theory. *Analysis* 65.210–218.

FERNANDO, T., and KAMP, H. 1996. Expecting many. In *Proceedings of the 6th Conference on Semantics and Linguistic Theory*, ed. by T. Galloway and J. Spence 53–68 Ithaca, NY. Cornell University.

VON FINTEL, K. 1997. A minimal theory of adverbial quantification. In *Context Dependence in the Analysis of Linguistic Meaning 2: Proceedings of the Workshops in Prague and Bad Teinach* 153–193. University of Stuttgart Working Papers.

HAMBLIN, C. L. 1973. Questions in Montague grammar. *Foundations of Language* 10.41–53. Reprinted in Partee (1976:247–259).

HAN, C. H. 2002. Interpreting interrogatives as rhetorical questions. *Lingua* 112.201–229.

HERMAN, R. 1996. Final lowering in Kipare. *Phonology* 13.171–196.

HIRSCHBERG, J., and WARD, G. 1992. The influence of pitch range, duration, amplitude and spectral features on the interpretation of the rise-fall-rise intonation contour in English. *Journal of Phonetics* 20.241–251.

HORN, L. 1978. Some aspects of negation. In *Universals of Human Language*, ed. by J.H. Greenberg, C.A. Ferguson, and E.A. Moravcsikg volume 4: Syntax 127–210. Stanford: Stanford University Press.

JUN, S.-A., and OH, M. 1996. A prosodic analysis of three types of wh-phrases in Korean. *Language and Speech* 39.37–61.

LAMBERT, L., and CARBERRY, S. 1991. A tripartite plan-based model of dialogue. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics* 47–54.

LEE, O. J. 2005. *The Prosody of Questions in Beijing Mandarin*. Ohio State University dissertation.

LEWIS, D. 1968. Counterpart theory and quantified modal logic. *Journal of Philosophy* 65.113–126.

—— 1971. Counterparts of persons and their bodies. *Journal of Philosophy* 68.203–211.

—— 1986. *On the Plurality of Worlds*. Oxford: Blackwell.

MOULTON, W. G. 1987. On the prosody of statements, questions, and echo questions. *American Speech* 62.249–261.

PARTEE, B. H. (ed.) 1976. *Montague Grammar*. New York: Academic Press.

PIERREHUMBERT, J. 1980. *The Phonology and Phonetics of English Intonation*. Cambridge, MA: MIT dissertation.

POTTS, C. to appear. Into the conventional-implicature dimension. *Philosophy Compass*.

QUIRK, R., GREENBAUM, S., LEECH, G., and SVARTVIK, J. 1985. *A Comprehensive Grammar of the English Language*. London: Longman.

ROOTH, M. E. 1985. *Association with Focus*. University of Massachusetts at Amherst dissertation.

—— 1992. A theory of focus interpretation. *Natural Language Semantics* 1.75–116.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 141–147.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Group Dialects in an Online Community

Patrick G. T. Healey
Interaction, Media and Communication Group
Department of Computer Science
Queen Mary University of London
ph@dcs.qmul.ac.uk

Carl Vogel
Computational Linguistics Group
School of Computer Science and Statistics
Trinity College
Dublin
vogel@tcd.ie

Arash Eshghi
Interaction, Media and Communication Group
Department of Computer Science
Queen Mary University of London
eshghi@dcs.qmul.ac.uk

Abstract

Variations in group sub-languages evolve quickly and are a key marker of social boundaries such as those between professions, workgroups, tribes and families. In this paper we present a quantitative analysis of the effects of group structure on language use in naturalistic interaction. The data come from text chat interactions in an online social community. Using statistical techniques developed for the analysis of authorship attribution we use this corpus to test three accounts of the emergence of group sub-languages: a) local coordination mechanisms b) network topology and c) influential individuals. The results indicate that it is influential individuals who have the strongest effects on sub-language convergence.

1 Introduction

Language use is sensitive to a variety of social and cultural factors. Place of residence, education, religion, occupation, hobby, age group, expertise and ethnic origin can all influence people's use of e.g., words, syntax prosody, and style. Communicative alignment –similarity in the forms of language used by participants in an interaction– is consequently a key indicator, for members and analysts alike, of community co-membership (Clark, 1996; Clark, 1998; Gumperz, 1996).

Field studies have shown that communicative alignment indexes social organisation at quite fine-grained resolutions. For example, linguistic homogeneity is a criterion for distinguishing tribal groupings in ethnographic studies of hunter-gather societies (Dunbar, 1993). Communication in institutional environments is often characterised by local, institution-specific, forms of talk (Bergmann and Luckmann, 1994). Distinct sub-languages have been documented within different subgroups in a single workplace (Robinson and Bannon, 1991) and families also frequently develop their own jargon words and idioms.

Communal sub-languages can emerge rapidly. Experimental studies have shown that semantically distinct sub-languages develop in small groups in less than an hour of group interaction (Healey, 1997; Healey et al., 2007). This divergence can interere with the intelligibility of communication across community boundaries (Gumperz, 1996; Shaw and Gaines, 1988). People can also sometimes use their ability to switch between different codes and repetoires as a means of establishing alignment with, or exclusion of, others (Gumperz, 1982).

We can distinguish three logicially independent, but not mutually exclusive, hypotheses that have been suggested to account for group sub-language co-ordination:

1. Local Dialogue Coordination: patterns of co-ordination are explained by local, pair-specific, dialogue mechanisms that are common across interactions (Garrod and Doherty, 1994; Clark, 1996; Healey et al., 2007).

2. Network Topology: patterns of co-ordination are explained by differences in the patterns of interaction amongst the members of a population (Garrod and Doherty, 1994; Healey et al., 2007).

3. Influential Individual: patterns of co-ordination are explained by reference to key individuals who have a disproportionate effect on the language of others in the group (Garrod and Anderson, 1987).

In this paper we investigate how well these hypotheses account for the patterns of language use observed in a text-based online community. The data consists of all the interactions over a three day period in a group of 150 individuals. This provides a unique opportunity to carry out a quantitative analysis of the relationship between of patterns of interaction in this community and similarity of language use.

Although previous work has looked at patterns of interaction and the emergence of group norms in email (Postmes et al., 2000) we believe this is the first quantitative study of conversational interactions across a whole community. The natural, conversational character, of the exchanges (see below) and the scale of the analysis help to address some of the key limitations of case-based and experimental studies of group sub-languages.

First we provide background information about the character of the online environment: 'Walford' and the data used in the analysis. Then we present the statistical technique -unigram statistics developed for forensic linguistics and the results of our analysis.

2 Interaction in Walford

Walford is a text-based online social community or 'talker' that has been established for more than a decade. It has approximately 1500 regular users who are predominantly based in North America and Europe. It emerged as one of the many variants on James Aspene's 'TinyMUD'[1] which was first created in 1989. The environment is structured around a spatial metaphor with rooms, objects, players and exits. Once users have reached a sufficient level of expertise they can create their own rooms, objects and commands (macros).

The residents of Walford have taken advantage of this structure to build up a complex environment. There are shared public spaces such as a high street, a pub, a townhall, a bank, a bus station. There is also a rubbish dump and a network of roads. Although based on a MUD, people's main preoccupation in Walford is with their interactions and social relationships with each other. This is illustrated by a sample of conversation topics from the logs: chocolate, outsourcing, mobile phones, births and deaths in resident's (real) families, relationships (both inside and outside Walford), economics assignments, redundancy and boredom.

A sample conversational sequence from the Walford pub is provided in Excerpt 1. The extract helps to illustrate the conversational character of theses exchanges. Multi-installment turns, clarification questions and ellipsis are common features.

The data analysed in this paper come from a corpus of chat logs collected over approximately one year in 2004-2005. For each person-to-person the ID of the 'speaker', their virtual location, the recipients ID and their virtual location is recorded. In order to protect the anonymity of participants the names of people, characters, places,

[1] MUD stands for multi-user dungeon, from its text-based computer gaming tradition.

Table 1: A Sample Dialogue from the Queen Vic pub in Walford

A:	`Yeah dave is a cool guy...  Good mechanic...  Good guy.`
A:	`though I wouldn't be surprised if he was a wife/child beater.`
B:	`he seemed very gentle`
B:	`but he did drink a lot`
B:	`he an my dad share war stories now that they've both had their prostates removed`
B:	`ugh`
B:	`the last thing you want to hear two old guys chatting about`
A:	`war = prostate ?  or vietnam?`
B:	`hehe yeah prostate`

some commands and the name of the environment have been changed. Users agreed as a condition of use to the system to the logs being used, in anonymised form, for the purposes of research and publication.

3 Methods

A sample of three consecutive days of logs of interaction in Walford were randomly selected for anaylsis. The logs were preprocessed to remove all automatic formating and command names. This yielded a total of 20,043 turns by 150 unique identities from 148 unique locations.

To analyze the data, we applied statistical text classification methods (Van Gijsel and Vogel, 2003; Vogel and Brisset, 2006; Vogel, 2007). This approach draws on research on authorship attribution in forensic linguistics in which there is a preference for methods that do not use content analysis. This helps to ensure more robust inter-judge reliability and for this reason letter n-grams are favoured (Chaski, 1999). Surprisingly, letter unigrams have provided remarkable results to date. Although, it seems counter-intuitive that letter unigrams might be effective in identifying categories such as authorship or genre the relative efficacy of predictive text on mobile phones suggests what is possible. Consider also the way that Scrabble boards distribute their hundred letter tiles across the alphabet differently in German and in English; or notice the fact that a latinate vocabulary will have a noticeable distribution of the letter "Q" (e.g. "horse riding" vs. "equestrian"). These observations indicate how word choice impacts on orthography (poetry and lipograms aside, written text does not involve word choice on the basis of the spelling of words). This approach also has advantages over measures based on shared words. A long tail of words in any corpus corresponds to singleton occurrences and many will appear in one text and not another. In addition, the closed class words will all be shared and differently inflected forms of the same root may appear. Thus, subword analysis is necessary. Letter unigrams are thus a limiting case and, unlike words, constitute a closed class.

Here we use the chi-square divided by degrees of freedom (cbdf) statistic adapted from other work in comparing corpora (Kilgarriff, 2001; Kilgarriff and Salkie, 1996). The idea is to compare the n-gram distributions between two files in any category. The overall similarity between two files is computed as the sum of the chi-square values of each of the n-grams between the two files, relativized to the number of distinct n-gram types compared. A smaller cumulative chi-square value thus indicates a smaller difference between two files (note that this is the opposite of the normal, contrastive, use of the chi-square test in order to locate significant differences). The similarity metric is computed for all pairwise comparisons of relevant files. These similarity metrics can then be used to rank order the files by the categories they comprise. That is, one has a category of all of the texts by a single author, versus all of the other texts. Mann-Whitney tests can then be used to examine whether each file in a category fits best with its natural category or with some other category.

The Walford log is organised as a temporally ordered sequence of turns with speaker ID, location, recipient ID(s) and their location(s) (local or remote). In the analysis reported here, we used speaker ID's as categories. For each speaker the logs were separated into single files corresponding to each continuous sequence of interaction with each recipient group. For example, if A speaks to B for 5 turns then C for 5 turns then B again for 5 turns this creates three files for speaker A. If by contrast they alternate between A and B for 10 turns this creates 10 files for speaker A. Each file thus consists of a contiguous sequence of turns by one speaker to a particular set of recipients.

With this background understood, it is possible to understand that the single-line file containing:

```
************************** ok ***************************
```

scores as most dis-similar to a file of 183 lines, with this representative start:

```
i live
```

This approach allows us to explore the relationship between absolute similarity among files and appropriateness in their category (speaker) and also to explore the similarity relationships between categories of speakers partitioned according to who interacted directly or not.

Suppose a speaker has 20 files. It is an open question whether each of the files in that 20 will be most like the other 19 produced by that speaker or more like the other files derived from the log. Further, one wants to know how well the file fits with the sets of files produced by other speakers (categories). In fact, it might fit with a number of speakers' files and to a relatively high degree of significance.

A speaker whose files are most similar to the rest of the same speaker's files is a self-homogeneous speaker. A speaker can also be homogeneous with respect to other speakers' files. The homogeneity of a speaker with respect to another speaker can be measured by the (relativized) number of files produced by the speaker which score as most similar for the category. For a given speaker, A, we calculate:

1. The similarity set: the set of other speakers whose files are reliably ($p > 0.05$) similar to individual files produced by A.
2. The contact set: all of the speakers that spoke to A and all the speakers that A spoke to.

In order to examine further the interrelationship between patterns of interaction and levels of similarity we also adopted the notion of a pivot. A pivot is a speaker who is has a common relationship to at least two other speakers and therefore can 'represent' them. A pivot set is a smallest set which represents all of the speakers. For example, the audience pivot set is the smallest set of recipients that everyone has sent at least one turn to at least one of. Here we distinguish the contact pivot set and the similarity pivot set.

The pivot set can be understood as a contrast with Gärdenfors' analyses of meaning-determining groups (Gardenfors, 1993). A filter, defined on sets of sets (here, the basic entities within the sets are individuals), is a construct smaller than the power set of the set of basic individuals. The entire set of individuals is a member of a filter, but the empty set is not. For any two sets of individuals that are elements of a filter, their intersection is also a member. Further, a filter is monotonically increasing – if there is a set of individuals in the filter, then every containing superset is an element as well. This is a useful construct for explicating various social structures for meaning. Distinctions are available through distinct subsets of individuals. If there is exactly one individual that is common to all sets of individuals in the filter, then that individual can be seen as a 'dictator' of meaning, (Gardenfors, 1993). In thinking of a pivot set, one is considering a set that characterizes a set of sets that is not necessarily a filter – thus, no unique determiner of meaning, and potentially no shared meaning. Thinking of a signature set of sets based on a set of individuals, with a monotonically increasing closure, a pivot set is the smallest set of individuals required to ensure that every set is represented by one individual. A dictator would correspond to a singleton pivot set, the entire set of individuals would constitute a pivot set just if none of the sets of subsets had any elements in common (Babel).

In our analysis, the pivot sets can be treated in terms of contact or by similarity – sets of individuals who communicated directly with each other or sets of individuals comprising similarity equivalence classes.

The *contact pivot set* is just the audience pivot set. This is intended to capture the degree of interconnectedness within the community. If the contact pivot set is large there is a relatively 'dispersed' network of interconnections between residents, if it is small there are a number of 'gatekeeper' or 'funnel' individuals who provide contact points between different, relatively isolated, groups.

The *similarity pivot* set is the smallest set of speakers who are reliably similar to all the other speakers. In effect they represent the degree of differentiation of sub-group 'dialects' in the sample. If the similarity set is large there is relatively little convergence in dialects amongst the residents if it is small there are correspondingly fewer distinguishable 'sub-languages'.

The *non-pivot* set is the speakers who are neither contact pivots nor similarity pivots.

It is worth noting that the similarity based pivot set and the contact based pivot set are logically independent. A population with a relatively dispersed network of interactions could, nonetheless, have a relatively homogenous dialect. Conversely a highly centralised population might nonetheless sustain multiple dialects.

4 Results

The results reported here are based on the first 25 percent of the data set, and consists of the turns of 39 different residents of Walford.[2] This resulted in 547 files, with an average of 14.03 files per speaker.

The first question concerns the degree of overlap between the similarity set and the contact set. Of the 39, 25 had spoken to someone they were similar to, 14 of the 39 did not speak to anyone they were similar to. In the receiving direction, 23 speakers had at least one of their similarity set who had spoken to them and 16 had none of their similarity set among the people who spoke with them.

The second question concerns the pivots. In the sample of 39 speakers there are 4 similarity pivots and 27 contact pivots. However, in part because the data analyzed was truncated as the first 25% of the overall three-day log, some of the contact pivots are not present as actual speakers. Thus, the set of contact pivots who were also speakers (and thus provided text that can be measured for similarity, see below) contains seven individuals.

Table 2: Average Self-similarity in Pivot Groups

	N	Self-Homogeneity	Files
Nonpivot	28	0.03	7.36
S-pivot	4	0.15	45.43
C-pivot	7	0.07	5.75

Table 2 shows the average levels of self-homogeneity amongst speakers in the different pivot groups. The Similarity pivots have the highest level of self similarity, the Contact pivots moderate and the Non pivots lowest.

5 Discussion

Prima facie, the results provide evidence that local dialogue mechanisms such as interactive alignment (Pickering and Garrod, 2004), grounding (Clark and Wilkes-Gibbs, 1986) or local repair and clarification (Healey and Mills, 2006; Healey et al., 2007) do not account for the patterns of similarity in language use, as measured by letter unigrams, observed in Walford. If the mechanisms of dialect co-ordination were primarily local then the main locus of influence should be the contact set. However, the results show that residents interact with relatively high proportion of 'disimilar' people. This is indicative that convergence is not primarily mediated by direct contact.

Moreover, it appears that the pattern of interconnections amongst residents or 'network topology' is also a poor predictor of the pattern of sub-language convergence. Although there are a relatively high number of contact pivots (27) – indicating that the network is relatively diffuse or fragmented – there are a relatively low number of similarity pivots (4) indicating a small set of (statistically) distinguishable 'dialects'.

[2] The three day sample involves comparison of approximately 4,500 files with each other, which yields a space to reason about similarity with about 10 million elements. The combinatorial problem is large but not insoluble. The second author is investigating this complexity problem.

More importantly, in the current data set no individuals were both Similarity pivots and Contact pivots. This is consistent with an 'influential individual' explanation of the emergence of sub-group languages (Garrod and Anderson, 1987). Particular individuals who contribute a high number of turns (but who are not particularly well interconnected with other residents) have a disproportionate influence on the patterns of language use in the group but this influence appears to be mediated indirectly.

It is also interesting that in terms of self-similarity the least homogeneous speakers were the non-pivots. By definition these speakers interacted with fewer people and were least similar to the others. It appears that being, in effect, peripheral nodes in the network correlates with less consistent language use. The speakers with the highest level of self-similarity were the Similarity pivots.

Considered together this analysis suggests that the factors which promote sub-language convergence operate through indirect patterns of influence over successive exchanges rather than through local patterns of influence within interactions.

Overall, this is generally consistent with a version of Putnam's linguistic division of labour (Putnam, 1975) explanation of co-ordination of meaning in which control of language use is effectively deferred to key individuals in a community. In Walford it is unclear whether this is due to sheer persistence and volume of communication or whether, as in Putnam's conjecture, it is a consequence of differences in expertise or perhaps esteem.

A key challenge in this analysis has been to develop techniques that can analyse large networks of communal interaction. Two problems arise, first we want to look at a much smaller grain size than is typical for corpus analysis; turns and groups of turns rather than extended texts. In addition, a clear implication of this work is that we must pay close attention to the pattern of possible direct and *indirect* inter-relationships in a community. This creates a formidable computational problem.

The uni-gram technique has the advantage that it avoids problematic judgements about the form or content or intended force of each contribution. However, it simultaneously raises questions about what is really being measured. It's main virtue for our purposes is as a crude but robust index of similarity. Future work will need to explore how it correlates with other linguistic and pragmatic structures.

Acknowledgements

We are grateful to Graham White for his work on creating the logs analysed in this paper and the residents of Walford for agreeing to participate in this research. Science Foundation Ireland (RFP 05/RF/CMS002) has supported this work.

References

J. R. Bergmann and T. Luckmann. 1994. Reconstructive genres of everyday communication. In U. Quasthoff, editor, *Aspects of Oral Communications*, pages 289–304. Berlin: Mouton de Gruyter.

Carole Chaski. 1999. Linguistic authentication and reliability. In *Proceedings of National Conference on Science and the Law*, pages 97–148.

H. H. Clark and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1–39.

Herbert H. Clark. 1996. Communities, commonalities, and communication. In John J. Gumperz and Stephen C. Levinson, editors, *Rethinking linguistic relativity*, pages 324–355. Cambridge: Cambridge University Press.

Herbert H. Clark. 1998. Communal lexicons. In K. Malmkjoer and J. Williams, editors, *Context in language learning and language understanding*, pages 63–87. Cambridge: CUP, 3rd edition.

R. Dunbar. 1993. Coevolution of neocortex size, group size and language in humans. *Behavioural and Brain Sciences*, 16:681–735.

P. Gardenfors. 1993. The emergence of meaning. *Linguistics and Philosophy*, 16:285–309.

Simon C. Garrod and Anthony Anderson. 1987. Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, 27:181–218.

Simon C. Garrod and Gwyneth Doherty. 1994. Conversation, coordination and convention: an empirical investigation of how groups establish linguistic conventions. *Cognition*, 53:181–215.

John J. Gumperz. 1982. Conversational code switching. In John J. Gumperz, editor, *Discourse Strategies*, pages 59–99. Cambridge: Cambridge University Press.

John J. Gumperz. 1996. The linguistic and cultural relativity of conversational inference. In John J. Gumperz and Stephen C. Levinson, editors, *Rethinking linguistic relativity*, pages 374–406. Cambridge: Cambridge University Press.

P.G.T. Healey and G. Mills. 2006. Participation, precedence and co-ordination in dialogue. In R. Sun and N. Miyake, editors, *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, pages 1470–1475.

P.G.T. Healey, N. Swoboda, I. Umata, and J. King. 2007. Graphical language games: Interactional constraints on representational form. *Cognitive Science*, 31:285–309.

P.G.T. Healey. 1997. Expertise or expert-*ese*?: The emergence of task-oriented sub-languages. In M.G. Shafto and P. Langley, editors, *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 301–306.

A. Kilgarriff and R. Salkie. 1996. Corpus similarity and homogeneity via word frequency. In *Proceedings of Euralex 96*.

Adam Kilgarriff. 2001. Comparing corpora. *International Journal of Corpus Linguistics*, 6(1):97–133.

M. Pickering and S. Garrod. 2004. The interactive alignment model. *Behavioral and Brain Sciences*, 27(2):169–189.

T. Postmes, R. Spears, and M. Lea. 2000. The formation of group norms in computer-mediated communication. *Human Communication Research*, 26(3):341–371.

Hilary Putnam. 1975. The meaning of meaning. In K. Gunderson, editor, *Language, Mind, and Knowledge, Minnesota Studies in the Philosophy of Science, 7*. Minneapolis: University of Minnesota Press.

M. Robinson and Liam Bannon. 1991. Questioning representations. In L. Bannon, M. Robinson, and K. Schmidt, editors, *Proceedings of the Second European Conference on CSCW*, pages 219–233. Dordrecht: Kluwer.

Mildred L. G. Shaw and Brian R. Gaines. 1988. A methodology for recognising consensus, correspondence, conflict and contrast in a knowledge acquisition system. In *Third Workshop on Knowledge Acquisition for Knowledge-Based Systems*. Banff.

Sofie Van Gijsel and Carl Vogel. 2003. Inducing a cline from corpora of political manifestos. In Markus Aleksy et al., editor, *Proceedings of the International Symposium on Information and Communication Technologies*, pages 304–310.

Carl Vogel and Sandrine Brisset. 2006. Hearing voices in the poetry of brendan kennelly. In *Varieties of Voice*. 3rd international BAAHE conference. Leuven, 7-9 December 2006.

Carl Vogel. 2007. N-gram distributions in texts as proxy for textual fingerprints. In Anna Esposito, Eric Keller, M. Marinaro, and Maja Brat anic, editors, *The Fundamentals of Verbal and Non-Verbal Communication and the Biometrical Issue*. IOS Press.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 149–154.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods

Gregory Aist[1] (gregory.aist@asu.edu), James Allen[2] (james@cs.rochester.edu), Ellen Campana[2,3,4,5] (ecampana@bcs.rochester.edu), Carlos Gomez Gallo[2] (cgomez@cs.rochester.edu), Scott Stoness[2] (stoness@cs.rochester.edu), Mary Swift[2] (swift@cs.rochester.edu), and Michael K. Tanenhaus[3] (mtan@bcs.rochester.edu)

[1]Computer Science
and Engineering
Arizona State
Tempe AZ USA

[2]Computer Science
University
of Rochester
Rochester NY USA

[3]Brain and
Cognitive Sciences
University of Rochester
Rochester NY USA

[4]Arts, Media,
and Engineering
Arizona State
Tempe AZ USA

[5]Department of
Psychology
Arizona State
Tempe AZ
USA

Abstract

Current dialogue systems generally operate in a pipelined, modular fashion on one complete utterance at a time. Converging evidence shows that human understanding operates incrementally and makes use of multiple sources of information during the parsing process, including traditionally "later" aspects such as pragmatics. We describe a spoken dialogue system that understands language incrementally, gives visual feedback on possible referents during the course of the user's utterance, and allows for overlapping speech and actions. We present findings from an empirical study showing that the resulting dialogue system is faster overall than its nonincremental counterpart. Furthermore, the incremental system is preferred to its counterpart – beyond what is accounted for by factors such as speed and accuracy. These results are the first to indicate, from a controlled user study, that successful incremental understanding systems will improve both performance and usability.

1 Introduction

The standard model of natural language understanding for dialogue systems is pipelined, modular, and operates on complete utterances. By pipelined we mean that only one level of processing operates at a time, in a sequential manner. By modular, we mean that each level of processing depends only on the previous level. By complete utterances we mean that the system operates on one sentence at a time.

There is, however, converging evidence that human language processing is neither pipelined nor modular nor whole-utterance. Evidence is converging from a variety of sources, including particularly actions taken while speech arrives. For example, natural turn-taking behavior such as backchanneling (uh-huh) and interruption occur while the speaker is still speaking. Evidence from psycholinguistics also shows incremental language understanding in humans (Tanenhaus et al. 1995, Traxler et al. 1997, Altmann and Kamide 1999) as evidenced by eye movements during language comprehension.

Many different sources of knowledge are available for use in understanding. On the speech recognition side, commonly used sources of information include acoustics, phonetics and phonemics, lexical probability, and word order. In dialogue systems, additional sources of information often include syntax and semantics (both general and domain-specific.) There are also however some sources of information that are less frequently programmed. These include such linguistics as morphology and prosody. Knowledge-based features are also available, such as world knowledge (triangles have three sides), domain knowledge (here there are two sizes of triangles), and task knowledge (the next step is to click on a small triangle. And, there is also pragmatic information available from the visual context (there is a small triangle near the flag.)

Here we discuss some of the progress we have made towards building methods for incremental understanding of spoken language by machines, which incorporates pragmatic information at the early stages of the understanding process. We also present a controlled experimental evaluation of our incremental system vs. its nonincremental counterpart.

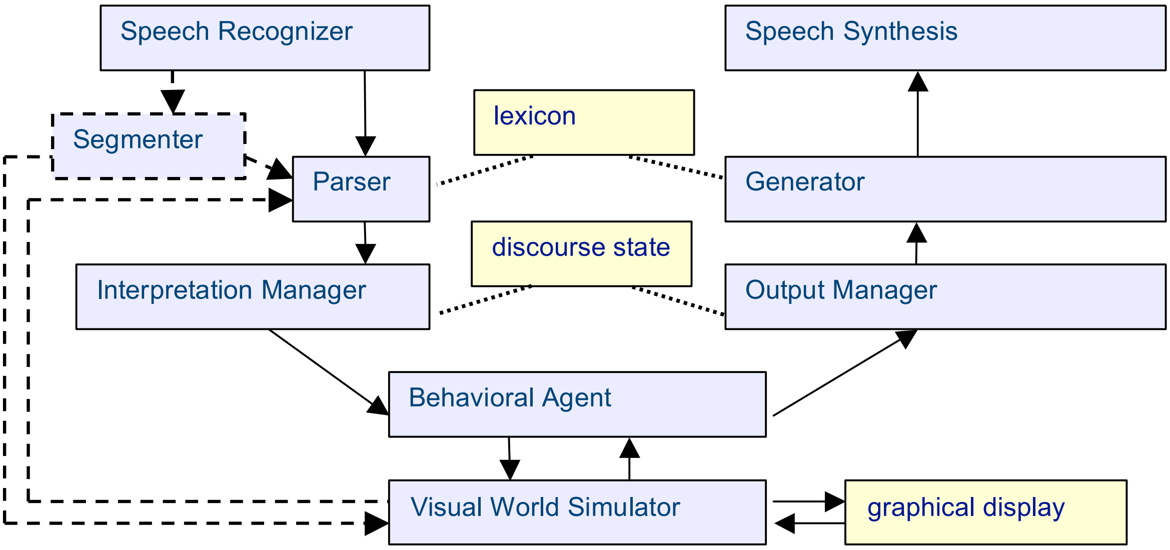


Figure 1. Changes to spoken dialogue system architecture to allow incremental understanding. Boxes show components; lines show message flow. In both types of systems, the lexicon and the discourse state are resources shared by input and output. Components and connections new to the incremental system are shown in **dashed lines**. Incremental understanding also places requirements on the speech recognizer (production of partial hypotheses), the parser (incremental construction of charts), the interpretation manager and behavioral agent (handling partial interpretations and actions), and the visual world simulator (incorporation of semantic models of partial actions) which are also important to the overall functioning of the system. This paper focuses on incremental *understanding* and thus the changes are to the *understanding* aspects of the dialogue system, including action-taking as representing direct evidence of understanding.

2 Traditional vs. Incremental Systems

Figure 1 shows a diagram of a traditional dialogue system architecture, with additional components and connections to added to support incremental understanding. Incremental language processing as we conceive it involves a number of fundamental and inter-related changes to the way in which processing occurs:

(a) input sentences are processed before the user turn ends, as opposed to processing only when turn is finished;

(b) components of the architecture operate asynchronously with several operating simultaneously, in contrast to a serial one where only one module at a time can be active;

(c) knowledge resources are available to several components at the same time, in contrast to a "pipeline" architecture where knowledge is sent from module to module;

(d) there is overlapping speech and graphical output ("action"), in contrast to presenting speech and other output sequentially;

(e) system and user turns and actions can overlap as appropriate for the dialogue.

We discuss some of these distinctions in more detail.

In a traditional dialogue system architecture, each component processes input from other components one utterance at a time. In our incremental architecture, each component receives input from other components as available, on a word-by-word basis.

In a traditional system, each component feeds forward into other components. In our incremental architecture, each component advises other components as needed – and advice can flow both "forward" in the traditional directions and "backward" from traditionally later stages of processing (such as pragmatics) to traditionally earlier stages of processing (such as parsing.) In a traditional system, the internal representations assume a strict division of time according to what's happening – the system is speaking, or the user is speaking, or the system is acting, and so forth. In our incremental architecture, representations can accommodate multiple events happening at once – such as the system acting while the user is still speaking.

In addition to these overall changes, our system incorporates a number of specific changes.

1. A Segmenter operates on incoming words, identifies pragmatically relevant fragments, and announces them to other system components such as the parser and the visual world simulator.

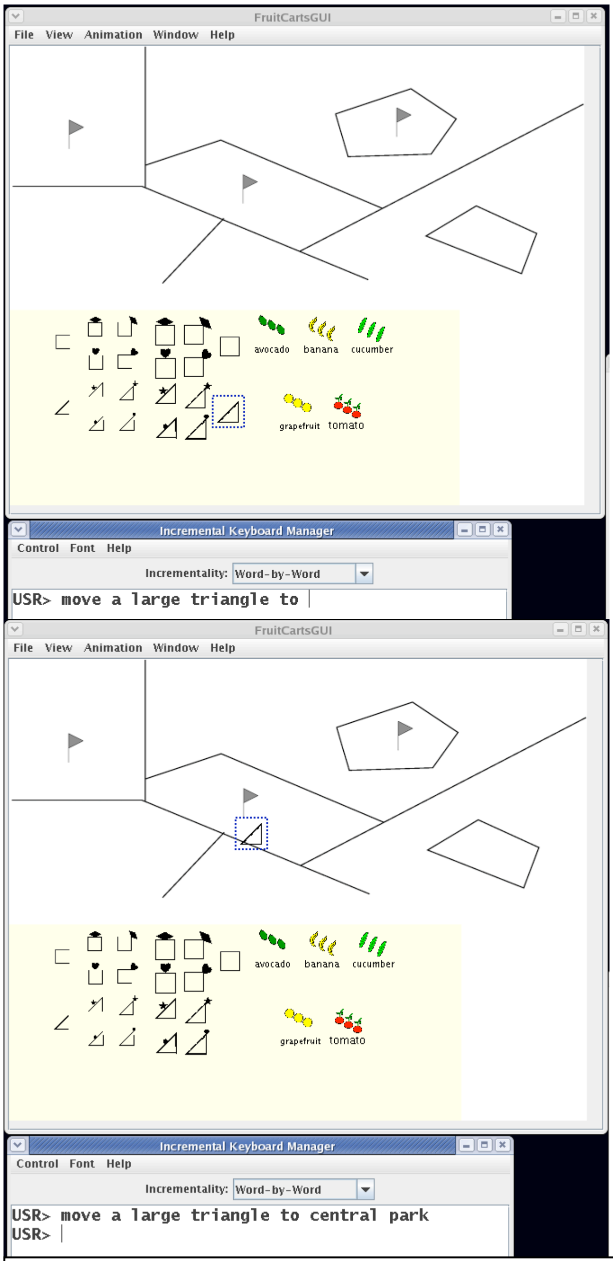
1 okay so
2 we're going to put a large triangle with nothing into morningside
3 we're going to make it blue
4 and rotate it to the left forty five degrees
5 take one tomato and put it in the center of that triangle
6 take two avocados and put it in the bottom of that triangle
7 and move that entire set a little bit to the left and down
8 mmkay
9 now take a small square with a heart on the corner
10 put it onto the flag area in central park
11 rotate it a little more than forty five degrees to the left
12 now make it brown
13 and put a tomato in the center of it
14 yeah that's good
15 and we'll take a square with a diamond on the corner
16 small
17 put it in oceanview terrace
18 rotate it to the right forty five degrees
19 make it orange
20 take two grapefruit and put them inside that square
21 now take a triangle with the star in the center
22 small
23 put it in oceanview just to the left of oceanview terrace
24 and rotate it left ninety degrees
25 okay
26 and put two cucumbers in that triangle
27 and make the color of the triangle purple

Figure 2. Example human-human dialogue in the fruit carts domain.

2. Pragmatic information is provided to the parser in order to assist with ongoing parses.
3. Modeling of actions and events is done by means of incremental semantics, in order to properly represent partial actions and allow for overlapping actions and speech.
4. Visual feedback is provided to the user about possible referents while the user is speaking.

3 Testbed Domain: Fruit Carts

To explore the effects of incremental understanding in human-computer dialogue, we devised a testbed domain (Figures 2, 3) where a person gives spoken instructions to a computer in order to reproduce a goal map. On the map, there are named regions, some of which contain flags as landmarks; the screen also has two kinds of objects: abstract shapes such as triangles and squares, and "fruit" of various kinds (avocados, bananas, cucumbers, grapefruits, and tomatoes.) In this domain, certain steps were taken in order to reduce complexity and increase the predictability of the spoken language. In particular, all objects and names of regions were chosen to be easy to name (or read) and easy for the speech recognizer to hear. In order to facilitate the study of incremental understanding of natural language by machines, the Fruit Carts domain contains various points of disambiguation based on factors including object size, color, shape, and decoration; presence or absence of a landmark; and phonetic similarity of geographically close regions of the map (e.g. "Morningside" and "Morningside Heights" are close together.) For example, a square with stripes could also be referred to as "the stripey square", but a square with diamonds on the corner cannot be referred to as *"the corner-diamonded square". We thus chose a set of shapes such as "a small square with a diamond on the edge", "a large triangle with a star on the corner", "a small triangle with a circle on the edge", and so forth. Human-



Incrementality: Word-by-Word

```
USR> move a large triangle to
```

Incrementality: Word-by-Word

```
USR> move a large triangle to central park
USR>
```

Figure 3. An example interaction with the incremental dialogue system. Note that in the top screenshot, halfway through the sentence, the large triangle is already highlighted. This figure shows typed input for clarity; the experiments used spoken input.

human dialogue collected in this domain was used in the construction of the dialogue system.

We collected a set of dialogs from human-human conversation in this domain. Our observations included the following:

1. End-of-sentence boundaries tend to be fairly clear (at least to a human listener). Where a sentence begins, however, is quite difficult to say precisely, due to disfluencies, abandoned utterances, and so forth. This is in contrast to domains where speakers might tend to begin a sentence clearly, such as information retrieval ("Search for books by Kurt Vonnegut").

2. There seem to be two distinct strategies that people can employ: saying a direction all at once ("Put it one inch below the flag") or continuously ("Put it near the flag [pause] but down a bit [pause] a bit more [pause] stop.")

3. Besides a pure All-at-once and Continuous strategy, people sometimes switch between them, employing Both. For example, the director might tell the actor to place an object "right on the flag [pause] down a bit [pause] keep going [pause] stop." We see these as possibilities along a continuum, using the same language mechanisms yet according different emphasis to the strategies.

Our previous findings about these types of language include that continuous-style language uses fewer words per utterance than all-at-once language, and the words themselves are shorter in length as well (reference omitted for review). Furthermore, the use of continuous language increases over the course of the dialogs. Specifically, the relative percentage of continuous language increases over trials. The relative increase in continuous language over time is statistically significant (by logistic regression; style as outcome, subject as categorical, trial as numeric. $B=0.104 \pm 0.037$, $\exp(B) \approx 1.11$, $p < 0.01$). So not only do people engage in dialogue that relies on incremental understanding on the part of the hearer, but such interactions actually becomes more important as the dialogue progresses.

We used these human-human conversations to form the basis for formalizing various aspects of continuous understanding, and for gauging the behavior of the spoken dialog system that we built to operate in this testbed domain. The resulting system is capable of interactions as shown in Figure 3, where the user's utterance is processed as it is received, visual feedback is provided during the course of the utterance, and speech and actions can overlap. As in the human-human interactions, moving an object from one location to another takes time in the working system – that is, the objects are shown moving in a straight line from the beginning point (e.g. the bin at the bottom of the screen) to the end point (the flag in central park.)

4 Related Work

We have previously shown that incremental parsing can be faster and more accurate than non-incremental parsing (references omitted for review.) In addition, we have shown that in this domain the relative percentage of language that is of a more interactive style also increases over time (references omitted.) A number of research efforts have been directed at incremental understanding, adopting a wide variety of techniques including the blackboard architecture, finite state machines (Ait-Mokhtar and Chanod 1997), perceptrons (Collins and Roark 2004), neural networks (Jain and Waibel 1990), categorial grammar (Milward 1992), tree-adjoining grammar (Poller 1994), and chart parsing (Wiren 1989). We compare our work to several such efforts.

Higashinaka et al. (2002) performed a linear regression experiment to find a set of features that predict performance of systems that understand utterances incrementally. The system evaluated by the authors is incremental in that dialogue states are updated as the sentence is processed. However this is a result of incrementally processing the input stream and not the type of continuous understanding we propose. In our approach we allow the parser to make use of information from different layers of processing (i.e. pragmatic constraints from verb-argument constructions, real world knowledge, etc).

Rosé et al. (2002) describe a reworking of a chart parser so that "as the text is progressively revised, only minimal changes are made to the chart". They found that incrementally parsing incoming text allows for the parsing time to be folded into the time it takes to type, which can be substantial especially for longer user responses. Our current work operates on spoken input as well as typed input and makes extensive use of the visual context and of pragmatic constraints during parsing.

DeVault and Stone (2003) describe techniques for incremental interpretation that involve annotating edges in a parser's chart with constraints of various types that must be met for to the edge to be valid. That has a clean and nice simplicity to it, but seems to impose uniformity on the sorts of information and reasoning that can be applied to parsing. In our approach, advice to the parser is represented as modifications to the chart, and can thus be in any framework best for the source.

Work by Schuler (2001 and following) has moved away from a pipeline architecture by accessing different sources of knowledge while parsing the sentence. Using real world knowledge about objects improves parsing and can only be achieved by analyzing the sentence from the start. Schuler makes use of potential referents from the environment much the same way that we have also done by the use of model-theoretic interpretations. Thus the system evaluates the logical expressions for all possible potential referents at each node of the tree to know whether they are possible in the current domain. The author provides an example where a PP attachment ambiguity is resolved by knowing a particular fact about the world which rules out one of the two possible attachments. Thus this sort of knowledge comes into play during parsing. Even though the system described in the present paper shares the same goals in using more than just syntactic knowledge for parsing, our parser feedback framework does not require the rewriting of the grammar used for parsing to incorporate environment knowledge. This approach based on probability feedback directly affecting the parser chart is simpler and thus more applicable to and easily incorporated in a wider range of parsers and grammars.

5 Evaluation

We conducted a controlled evaluation comparing incremental understanding to its nonincremental counterpart in our testbed domain. In the nonincremental system, speech and actions alternate; in the incremental system, the actions and speech overlap.

A total of 22 dialogues were collected, each of which consisted of two utterances and the corresponding system responses. Eleven of the dialogues were in the control (nonincremental) condition and eleven of the dialogues were in the experimental (incremental) condition. The utterances were in-domain and understandable by both the nonincremental and incremental versions of the system, they were pre-recorded; and the same utterances were played to each version of the system; this technique allowed us to minimize variance due to extraneous factors such as interspeaker variability, acoustic noise, and so forth, and concentrate specifically on the difference between incremental processing and its nonincremental counterpart. The resulting dialogues were recorded on digital video.

The incremental system was approximately 20% faster than the nonincremental system in terms of time to task completion for each two-utterance dialogue, at 44 seconds per dialogue vs. 52 seconds for the control condition (single-factor ANOVA, F=10.72, df=21, p-value 0.004).

To further evaluate the effectiveness of the incremental system, we conducted an onlooker study where 18 subjects, mostly from the University of Rochester community, rated the interactions in the dialogues. First, each subject watched one video clip once and only once; then, the subject filled out written responses to questions about that video clip. Subjects provided responses for each dialogue video clip to four Likert-scaled (1-7, 1=less) questions on speed, accuracy, match-to-intent, and satisfaction:

[FAST] "How fast did the computer respond?"

[ACC] "How accurately did the system understand?"

[ACT] "How well matched were the computer's actions to what the person wanted?"

[SAT] "If you had been the person giving the commands, how satisfied overall would you be with the interaction?"

In order to check that people's responses were objectively correlated with actual system performance, four "wrong" system videos were included in the study, two for each condition (nonincremental control and incremental / experimental condition). That is, the user in the video said one thing, but the system did something else. In this way, we experimentally manipulated the "right/wrong" response of the system to see how people would rate the system's correctness.

We measured speed, accuracy, and match to user intentions with a subjective survey; as it happens, our results are compatible with methods that measure these factors objectively and then relate them to subjectively reported user satisfaction. For example, the PARADISE model (Walker et al. 1997) found that speed, accuracy, and match to user intentions well predicted user satisfaction. Using a linear regression model as in the original PARADISE framework, we confirmed that with our data a linear model with speed (FAST), accuracy (ACC), and match-to-actions (ACT) as input variables predicts well the output variable satisfaction (SAT) (R=.795, R Square=.631, Adj. R Square=.625; df=3, F=91.389, p<0.001).

Since the input and output variables are seven-item Likert scale responses it turns out that ordinal regression models are a better match to the experimental setup than the linear regression models. Ordinal regression models are specifically designed for cases where the variables are a set of levels that are ordered (N+1>N) but not necessarily linear (1 to 2 may not be the same as 4 to 5.) We thus adopted ordinal regression models for the remainder of the analyses. In addition, since some of the subjects indicated in written comments that they got used to the behavior of the system over time, we included the dialogue number (NTH; 1=first seen, 22=last seen) as a covariate. And, since individual subjects tend to vary in their responses (some subjects more negative than other subjects), we also included subject (SUBJ) as an input variable.

The model we built to analyze the effects of right/wrong system response (RIGHT) and nonincremental vs. incremental processing (INC) was as follows. We built an ordinal regression model predicting satisfaction (SAT) by right/wrong (RIGHT) and nonincremental/incremental (INC) and subject (SUBJ) with FAST, ACC, and ACT as covariates (Table 1).

The first result we found was that there was a significant effect for RIGHT as a predictor of user satisfaction, in the expected direction: wrong responses predict lower satisfaction (or, equivalently, correct responses predict higher satisfaction.) These results help validate the external reliability of the experimental design.

Next, to evaluate the effects of incremental vs. nonincremental processing, we examined the model coefficient for INC. In this case, nonincremental processing was a significant predictor of lower satisfaction (p=.026) – or, equivalently, incremental processing was a significant predictor of higher satisfaction.

6 Conclusion

Our results show that – at least for this task – incremental processing predicts higher user satisfaction. Why? The statistical model makes clear that this preference is the case after controlling for factors such as speed, accuracy, and match-to-intent. Explanatory factors that remain include naturalness – that is, the ways in which incremental systems are more like human-human conversation than their nonincremental counterparts. Nonincremental dialogue systems require many artificial restrictions on what the user and the system can say and when they can say it, and therefore exclude many important characteristics of natural human dialogue. Incremental understanding has the potential to remove such obstacles. The work presented here suggests that successful incremental understanding systems will improve both performance and usability

Table 1. Parameters of ordinal regression model predicting satisfaction (SAT).

Variable	Estimate	Std. Error	Sig.
NTH	.188	.058	.001
FAST	.770	.176	.000
ACC	1.411	.341	.000
ACT	.616	.304	.043
RIGHT=0 (0=wrong, 1=right.)	-1.855	.903	.040
INC=0 (0=control 1=incr.)	-2.336	1.051	.026

References

Ait-Mokhtar, S. and Chanod, J.-P. Incremental finite-state parsing. ANLP 1997.

Altmann, G. and Kamide, Y. Incremental interpretation at verbs: restricting the domain of subsequent reference. *Cognition* 73(3):247-264. 1999.

Collins, M. and B. Roark. Incremental parsing with the perceptron algorithm. ACL 2004.

DeVault, D. and Stone, M. Domain inference in incremental interpretation. ICOS 2003.

Higashinaka, R., Miyazaki N., Nakano, M., & Kiyoaki, A. A method for evaluating incremental utterance understanding in spoken dialogue systems. ICSLP 2002.

Jain, A. & Waibel, A. Incremental parsing by modular recurrent connectionist networks. NIPS 1990.

Milward, D. Dynamics, dependency grammar and incremental interpretation. COLING 1992.

Poller, P. Incremental parsing with LD/TLP-TAGS. Computational Intelligence 10(4). 1994.

Rosé, C.P., Roque, A., Bhembe, D., and Van Lehn, K. An efficient incremental architecture for robust interpretation. HLT 2002.

Schuler, W. Computational properties of environment-based disambiguation. ACL 2001.

Tanenhaus, M.K., Spivey-Knowlton, M.J., Eberhard, K.M., & Sedivy, J.C. Integration of visual and linguistic information in spoken language comprehension. *Science*, Vol. 268 (5217), 1632-1634. 1995.

Traxler, M.J., Bybee, M.D., & Pickering, M.J. Influence of Connectives on Language Comprehension: Eye-tracking Evidence for Incremental Interpretation. The Quarterly Journal of Experimental Psychology: Section A, 50(3), 481-497. 1997.

Wiren, M. Interactive incremental chart parsing. In Proceedings of the 4th Meeting of the European Chapter of the Association for Computational Linguistics. 1989.

Walker, M., Litman, D., Kamm C., and Abella, A. PARADISE: A Framework for Evaluating Spoken Dialogue Agents. ACL 1997.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 155–156.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

A Game-Based Strategy for Optimizing Agents' Argumentation in Deliberation Dialogues

Gemma Bel-Enguix and **M. Dolores Jiménez-López**
Research Group on Mathematical Linguistics
Universitat Rovira i Virgili
Pl. Imperial Tarraco, 1
43005 Tarragona, Spain
gemma.bel,mariadolores.jimenez@urv.cat

Dialogue can be understood as a pragmatic entity where the participants try to maximize the possibilities of success in their argumentation.

Reed and Long (1997) make an interesting distinction between cooperation and collaboration. For a dialogue to be brought about, cooperation is necessary, but collaboration not always exists.

For us, a crucial and non-static element in dialogue is context, understood as the environmental and personal states and circumstances that can affect the development of the dialogue. This context is in constant evolution, not only because of external factors, but also because of the speech acts of participants. Therefore, like Bunt (1994), we think that the configuration of the dialogue is directly related to the intentions of the speakers/hearers and to the context.

In what refers to the types of dialogues according to human argumentation, Walton and Krabbe (1995) introduced a taxonomy that has become classical. They distinguish between *information seeking*, *inquiry*, *persuasion*, *negotiation*, *deliberation* and *eristic* dialogues. Our work is mainly focused in *deliberation*, a kind of dialogue in which participants have to reach an agreement and make a decision.

We approach deliberation from the perspective of dialogue games (Carlson, 1983) with two participants. We use the extensive form of games representation because we assume the participation of the speakers is sequential and they alternate in turn. In this research, we are mainly interested in defining games where the participants in the deliberation have secret intentions. In the sequel, the term dialogue refers to "deliberation dialogue".

The first step for describing deliberation is to define two participants, A_1 and A_2. Each one has a set of dialogue acts $\Theta(A_1)$, $\Theta(A_2)$, which are subsets of the acts store $\Theta = \{p, r, s, a, q, x\}$. Such store is an intentionally limited one, where p and s are two different types of arguments, r is a counter-argument rejection, a is acceptance, q is a question and x represents that an agent is quitting the dialogue. We also establish that r and a cannot be initial productions of the dialogue because they are only valid as a counter-argument.

$\mathcal{R}$ is a set of combinations of argumentation-counterargumentation that relates elements from $\Theta(A_1)$ to acts belonging to $\Theta(A_2)$. These rules have the form $p \rightarrow q$. Every agent has its own set of rules, R_1 for A_2 and R_2 for A_2. If single elements are found in the sets of rules of the agents, they can be used only as starting productions. They are, then, the starting symbols of the system. By definition, the participant that starts the dialogue is A_1, if it has at least one starting symbol. Therefore, if both agents have starting acts, only A_1 will be able of using them.

We denote a production w of an agent A_n in a given state as $A_n(w)$, and the set of possible productions for an agent A_n in a given state as $\theta(A_1)$.

The possible outcomes of the deliberation are represented with upper-case roman letters. They belong to the set O, such that $O = \{A, B, ..., Z\}$. Some of the elements of Θ are associated to elements of O by an application $\mathcal{F}$. Such elements are named terminal acts.

Keeping in mind the parameters explained above, a definition of deliberation games can be introduced:

Definition 1 *Having two speakers A_1 and A_2, a deliberation game G between them is defined as a 4-tuple:*

$$G = (\Theta, \mathcal{R}, O, \mathcal{F})$$

where:

- *Θ is an acts store;*
- *$\mathcal{R} = R_1 \cup R_2$ is the set of argumentation rules for each agent;*
- *O is the set of possible outcomes of the deliberation;*

- *$\mathcal{F}$ is an application relating elements of Θ to elements of O. Such application is denoted by the symbol '$\Rightarrow$'. If there is not an O element for a sign belonging to Θ, then the result is Ind, which means that the outcome is undecidable and the deliberation has to go on.*

As for the tree diagram, we introduce a distinction between *terminal nodes* and *final nodes*. Terminal refers to the nodes which cannot be developed any more, which corresponds to the classical definition of "terminal". However, final nodes are the last nodes generated after a given move. The nodes that, after the application of $\mathcal{F}$ are not labelled wit Ind are terminal. Nodes Ind are final but non terminal nodes. Tree-diagram will show all the possible productions of the game, where the nodes are the agents speaking and the edges denote dialogue acts.

A *trajectory of dialogue* is every lineal path of the tree starting in the initial node. A *complete trajectory* is every path from the starting utterance to a terminal symbol.

Being G a deliberation game, and $\Theta = \{w\}$ the acts store, we denote a trajectory n of this game in the form $G_n(w_1, w_2, ..., w_n)$, being $w_1, w_2, ..., w_n$ the utterances generated to reach the final agreement in order of generation. Since a dialogue has as many trajectories as final results, then we say that a $G = \{G_1, G_2, ..., G_n\}$. The width of a dialogue $width(G)$ is the maximal number of trajectories it has. The trajectories are ordered starting with the leftmost and finishing by the rightmost. We call *paired trajectories* those that have an even number of edges and unpaired *trajectories* those that have an odd number of edges.

We define a move M as an adjacency pair that consists of argument and counterargument. A sequence is a set of moves $M_m, M_n, ..., M_i$. A deliberation game can have one or more moves. As in real life, some dialogues stop after a number of productions that has been determined before, and other can compute after all possibilities have been explored. The productions generated after a move M_n are $\theta(M_n)$. In $\theta(M_n)$, two types of acts can be distinguished: non-terminal $nt(M_n)$ and terminal $t(M_n)$. The state of the dialogue after M_n, denoted $\Theta(M_n)$ includes $\theta(M_n)$ and all the terminal acts that have been achieved before M_n, denoted by $T(M_n)$. Being M_m, M_n the first and second moves in a deliberation, it is clear that in M_m, $\Theta(M_m) = \theta(M_m)$, while in M_n, $\Theta(M_n) = t(M_m) \cup \theta(M_n)$. Being $M = \{M_m, M_n, ..., M_i\}$, $\Theta(M_i) = t(M_m) \cup t(M_n) ... \cup ... \theta(M_i)$, or its equivalent $\Theta(M_i) = T(M_{i-1}) \cup \theta(M_i)$. If in a given move M_n, $\theta(M_n) = t(M_n)$, then the dialogue is complete.

The results of the productions in a move M_n are designed by $g(M_n)$, and they are obtained by applying $\mathcal{F}(\theta(M_n) \Rightarrow O)$. The possible agreements of the deliberation once the move M_n has been performed, are denoted by $G(M_n)$. They are obtained by applying $\mathcal{F}(\Theta(M_n) \Rightarrow O)$.

In this research, we assume agents have a clear order of preferences, even if they want to reach an agreement. In order to optimize the options to obtain a good deal, two very simple techniques can be carried out: *horizontal scoring* and *balance scoring*.

Horizontal scoring measures the potential index of success for each agent in a given move, if the final agreement is achieved in that move. It just calculates the average of the score for both agents in each move.

Balance scoring is a technique that calculates the possibilities of success for every one of the utterances that an agent can perform in every move. To do that, the sub-trees produced for every potential production are measured.

By means of this method we attempt to explore some mathematical properties of deliberation that can be applied to the design of strategies for the agents to achieve a good agreement. The participants in the dialogue have to calculate the convenience of having a large exchange as well as the index of success for every trajectory. The model assumes an evolution in the internal state of the agents, in the strategies of the participants and the environment where the conversation takes place.

References

Harry Bunt. 1994. Context and dialogue control. *Think*, 3:19–30.

Lauri Carlson. 1983. *Dialogue Games. An Approach to Discourse Analysis.* Reidel, Dordretch.

Chris Reed and Derek Long. 1997. Collaboration, cooperation and dialogue classification. In K. Jokinen, editor, *Working Notes of the IJCAI97 Workshop on Collaboration, Cooperation and Conflict in Dialogue Systems*, pages 73–78, Nagoya.

Douglas N. Walton and Eric C.W. Krabbe. 1995. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning.* SUNY Press, Albany, NY.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 157–158.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

The Fyntour Multilingual Weather and Sea Dialogue System

Eckhard Bick
University of Southern Denmark
Odense
echard.bick@mail.dk

Jens Ahlmann Hansen
University of Southern Denmark
Odense
ahlmann@voicetech.dk

1 Introduction

The Fyntour multilingual weather and sea dialogue system provides pervasive access to weather, wind and water conditions for domestic and international tourists who come to fish for seatrout along the coasts of the Danish island of Funen. Callers access information about high and low waters, wind direction etc. via spoken dialogues in Danish, English or German. We describe the solutions we have implemented to deal with number format data in a multi-language environment. We also show how the translation of free text 24-hour forecasts from Danish to English is handled through a newly developed machine translation system. In contrast with most current, statistically-based MT systems, we make use of a rule-based apporach, exploiting a full parser and context-senstitive lexical transfer rules, as well as target language generation and movement rules.

2 Number Format Data

The Fyntour system provides information in Danish, English and German. A substantial amount of data is received and handled in an interlingua format, i.e. data showing wind speed (in m/s) and precipitation (in mm) are language-neutral numbers which are simply converted into language-specific pronunciations by specifying the locale of the speech synthesis in the VoiceXML , e.g.

<prompt xml:lang="da-DK"> 1 </prompt> ”en”
<prompt xml:lang="de-DE"> 1 </prompt> ”ein”
<prompt xml:lang="en-GB"> 1 </prompt>
”one”

In Germany, wind speed is normally measured using the Beaufort scale (vs. the Danish m/s norm), while visitors from English speaking countries are accustomed to the 12-hour clock (vs. the continental European 24-hour clock). These cultural preferences can be catered for by straightforward conversions of the shared number format data – performed by the application logic generating the dynamic VXML output of the individual languages.

However, the translation of dynamic data in a free text format, from Danish to English and Danish to German, – such as the above-mentioned forecasts, written in Danish by different meteorologists – is more complex. In the Fyntour system, the Danish-English translation problem has been solved by a newly developed machine translation (MT) system. The Constraint Grammar based MT-system, which is rule-based as opposed to most existing, probabilistic systems, is introduced below.

3 CG-based MT System

The Danish-English MT module, Dan2eng, is a robust system with a broad-coverage lexicon and grammar, which in principle will translate unrestricted Danish text or transcribed speech without strict limitations to genre, topic or style. However, a small benchmark corpus of weather forecasts was used to tune the system to this domain and to avoid lexical or structural translation gaps, especially concerning time and measure expressions, as well as certain geographical references and names.

Methodologically, the system is rule-based rather than statistical and uses a lexical transfer approach with a strong emphasis on source language (SL) analysis, provided by a pre-existing Constraint Grammar (CG) parser for Danish, DanGram (Bick 2001). Contextual rules are used at 5 levels:

1. CG rules handling morphological disambiguation and the mapping of syntactic func-

tions for Danish (approximately 6.000 rules)

2. Dependency rules establishing syntactic-semantic links between words or multi-word expressions (220 rules)
3. Lexical transfer rules selecting translation equivalents depending on grammatical categories, dependencies and other structural context (16.540 rules)
4. Generation rules for inflexion, verb chains, compounding etc. (about 700 rules)
5. Syntactic movement rules turning Danish into English word order and handling sub-clauses, negations, questions etc. (65 rules)

At all levels, CG rules may be exploited to add or alter grammatical tags that will trigger or facilitate other types of rules.

As an example, let us have a look at the translation spectrum of the weatherwise tedious, but linguistically interesting, Danish verb *at regne (to rain),* which has many other, non-meteorological, meanings *(calculate, consider, expect, convert ...)* as well. Rather than ignoring such ambiguity and build a narrow weather forecast MT system or, on the other hand, strive to make an "AI" module *understand* these meanings in terms of world knowledge, Dan2eng chooses a pragmatic middle ground where grammatical tags and grammatical context are used as *differentiators* for possible translation equivalents, staying close to the (robust) SL analysis. Thus, the translation *rain (a)* is chosen if a daughter/dependent (D) exists with the function of situative/formal subject (@S-SUBJ), while most other meanings ask for a human subject. As a default[1] translation for the latter *calculate (f)* is chosen, but the presence of other dependents (objects or particles) may trigger other translations. *regne med (c-e),* for instance, will mean *include,* if *med* has been identified as an adverb, while the preposition *med* triggers the translations *count on* for human "granddaughter" dependents (GD = <H>), and *expect* otherwise. Note that the *include* translation also could have been conditioned by the presence of an object (D = @ACC), but would then have to be differentiated from (b), *regne for ('consider').*

regne_V[2]
(a) D=(@S-SUBJ) :rain;
(b) D=(<H> @ACC) D=("for" PRP)_nil :consider;
(c) D=("med" PRP)_on GD=(<H>) :count;
(d) D=("med" PRP)_nil :expect;
(e) D=(@ACC) D=("med" ADV)_nil :include;
(f) D=(<H> @SUBJ) D?=("på")_nil :calculate;

It must be stressed that the use of grammatical relations as translation differentiators is very different from a simple memory based approach, where chains of words are matched from parallel corpora. First, the latter approach - at least in its naïve, lexicon-free version - cannot generalize over semantic prototypes (e.g. <H> for human) or syntactic functions, conjuring up the problem of sparse data. Second, simple collocation, or co-occurrence, is much less robust than functional dependency relations that will allow interfering material such as modifiers or sub-clauses, as well as inflexional or lexical variation.

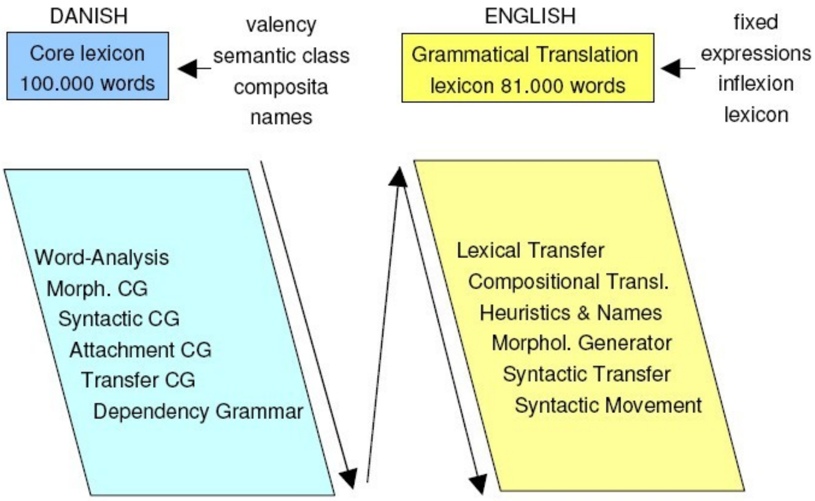


Fig 1: The Dan2eng system

For more details on the Dan2eng MT system, see http://beta.visl.sdu.dk/ (demo, documentation, NLP papers).

[1] The ordering of differentiator-translation pairs is important - defaults, with fewer restrictions, have to come last. For the numerical value of a given translation, 1/rank is used.

[2] The full list of differentiators for this verb contains 13 cases, including several prepositional complements not included here *(regne efter, blandt, fra, om, sammen, ud, fejl ...)*

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 159–160.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Dialog OS: an extensible platform for teaching spoken dialogue systems

Daniel Bobbert
CLT Sprachtechnologie GmbH
Science Park Saar
66123 Saarbrücken, Germany
bobbert@clt-st.de

Magdalena Wolska
Computational Linguistics
Universität des Saarlandes
66041 Saarbrücken, Germany
magda@coli.uni-sb.de

1 Introduction

With the area of spoken dialogue systems rapidly developing, educational resources for teaching basic concepts of dialogue systems design in Language Technology and Computational Linguistics courses are becoming of growing importance. Dialog OS[1] is an extensible platform for developing (spoken) dialogue systems that is intended, among others, as an educational tool.[2] It allows students to quickly grasp the main ideas of finite-state-based modelling and to develop relatively complex applications with flexible dialogue strategies. Thanks to Dialog OS' intuitive interface and extensibility, system implementation tasks can be distributed among non-technically- and technically-oriented students making the tool suitable for a variety of courses with participants of different backgrounds and interests. Below, we give a brief overview of the framework and outline some of the student projects in which it was used as a basis for dialogue management and modelling.

2 Dialog OS: a brief overview

Dialog OS is an extensible platform for managing and modelling (spoken) dialogue systems. It comprises an intuitive Graphical User Interface (GUI), default dialogue components, and a communications API to build new components. Dialog OS is written in Java and operates in a client-server mode. The central component can handle connections with an arbitrary number of client components (or "Devices", in Dialog OS terminology) via TCP/IP sockets. Technical requirements for Dialog OS are: 1 GHz Pentium, 512 MB RAM, Windows 2000/XP, Java Runtime 1.5 or newer.

Default components Dialog OS comes with built-in modules for professional quality speech input and output using technology from Nuance and AT&T. As part of the platform, Dialog OS provides a number of default input/output device clients that can be directly connected without extra programming. Among those are: a simple text console for text-based input and output, a sound player, and a default client for a connection to an SQL database. CLT can also provide built-in connections to a number of other research and commercial Automatic Speech Recognition (ASR) and Text-To-Speech (TTS) systems.

Extensibility Dialog OS can be extended to work with an arbitrary number of clients through a Java-based API. The low-level communication between Dialog OS and the clients is handled by a dedicated internal protocol and remains invisible to the user. Programming a new client involves a Java implementation of a high-level functional protocol for the given client, without having to deal with the details of network connection with the dialogue engine itself.

FSA-based dialogue modelling The central part of the dialogue system is the dialogue model. Dialog OS offers an intuitive way of modelling dialogues using Finite State Automata (McTear, 2002). Building a dialogue model consists of adding and linking dialogue graph nodes represented as icons on a GUI workspace. Those include input/output nodes and internal nodes, for example, to execute scripts, set and test variables, enter a sub-graph (i.e. execute a sub-automaton).[3] The dialogue model is stored in an XML format.

Dialog OS builds on the functionality of its predecessor, DiaMant (Fliedner and Bobbert, 2003). Below, we list some of the features taken over, extended or enhanced in Dialog OS:

[1] Dialog OS is a registered trademark of CLT Sprachtechnologie GmbH. Other product and company names listed are trademarks or trade names of their respective owners.

[2] Dialog OS is developed and distributed by CLT Sprachtechnologie GmbH: http://www.clt-st.de/dialogos

[3] The expressive power of the dialogue models is effectively that of push-down automata.

User input The input nodes for text-based or spoken interaction allow to specify a list of expected input values; outgoing edges are created automatically. User input may be matched directly against the list, or against a regular expression. For spoken input via default ASR components, both the recognised string and the recognition confidences can be accessed.

Built-in data types Global variables can be of simple types (e.g. String, Integer, etc.) as well as more complex data structures of key-value pairs.

Scripting language Dialog OS includes an interpreter of a JavaScript-like scripting language for simple data manipulation functions, e.g., to match input against a regular expression. These can be integrated through a *Script* node.

Sub-automata The *Procedure* node allows for flexible and modular dialogue modelling. Recurring parts of the dialogue can be saved as individual parameterisable sub-automata, direct counterparts of sub-routines in programming languages.

Wizard-of-Oz (WOz) mode Dialog OS can be run in WOz mode (Fraser and Gilbert, 1991) in which one or more of the "Devices" are simulated and dialogue execution details are saved in logfiles; this allows to set up small-scale WOz experiments.

3 Dialog OS in the classroom

We have been using Dialog OS and its predecessor at Saarbrücken in a number of courses involving spoken dialogue systems. Notable features that make it suitable for educational purposes include:

Intuitive interface: Learning to use Dialog OS takes very little time. Thanks to the GUI, even non-computational students can easily configure a functional system with little (or even no) knowledge of programming. The low learning overhead allows to concentrate on modelling interesting dialogue phenomena rather than technical details.

High-level language for building new components: A Java-based API makes the development process efficient and allows for the final system to be built on a single programming platform and kept highly modular.[4]

Below we briefly outline larger spoken dialogue systems developed as part of software projects using the Dialog OS framework.

Talking Robots with LEGO MindStorms® Within two runs of the course, students built various speech-enabled mobile robots using LEGO and Dialog OS as dialogue framework (Koller and Kruijff, 2004). Integration involved writing a client to control the MindStorms RCX (Dialog OS provides built-in support for MindStorms NXT). Luigi Legonelli, the Shell Game robot, and a modified version of Mico, the bar-keeper,[5] have been presented at CeBIT '03 and '06, respectively.

Campus information system A group of three students built a spoken information system for Saarland University campus. The system can answer questions on employee's offices, telephone numbers, office locations, etc. The highlights of the system are modularity[6] and an adaptive clarification model needed to handle many foreign names and foreign user accents.

Talking elevator In two editions of this course, students built speech interfaces to the elevators in the institute's buildings. In the first course, a simple mono-lingual system was developed. In an ongoing project, students are building a trilingual system with speaker identification, using their own version of a Nuance client and an elevator client that communicates with the elevator hardware via a serial protocol.

References

G. Fliedner and D. Bobbert. 2003. DiaMant: A Tool for Rapidly Developing Spoken Dialogue Systems. In *Proc. of DiaBruck*, pages 177–178, Wallerfangen, Germany.

N. M. Fraser and G. N. Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5:81–99.

A. Koller and G-J. M. Kruijff. 2004. Talking robots with lego mindstorms. In *Proc. of COLING-04*, pages 336–342.

O. Lemon and X. Liu. 2006. DUDE: A dialogue and understanding development environment, mapping business process models to information state update dialogue systems. In *Proc. of EACL-06*, pages 99–102, Trento, Italy.

M. F. McTear. 1999. Using the CSLU toolkit for practicals in spoken dialogue technology. In *Proc. of ESCA/SOCRATES Workshop on Method and Tool Innovations for Speech Science Education*, pages 113–116, UK.

M. F. McTear. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys (CSUR)*, 34(1):90–169.

[4] A GUI is also part of CSLU (McTear, 1999) and DUDE (Lemon and Liu, 2006) dialogue toolkits. However, DUDE has not yet been tested with novice users, while extending CSLU Toolkit involves programming in C, rather than in a higher-level language such as Java.

[5] Mico's mechanics were substantially re-designed by CLT, however, the dialogue model was, for the most part, taken over from the student project.

[6] Currently, the system supports only German, but the modular design was motivated by anticipated extensions for English and French.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 161–162.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Complex Taxonomy Dialogue Act Recognition with a Bayesian Classifier

Mark Fishel
Dept of Computer Science
University of Tartu
Tartu, Estonia
`fishel@ut.ee`

1 Introduction

This paper describes the experiments of performing dialogue act (DA) recognition with a complex DA taxonomy using a modified Bayes classifier.

The main application of DA recognition is in building dialogue systems: classifying the utterance and determining the intention of the speaker can help in responding appropriately and planning the dialogue. However, in this work the target application is human communication research: with tagged DAs it is easier to search for utterances of a required type in a dialogue corpus, to describe the dialogues with a general model of dialogue moves, etc.

The DA taxonomy, used in the current work, was designed for the Estonian Dialogue Corpus (EDiC) (Hennoste et al., 2003). This means two additional difficulties for DA recognition. Firstly, DA taxonomies used for human communication research are as a rule much more detailed than in case of dialogue systems (e.g., comparing DCIEM (Wright, 1998) and CallHome Spanish (Ries et al., 2000) taxonomies); therefore, more DAs have to be distinguished, with several of them having unclear meaning boundaries. Secondly, Estonian is an agglutinative language with 14 cases, a complex system of grouping and splitting compound nouns, heterogeneous word order and several other features that make natural language processing harder.

2 Experiments

2.1 Experiment Setup

In order to determine the optimal set of input features additive feature selection was applied. All of the tests were performed using 10-fold cross-validation.

In this work we only tried simple features, not involving morphological analysis, part-of-speech tagging, etc. The used ones included DA tag bi- and trigrams, keywords and the total number of words in the utterance. Keyword features included the 1st word, first 2 words and first, middle and last words as a single dependency. We also tried stemming the words and alternatively leaving only the first 4 characters of the word.

The learning model used in this work is the Bayes classifier. Its original training/testing algorithm supports only a fixed number of input features. This makes it harder to include information with variable size, such as the set of the utterance words. In order to overcome this limitation, we slightly modified the algorithm by calculating the geometrical average of the conditional probabilities of the DA tag, given each utterance word. With this approach the probabilities remain comparable despite the variable length of the utterances.

The corpus used for training and testing is described in greater detail in (Gerassimenko et al., 2004), updated information can be found online[1]. The version used in the experiments contains 822 dialogues (a total of 32860 utterances) of mixed content (telephone conversations in an information service, at a travelling agency, shop conversations, etc).

2.2 Results

After the feature selection process converged, the following features were included into the selection:

[1] `http://math.ut.ee/~koit/Dialoog/EDiC`

DA tag trigram probabilities, the geometrical mean of the word-tag conditional probabilities and the number of words in the utterance. Stemming was not performed in the final preprocessing.

The resulting cross-validation precision over the whole set of dialogues was 62.8% with the resulting feature set. In general the most typical DA tag to be confused with was the most frequent one. In addition, some tags were frequently confused with each other.

In addition to the objective precision estimation provided by cross-validation, we also wanted to have a direct comparison of the resulting DA tagger with the human taggers. For that we applied the tagger to both human tagged parts, used for calculating the human agreement. The resulting precisions for the two parts are 80.5% and 78.6%.

3 Discussion

It is interesting to note that the resulting selection of representation features included only simple text-based features. Although the task of DA recognition belongs to computational pragmatics in natural language processing, in this case it gets solved on the level of pure text, which is even lower than the morphology level.

Future work includes several possibilities. In particular, several output errors of the trained classifier seem obvious to solve to a human tagger. For instance, several utterances containing wh-words are misclassified as something other than wh-questions. There are at least two possibilities to treat that kind of problems. Firstly, a set of rules can be composed by professional linguists to target each output problem individually. This approach has the advantage of guaranteed improvement in the required spot; on the other hand, manually composing the rules can result in overlooking some global influences on the remaining utterance cases, which can cause decreased performance in general. Another way to address the output errors would be to add more descriptive features to the input.

4 Conclusions

We have described a set of experiments, aimed at applying a Bayes classifier to dialogue act recognition. The targeted taxonomy is a complex one, including a large number of DA tags.

Additive feature selection was performed to find the optimal set of input features, representing each utterance. The tested features included n-gram probabilities and keyword-based features; the latter were tested both with and without stemming.

The resulting precision of the trained model, measured with 10-fold cross-validation is 62.8%, which is significantly higher than previously achieved ones. The selected features included DA tag trigram probabilities, number of words probability and the geometrical mean of the word-tag conditional probabilities of all the utterance words.

The model was compared to the agreement of human taggers in the targeted taxonomy – this was done by applying it to the same test corpus that was used in calculating the agreement. The two resulting precisions are 80.5% and 78.6%, which is very much near the human agreement (83.95%).

There is much room for further development of the classifier. This includes adding more specific features to the model's input, manually composed output post-processing rules, etc.

References

Olga Gerassimenko, Tiit Hennoste, Mare Koit, Andriela Rääbis, Krista Strandson, Maret Valdisoo, and Evely Vutt. 2004. Annotated dialogue corpus as a language resource: An experience of building the estonian dialogue corpus. In *Proceedings of the 1st Baltic Conference "Human Language Technologies. The Baltic Perspective"*, pages 150–155, Latvia, Riga.

Tiit Hennoste, Mare Koit, Andriela Rääbis, Krista Strandson, Maret Valdisoo, and Evely Vutt. 2003. Developing a typology of dialogue acts: Tagging estonian dialogue corpus. In *Proceedings of the 7th Workshop on the Semantics and Pragmatics of Dialogue, DiaBruck 2003*, pages 181–182, Saarbrücken, Germany.

Klaus Ries, Lori Levin, Liza Valle, Alon Lavie, and Alex Waibel. 2000. Shallow discourse genre annotation in callhome spanish. In *Proceecings of the International Conference on Language Ressources and Evaluation (LREC-2000)*, Athens, Greece.

H. Wright. 1998. Automatic utterance type detection using suprasegmental features. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP'98)*, volume 4, page 1403, Sydney, Australia.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 163–164.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Default preferences in *donkey* anaphora resolution

Francesca Foppolo
Dept. of Psychology - University of Milano Bicocca
v.le dell'Innovazione 11
20126 Milano (Italy)
francesca.foppolo@unimib.it

Abstract

I will present an experimental study on the interpretation of pronouns in donkey sentences, i.e. sentences such as "Every farmer who owns a donkey beats *it*" that admits of two interpretations: the *universal* (= Every farmer who owns a donkey beats all the donkeys he owns) or the *existential* interpretation (=Every farmer who owns a donkey beats one of the donkeys he owns). By means of two reaction time experiments I show: (i) that the distribution of the two interpretations is the one predicted by Kanazawa's generalization (1994): the interpretation of donkey pronouns seems to be sensitive to the left monotonicity properties of the head determiner (Experiment 1); (ii) that such interpretations seem to be a matter of preference, i.e. a default that comes about in relatively "neutral" contexts and that appropriate context manipulations can override (Experiment 2).

1 Introduction

I will present an experimental study conducted with Italian adults concerning the interpretation of pronouns in donkey sentences. Consider the standard example in (1):

(1) Every farmer who owns a donkey beats *it*

As is well known from the literature, the pronoun *it* in (1) admits of two interpretations, the *universal* (∀) one and the *existential* (∃) one, interpretations whose truth conditional import can be represented as in (2) and (3) respectively :

(2) ∀-reading:
$\forall x\ [[\text{farmer}(x) \wedge \exists y\ \text{donkey}(y) \wedge \text{has}(x,y)] \rightarrow \forall z\ [\text{donkey}(z) \wedge \text{has}(x,z) \rightarrow \text{beats}(x,z)]]$
= *Every farmer who owns a donkey beats all the donkeys he owns*

(3) ∃-reading:
$\forall x\ [[\text{farmer}(x) \wedge \exists y\ \text{donkey}(y) \wedge \text{has}(x,y)] \rightarrow \exists z\ [\text{donkey}(z) \wedge \text{has}(x,z) \wedge \text{beats}(x,z)]]$
= *Every farmer who owns a donkey beats one of the donkeys he owns*

There are many proposals as to how these readings come about. However, our concern here is not so much to choose among such proposals (though eventually, we believe that our results will be relevant to such an issue). Our immediate concerns here are rather to experimentally test an interesting generalization regarding the distribution of ∀- and ∃-interpretations, put forth in Kanazawa (1994). According to Kanazawa, the preferred interpretation of donkey pronouns is the one that preserves the monotonicity properties of the determiner. This makes the following predictions on the sample set given in (4).

(4)

Det.	Monotonicity	interpretation
Every	↓↑	∀
No	↓↓	∃
Some	↑↑	∃

Kanazawa's point, to whose work we must refer for details, is that the interpretations in the last column in (4) are the only ones that preserve (in a donkey anaphora context) the monotonicity properties of each lexical determiner, spelled out in the second column. While there has been some experimental work on how donkey pronouns are interpreted (cf., e.g. Geurts, 2002), no work has tried to experimentally probe Kanazawa's claim. Yet, if empirically supported, such a claim would be important, as it would show that the semantic processor must have access to an abstract formal property of an unprecedented kind (namely, monotonicity preservation in non C-command anaphora).

2 The experimental study

1.2 Material and Procedure

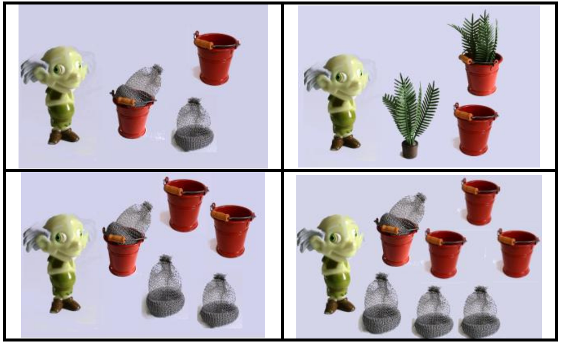
We carried out a reaction-time study with a total of 66 Italian-speaking adults. Subjects were asked to evaluate *donkey sentences* introduced by different types of quantifiers with respect to scenarios displaying four pictures. Sentences were presented in two critical conditions: in the absence of an extra-linguistic context (Exp.1) and after the addition of a biasing context (Exp. 2). In both cases, to avoid interferences from extra-linguistic knowledge, we used strange characters (introduced as aliens) with weird objects to which only fantasy names were given. A sample of critical sentences used is given in (5)-(7), and one of the scenarios proposed is presented next:

(5) *Every Flont that has a vilp keeps it in a bin*

(6) *No Flont that has a vilp keeps it in a bin*

(7) *Some Flont that has a vilp keeps it in a bin*

<u>Scenario</u> (in critical condition)



Note that, given that two alternative interpretations can be associated to each sentence (as shown is (2) and (3) above), the scenario above makes the critical sentences true under one interpretation, but crucially false under the other. In case of Exp. 2, a biasing context was added before the same scenario appeared, in the aim of inducing subjects to accept the *donkey sentence* under the reading predicted as dispreferred by Kanazawa's generalization.

2.2 Results

Subjects' answers in Exp. 1 seems to conform to the predictions derived from the generalization in (4), at least in case of *Some* and *No*: in both cases, the reading that emerged as preferred in the critical condition was the *existential* one (87% and 93% in case of *Some* and *No* respectively). In case of *Every*, instead, subjects split. However, this result is compatible with the results obtained in Exp. 2, which show that subjects do in fact access the alternative interpretation of the anaphora, but crucially that its availability varies in accordance with the initial head determiner: the dispreferred (∃) reading is very easily accessed in case of *Every* (a significantly higher proportion of subjects (i.e. 81%) judged sentence (5) TRUE in the scenario above in Exp. 2). Conversely, the access to the dispreferred (∀) interpretation of the anaphora is much harder in case of sentences (6) and (7), even in presence of a context that biases subjects towards this interpretation.

3 Conclusion

Two main points emerge from our results. First, Kanazawa's generalization does appear to be empirically supported. How donkey pronouns are interpreted seems to be sensitive to the monotonicity properties of the determiners involved along the lines indicated in (4). Second, such interpretations seem to be a matter of preference (i.e. a default that comes about in relatively "neutral" contexts). As Exp. 2 shows, appropriate context manipulations lead to the emergence of the alternative interpretation. These results illustrate several general points. For one thing, they show that speakers unconsciously and systematically compute abstract properties pertaining to entailment patterns, as they tend to choose the interpretation of the donkey pronouns that retains the lexical properties of the determiner. Work on negative polarity has arguably shown sensitivity to monotonicity patterns in determining the distribution of items like *any*. Here we detect a similar phenomenon in connection with a purely interpretive task (namely, how pronoun readings in non C-command anaphora are accessed). This paves the way for further research (e.g., with respect to figuring out *how* various readings come about, and with respect to testing the present claim with other determines and settings) and confirms the value of integrating theoretical claims in semantics with experimental work.

Selected References.

Chierchia, G. (1995). *Dynamics of meaning: anaphora, presupposition, and the theory of grammar*. Chicago, University of Chicago Press.

Kanazawa, M. (1994). Weak vs. Strong Readings of Donkey Sentences and Monotonicity Inference in a Dynamic Setting. *Linguistics and Philosophy* 17: 109-158.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 165–166.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Discourse Management in Voice Systems for Accessing Web Services

Marta Gatius **Meritxell González**
Technical University of Catalonia, Software Department
Jordi Girona 1-3, Campus Nord, 08034 Barcelona, Spain
{gatius,mgonzalez}@lsi.upc.edu

Abstract

This paper describes the discourse management component of a dialogue system that supports voice and text access to the web services in different languages: English, Spanish, Catalan and Italian. The dialogue manager follows the information state theory and uses communication plans that are generated when the system is adapted to a new web service. To facilitate the generation of these plans we have defined general communication plans for different types of web services.

1 Introduction

There already exist voice systems for accessing specific web services. However, most of these services only support system-initiative dialogues, in which the system drives the interaction asking the user the information the service needs.

System-initiative dialogues have proved efficient when only the system knows which particular data the service needs from the user, i.e. when inexperienced users wants to perform an online transaction that require particular data. However, there are situations in which the users can take the initiative to give the information the system needs, e.g., in the case of information seeking, only the user knows what (s)he wants to find.

Understanding what the user's intends becomes more difficult in mixed-initiative dialogues, because user's interventions may be difficult to predict (i.e., they may be unrelated to the system's questions previously asked). In this paper we describe the approach we followed in the multilingual dialogue system (DS) to access public administration web services we developed in the context of the European project HOPS (http://www.hops-fp6.org/). In this DS, the dialogue is controlled by a general module, the dialogue manager (DM), which uses communication plans to recognize the user's intention and decide next actions.

2 The Dialogue System

The DS has been designed to support access to different types of web services through several channels. The final prototype supports access through the telephone and the web in different languages (English, Spanish, Catalan and Italian) to two web services: a transactional service for collecting large objects and an informational service on cultural events.

The voice component of the DS is that of the Loquendo multilingual VoiceXML platform. The speech recognizer uses grammars (following the standard SRGS syntax) modeling user utterances. The transcription of the utterance is passed to a natural language parser and processor, which performs a deep syntactic and semantic analysis. The resulting semantic interpretation is then passed to the DM. The same parser is used to process the text introduced through the web.

3 The Dialogue Manager

The discourse management component we have developed supports mixed-initiative dialogues and other complex phenomena, such as feedback strategies, which are especially needed to deal with misunderstandings. An example of dialogue supported by our system can be seen in Figure 1.

Our DM follows the information state theory (IST) of language modeling (Traum et al., 1999). This model combines the efficiency of structural approaches and the flexibility of plan-based approaches. Although the IST deals with complex aspects of dialogue, such as beliefs, intentions and plans, it is not difficult to implement because it provides the tools to formalize this knowledge.

As in other practical DSs (Larsson, 2002), the plans used by our DM to recognize the user's intention and determine next communication actions are generated statically, when a new service is incorporated to the service. Because we consider that each task a web service can perform is a possible user's goal, we generate a communication plan for each service task.

S1: Welcome to the automatic platform of Barcelona. May I help you?
U1: I search cultural events
S2: You said I search cultural events. I'm sorry, I don't understand.
U2: I search for information about cultural events
S3: Wait a moment, please. Can you tell me some data about the event or events you are interested in?
U3: Movies in the filmoteca
S4: Ok, you are interested in the title of the event. The event type is cinema. The place is filmoteca.
*** <database consultation>
....

Figure 1: An example of dialogue

4 The Communication Plans

The objects of the plans are the service tasks and their parameters. The operators (actions) of the plans can be simple and complex actions. Possible simple actions are any of the central dialogue moves (ask, answer and ask for confirmation) and the accesses to the web services. Complex actions consist of a sequence of simple actions. The preconditions of the actions are related to the parameters values of the service tasks. Additional information about the service tasks parameters (i.e., the set of possible values) can also be included in the plans.

In order to facilitate the generation of the plans we have defined general communication plans for different types of services. There already are general descriptions of the interactions that take place when accessing web services. We have represented those general descriptions as top-level plans with abstract operators that can be decomposed into a group of steps. These steps can bee semi-automatically instantiated for each web service. We have considered two types of services: transactional and informational.

4.1 Plans for Transactional Services

We consider transactional services are those performing a transaction. Usually, those services require specific information from the user. The top-level plan to follow when accessing those services consists of the following sequence of general actions:

- *Obtaining the value of the input parameters*. The system asks the user this data.
- *Access the transactional service*.
- *Give information about the transaction*.

4.2 Plans for Informational Services

Seeking-information dialogues have focused several works. We have only considered the communication that usually takes place when the user accesses a simple web service describing a particular item (or a set of items). The top-level plan to access those services is the following:

- *Obtaining the value of the focus parameters*. Focus parameters are those needed to restrict the search. Although usually there is a set of possible parameters to restrict the search, unlike in the case of transactional services, the user may choose to give the value of only a subset of them. Optionally, the system also asks the specific data the user wants about the element searched (i.e., the location).
- *Access the informational service*.
- *Presentation of the results*. If two many elements are obtained the DM determines the new parameters to restrict the search, which are then asked to the user.

5 Conclusion and Future Work

Experiments performed by ten volunteers showed that the task success achieved using the text mode is high (90%), higher in transactional services (95%) than in informational services (85%). More formal tests on the performance of the DM will be done in the future. Future work will also include the adaptation of the DM to access other different types of web contents.

Reference

Araki, M. and Tachibana, K. Multimodal Dialog Description Language for Rapid System Development. In the Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue, (Sidney, 2006).

Larsson, S. Issue-based Dialogue Management. PhD. Thesis, Goteborg University, 2002.

Traum, D., Bos, J., Cooper, R., Larsson, S., Lewin, I., Mathesson, C., Poesio, M. A model of Dialogue Moves and Information State Revision. Trindi Technical Report D2.1, 1999. http://www.ling.gu.se/projekt/trindi/publications.html.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 167–168.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Adaptation of the use of colour terms in referring expressions

Markus Guhe, Ellen Gurman Bard
Human Communication Research Centre,
Linguistics and English Language
University of Edinburgh
40 George Square, Edinburgh, EH8 9LL
{m.guhe, e.bard}@ed.ac.uk

Abstract

In a modified Map Task we looked at the use of colour terms. Colour terms in this version of the Map Task are unreliable, because (1) they can mismatch between the maps (2) about half of them are obscured on the Instructions Follower's map by 'ink blots'. The data show that the dialogue partners adapt to this property of the task environment by using fewer colour terms over time.

1 Introduction

When referring to objects linguistically, humans use referring expressions, that is, expressions that single out one object from the set of potential referents. A standard assumption in the literature on generating referring expressions is that the semantic structure of the expression can be specified by a set of attributes, e.g. type (alien, fish), size, colour. Given this, the main problems are (1) to find an efficient generation algorithm that selects attributes which single out one object and (2) to generate naturally sounding expressions.

The most prominent proposal for what an efficient, cognitively plausible algorithm could be is Dale and Reiter's (1995) algorithm, which has been enhanced and modified in many ways. The main problem that has to be solved by such algorithms is that they have to select those attributes that humans choose in the same situation. Jordan and Walker (2005) present modifications to Dale and Reiter's algorithm on how the selection of attributes can be adapted to the properties of linguistic corpora. These algorithms already incorporate results of psychological findings (e.g., Brennan and Clark's 1996 conceptual pact model), but they do not account for changes over time.

2 Experiment

In a modified Map Task (Anderson et al., 1991; Guhe et al., 2006) we asked whether the participants adapt to properties of the task environment (the maps) when referring to the landmarks. In the Map Task two dialogue partners – the Instruction Giver (IG) and the Instruction Follower (IF) – each have a map of the same location (Fig. 1). IG's map contains a route not present on IF's

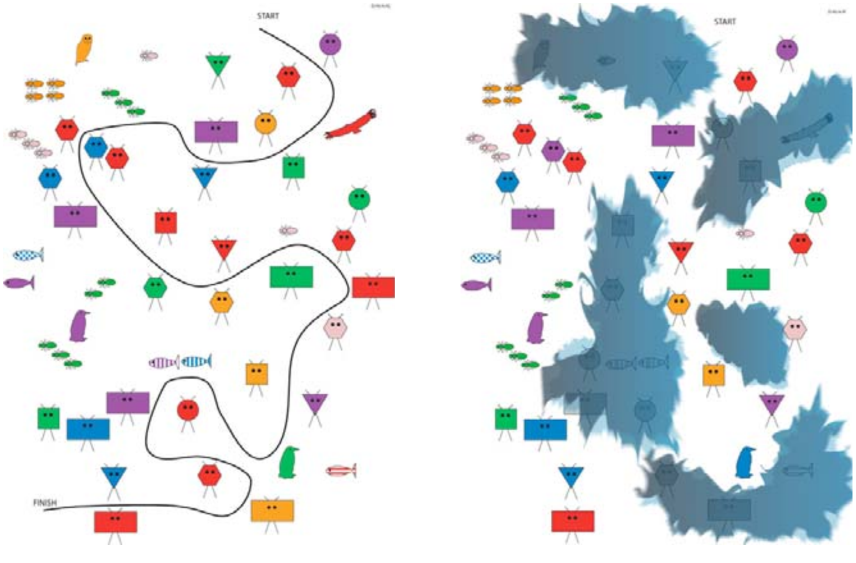


Figure 1: Maps for the analysed dialogues; IG's map (left) contains a route and a START and STOP mark; IF's map contains 'ink blots' that obscure the colour of some objects; circles (added here for expository purposes) indicate the differences between the maps

map. They communicate to reproduce IG's route on IF's map. Players cannot see each other's maps. They use landmarks for navigation. Although most landmarks are identical on both maps, some differ by: (1) being absent on one of the maps or present on both; (2) having clearly different attributes; (3) being affected or not by 'ink damage' that obscures the colour of some landmarks on IF's map.

Our Map Task (Fig 1) has three experimental variables: (1) homogeneity (whether the landmarks are of one or different kinds, e.g. only aliens, or aliens and fish); (2) orderliness (whether the 'ink blot' obscures a continuous stretch of the route); (3) animacy. These are varied factorially so that each pair of participants (dyad) completes a set of 8 map pairs. There are 32 dyads.

3 Data

Currently 210 of the 256 dialogues are transcribed and used here. Each dialogue is about 10 minutes long. Overall the 210 transcripts contain 184,711 words of which 5,251 are colour terms.

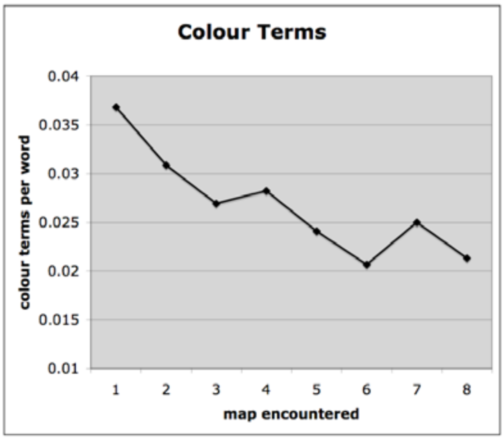


Figure 2: Use of colour terms per word over time.

Fig. 2 shows the mean number of colour terms per spoken word across the 8 map pairs that each dyad encounters. As the task environment affords no other occasions to use colour terms, we make the simplifying assumption here that all colour terms are used for referring to landmarks. The mean number of colour terms decreases over the course of the 8 maps. There is a significant negative correlation (r = -0.172, p < 0.01) between the rate of colour terms used and the number of the encountered map.

A 3-way repeated measures ANOVA showed that of the 3 experimental variables only landmark homogeneity affected the use of colour terms: (F_1(1,20) = 12.26, p = 0.02) on average the mixed landmark condition attracts fewer colour terms per word (0.024) than the uniform landmark condition (0.032).

4 Conclusions

The participants in our Map Task pick up the fact that colour is an unreliable attribute in referring to the landmarks on the maps. The adaptation is not a sudden change in behaviour but is a gradual adaptation to the properties of the items they have to refer to.

The effect of homogeneity is most likely due to the difficulty of the maps with landmarks of just one kind: the type attribute does not distinguish such landmarks; colour must be used to identify the target landmark.

The main result is that the use of colour terms changes over time during a task, which is not accounted for in Jordan and Walker's (2005) model, and to our knowledge such a model does not exist yet. For an adequate dialogue model it is insufficient to simply let the computer choose a level of colour terms (observed in a suitable corpus), because that would be unnatural. In such models the referring expressions in the first maps would not be natural, because they would use the colour attribute less often than humans (analogously too often in the last maps). One goal of our current project is to develop such a model.

Acknowledgements

This research was supported by the National Science Foundation (IIS-0416128). Thanks to the Memphis team for providing transcriptions: Max Louwerse, Gwyneth Lewis and Megan Zirnstein.

References

Anderson, A., et al (1991). The HCRC Map Task Corpus. *Language and Speech*, 34: 351–366.

Brennan, S.E. and Clark, H.H. (1996) Conceptual Pacts and Lexical Choice in Conversation. *JEP: Learning, Memory, and Cognition*, 22:1482–1493.

Dale, R. and Reiter, E. (1995). Computational interpretations of the Gricean maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Guhe, M., Steedman, M., Bard, E.G., Louwerse, M.M. (2006) Prosodic marking of contrasts in information structure. In: *Proceedings of BranDial'06.* 179–180.

Jordan, P. W. and Walker, M. A. (2005). Learning content selection rules for generating object descriptions in dialogue. *Journal of Artificial Intelligence Research*, 24:157–194.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 169–170.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

A Platform for Designing Multimodal Dialogic and Presentation Strategies

**Meriam Horchani[1,2], Dominique Fréard[1,3],
Eric Jamet[3],Laurence Nigay[2], Franck Panaget[1]**

[1] France Télécom R&D
22300 Lannion, France
{*firstname.name*}
@orange-ftgroup.com

[2] LIG, HCI Team
38000 Grenoble, France
laurence.nigay
@imag.fr

[3] University of Rennes 2
35000 Rennes, France
eric.jamet@uhb.fr

Interactive and dialogue systems are daily used in various contexts and with different devices. This diversity guarantees the current and upcoming success of multimodal services. Although several multimodal dialogue systems have been built, their design, their implementation and their testing remain a difficult task. We address this problem by focusing, in this paper, on a software component dedicated to the implementation and testing of dialogic and presentation strategies. We characterize data manipulated by this component, using results from experimental studies on impact of presentation strategies.

1 Our approach and our platform

For the generation of outputs in multimodal dialogue systems, two concepts are essential: a modality and a presentation task. Adopting a system-oriented perspective, we consider a *modality* (input or output) as the coupling [*d,L*] of a physical device *d* with an interaction language *L* (Nigay & Coutaz, 1995). A *presentation task* refers to the presentation of a coherent piece of information. This piece can be either elementary or composed. The granularity of elementary presentation tasks is at the discretion of the designers. Each answer of the system is composed of at least one presentation task.

The generation process generally consists of three choices: (1) the content of the answer of the system; (2) the modalities to use in order to present this answer; (3) the distribution of the answer on these modalities. Within this process, we distinguish the dialogic strategy (DS) selection from the presentation strategy (PS) selection. DS is generally determined during step (1) and PS is shared out among steps (2) and (3).

The DS selection involves the selection of the answer. We identify three initial DS in cooperative multimodal dialogue information systems:

- DS1, "relaxation": the system suggests alternative solutions or alternative search criteria;
- DS2, "statement": the system provides found solutions;
- DS3, "restriction": the system suggests possible criteria to restrict the solution set.

The PS selection refers to the selection of the modalities for each piece of information. The PS influences the user's processing and the user's behaviour (cf. Section 2). In addition, presentation constraints and available modalities must influence the selection of a particular DS. That is why we think that DS and PS are inter-related and as such they must be decided in parallel at each step. This leads us to propose a platform for implementing and testing output strategies in multimodal dialogue systems that includes a component dedicated to select both DS and PS.

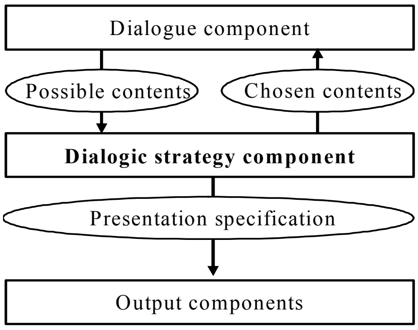


Figure 1. The platform for exploring dialogic and presentation strategies

Figure 1 shows our platform based on the ARCH meta-model architecture (UIMS, 1992). It includes a Dialogic Strategy Component (DSC) which acts as an intermediary between the classical dialogue component and the output (i.e. presentation and interaction toolkit) components. Instead of the dialogue component selecting a DS for each dialogue turn, it sends to the DSC all the possible contents (i.e. all the possible DS). The DSC then selects simultaneously the DS and the PS and it defines the presentation specification of the multimodal answer: a presentation specification is a composition of at least two presentation tasks using the CARE (Complementarity, Assignment, Redundancy, Equivalence) properties (Nigay & Coutaz, 1997). In addition, the DSC conveys the chosen contents to the dialogue component in order to maintain an accurate dialogic history. So the DSC manages the complete generation process. For further details, see (Horchani et al., 2007).

To improve our platform, we need to specify concepts which are manipulated by the DSC.

2 Contribution of a study on impacts of presentation strategies

The aim of the experiment is to study the users' reaction (verbal behaviour, cognitive load, and memorization) according to the multimodal answer of the system. We need to characterize output information in order to identify links between modalities and types of information and to test these links during the experiment.

We identify a dual task analysis of interactive and dialogue systems. On the one hand, three main types of information communicable to the user are suggested in order to structure the design of dialogue outputs for any kind of systems (Nievergelt & Weydert, 1980): trails refer to past actions, sites correspond to the current action or information to give and modes are about possible actions to come. In the context of human-computer dialogue, trails are generally called *feedback* ("You want an appointment Friday"), sites are called responses ("There are *x* available appointments") and modes are called *openings* ("What is your choice?"). On the other hand, users often carry out more than a single task when communicating with dialogue systems: we distinguish the field task – which is reached thank to the responses – and the interaction task – which includes feedbacks and openings.

For our experiment, information which reaches one task was allocated to one modality. Using a complementary combination of auditory outputs (A = [loudspeakers, natural language]) and visual outputs (V = [screen, hypertext]), we tested four PS {AAA, AVA, VAV, VVV}: the first letter refers to the feedback modality, the second one to the response modality and the last one to the opening modality. During the experiment, the participants conversed with a Wizard of Oz simulating a system dedicated to fix medical appointments. Four groups (one for each PS) of 20 participants (10 males and 70 females, 17-26 years old students (M=19)) took part in the experiment. The results showed the relevancy of considered dual task analysis and it underlines that modalities are not equivalent with regard to the type of information: the PS, as the DS, has an impact on the dialogue. For further details, see (Fréard et al., 2007).

These conclusions are used to improve our platform. Using the three types of information, we characterize presentation tasks into three types in our platform: feedback presentation tasks, response presentation tasks and opening presentation tasks. This better characterization of the presentation tasks increases the set of possibilities for multimodal outputs: given a set of possible contents, the answer of the system results from the selection of the DS (i.e. the content to convey) and of the PS (i.e. the types of presentation tasks and their modality allocation).

Conclusion

We have presented a platform including a component dedicated to the intertwined management of dialogic and presentation strategies. Using conclusions from an experimental study on the impact of presentation strategies on the user's reaction, we detail information manipulated by our component: indeed, a presentation task can be a feedback presentation task, a response presentation task or an opening presentation task. The answer of the system is a combination of these tasks. In future work, we will use our platform and to perform experimental studies on links between quantity of information and selected strategies.

References

Fréard, D., Jamet, E., Le Bohec, O., Poulain, G., & Botherel, V. (2007). Subjective measurement of workload related to a multimodal interaction task. *HCII'07*.

Horchani, M., Nigay, L., & Panaget, F. (2007). A platform for output dialogic strategies in natural multimodal dialogue systems. *IUI'07*.

Nievergelt, J., & Weydert, J. (1980). Sites, modes, and trails. In *Methodology of interaction* (pp. 327-338).

Nigay, L., & Coutaz, J. (1995). A generic platform for ad-dressing the multimodal challenge. *CHI'95*.

Nigay, L., & Coutaz, J. (1997). Multifeature systems: The CARE properties and their impact on software design. In *Intelligence and multimodality in multimedia interfaces*.

UIMS. (1992). A metamodel for the runtime architecture of an interactive system. *SIGCHI bulletin, 24*, 32-37.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 171–172.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Adapting a Statistical Dialog Model for a New Domain

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

Abstract

In this paper, we present our current work for adapting a statistical methodology for dialog management within the framework of a new domain. This methodology, that is automatically learned from a data corpus and is based on a classification process, has been previously applied in a spoken dialog system that provides railway information. We summarize this approach and the work that we are currently carrying out to apply it for developing a dialog system for booking sports facilities.

1 Introduction

Within the framework of dialog systems, the application of statistical methodologies to model the behavior of the dialog manager is nowadays a growing research area (Williams and Young, 2007).

In this field, we have recently developed an approach to manage the dialog using a statistical model that is learned from a data corpus (Hurtado et al., 2006). This work has been applied within the domain of a Spanish project called 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. This corpus was labeled in terms of dialog acts to train the dialog model.

Currently, we are adapting this methodology in order to develop a dialog manager for a new project called EDECAN. The objective of the ongoing EDECAN project is to increase the robustness of a spontaneous speech dialogue system through the development of technologies for the adaptation and personalization of the system to different acoustic and application contexts. The task that we have selected is the booking of sports facilities in our university. Users can ask for the availability, the booking or cancellation of facilities and the information about his/her current bookings.

2 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). A formal description of the proposed statistical model is as follows:

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 (i.e., in the history of the dialog preceding time i). 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 order in which the user provided the information is not taken into account, and the selection of the best system answer is made using this maximization:

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

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 information that is not contained in the DR, but is important to decide the next system answer. This is the case of the task-independent information (*Affirmation*, *Negation* and *Not-Understood* dialog acts).

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.

3 Our present work

The task defined for the EDECAN project is to provide an oral interface for booking sports facilities. The main difference with regard to the DIHANA task is that now the dialog manager not only provides information but also modifies the application data (i.e. after booking or cancelling a court). The module that controls the booking application (*Application Manager, AM*) performs two main operations. On the one hand, this module has to perform the queries to the database. On the other hand, it has to verify if the user query follows the regulations defined for the task (a user can book only one court a day, the facilities can not be booked if the user is suspended, etc.).

The result of the queries to the AM has to be considered in order to generate the system answer. For instance, in order to book the facilities (i.e. a tennis court), if there is not any available court, the system can suggest a change in the user restrictions (i.e the AM verifies if it is possible to perform the booking changing the hour). In case of only one court available, the system confirms if everything is correct before making the booking. Finally, if there is more than one available court, the system asks which court has to be booked.

In order to use the information provided by the AM for selecting the system answer, we consider that two phases are needed. In the first phase, the information contained in the DR and the last state S_{i-1} are used to select the best request to be made to the AM ($\hat{A}_{1_i}$):

$$\hat{A}_{1_i} = \operatorname*{argmax}_{A_{1_i} \in \mathcal{A}_1} P(A_i | DR_{i-1}, S_{i-1})$$

where $\mathcal{A}_1$ is the set of possible requests to the AM.

In the second phase, the final system answer ($\hat{A}_{2_i}$) is generated taking into account $\hat{A}_{1_i}$ and the information provided by the AM (AM_i):

$$\hat{A}_{2_i} = \operatorname*{argmax}_{A_{2_i} \in \mathcal{A}_2} P(A_i | AM_i, A_{1_i})$$

where $\mathcal{A}_2$ is the set of possible system answers.

A preliminary evaluation of this approach has been made by labeling the person-to-person dialog corpus and defining a training and test partitions. Currently we are working in the development of the different modules in the system in order to carry out a supervised acquisition with real users, using the Wizard of Oz technique but also evaluating the automatic answers provided by the DM. A user simulator has also been developed to test and improve the behavior of the DM.

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.

L.F. Hurtado, D. Griol, E. Segarra, and E. Sanchis. 2006. A stochastic approach for dialog management based on neural networks. In *Proc. of Interspeech'06-ICSLP*, pages 49–52, Pittsburgh.

J. Williams and S. Young. 2007. Partially Observable Markov Decision Processes for Spoken Dialog Systems. In *Computer Speech and Language 21(2)*, pages 231–422.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 173–174.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

An empirical study on detection and prediction of topic shifts in information seeking chats

Ichikawa Hiroshi **Tokunaga Takenobu**

Department of Computer Science
Tokyo Institute of Technology
Tokyo Meguro Ôokayama 2-12-1, 152-8552 Japan
take@cl.cs.titech.ac.jp

Introduction This paper describes an empirical study of detecting and predicting topic shifts in *information seeking chat* (Stede and Schlangen, 2004), which is characterised by its more exploratory and less task-oriented nature, where the user does not have a specific goal but obtains useful information of his interest through interaction with the system.

Unlike (Stede and Schlangen, 2004), we do not assume predefined domain specific knowledge to navigate the dialogue; instead we rely on more superficial clues to deal with a broad range of topics.

Differing from a series of topic segmentation research (Hearst, 1997; Ries, 2001; Galley et al., 2003; Olney and Cai, 2005; Arguello and Rosé, 2006), we deal with topic shifts in real-time interaction, aiming at using this technique in a dialogue system. In addition, we predict relevant topic shifts as well as topic shift detection.

Corpus analysis To find useful features indicating topic shifts, we manually analysed a part of the *Mister O corpus* (Ochiai et al., 2005), which is a cross-linguistic video corpus consisting of various types of conversations. Based on the predefined criteria, the transcribed texts of 6 Japanese conversations were divided into topic segments by one of the authors. Extracted features indicating *topic shift utterances (TSU)*[1], and utterances ahead of them are summarized in Table 1 and 2.

Automatic feature detection The features with the asterisk in Table 1 and 2 can be automatically extracted by using superficial clues as follows.

clue expressions: Since we do not have a thorough list of Japanese cue phrases as in (Hirschbeerg and Litman, 1993), we collected a set of cue expressions suggesting topic shifts based on the corpus analysis and introspection. These cue expressions imply this feature.

Table 1: Features of topic shift utterances

Feature	Occurrences
*clue expression**	11
*new words**	10
*initiative change**	9
prior topic	5
others	11
Total	45

Table 2: Features of preceding utterances ahead of topic shifts

Feature	Occurrences
*back-channel**	14
*silence**	13
repetition	6
generalisation	4
*impression**	4
others	13
Total	45

new words: We assume every content word in each utterance is accumulated in a word pool during interaction. A new content word in an utterance implies this feature.

initiative change: A heuristic algorithm of initiative change detection using predefined cue expressions and information of the speaker suggests this feature.

back-channel: When two consecutive utterances by different speakers include back-channel cue expressions, and they do not include any content words, reciprocal back-channel is implied.

silence: A silence longer than 1 second between utterances implies this feature.

[1] The first utterance of a topic segment.

impression: Cue words suggesting impression and sentiment in an utterance implies this feature.

Detecting and predicting topic shifts To detect topic shifts, we use *clue expression*, *new words*, *initiative change* and their combinations. When at least one of them is detected in an utterance, it is considered as a topic shift utterance.

To predict topic shifts, we use *back-channel*, *silence*, *impression* and their combinations. Since these features tend to be observed just before topic shifts, when at least one of them is detected in an utterance, we predict a topic shift after this utterance. We call this in-between point *topic shift relevance place (TSRP).*

Table 3: Results of TSU detection

Combination of features	Prec.	Recall	F
clue expression	0.43	0.10	0.17
new words	0.09	0.31	0.14
initiative change	0.39	0.10	0.16
clue expression+new words	0.10	0.38	0.16
clue expression+initiative change	0.41	0.21	**0.27**
initiative change+new words	0.10	0.35	0.15
clue exp.+init. chg.+new words	0.11	0.41	0.17

Table 4: Results of TSRP detection

Combination of features	Prec.	Recall	F
back-channel	0.25	0.59	**0.35**
silence	0.23	0.31	0.26
impression	0.24	0.35	0.28
back-channel+silence	0.24	0.76	**0.36**
back-channel+impression	0.25	0.69	**0.36**
silence+impression	0.23	0.48	0.31
back-channel+silence+impression	0.23	**0.79**	**0.36**

Evaluation The held-out data of 20 dialogues from *Mister O corpus* were manually divided into topic segments by three annotators. The average of pair-wise κ was 0.41. The outputs of the system with various combinations of features were compared with the manual annotation.

We used different gold standards in calculating precision and recall. All three annotators should agree for the gold standard in calculating recall, while a single annotator is enough in calculating precision. When a topic shift is found within two succeeding utterances after a TSRP, that TSRP is considered correct (Reynar, 1994).

Table 3 and 4 show the result of the evaluation using the *Mister O corpus*. F measure is calculated by $F = 2PR/(P + R)$.

Table 3 shows that the combination of *clue expression* and *initiative change* provides the best performance, although it is still worse than the past work as (Arguello and Rosé, 2006). What is interesting is that their recall is poor separately, but combining them doubles the value.

Table 4 shows that combinations including *back-channel* give rise to good F-measure values, suggesting that reciprocal back-channel is a good clue of an ending topic. Recall of detecting TSRPs is fairly good in contrast to precision.

Conclusion Based on the analysis of a free conversation corpus, we proposed a method of detecting topic shifts and topic shift relevance places in information seeking chats. The algorithm was evaluated in comparison with human performance. Although there is much room for improvement, we obtained initial clues for managing topic shifts in real-time information seeking chats. Future work includes sophisticated feature detection modules at the same time as devising a method to select an appropriate new topic at a topic shift.

Acknowledgement This research used the *Mister O corpus* which was built under the project on "Empirical and theoretical studies on culture, interaction, and language in Asia" (No. 15320054, directed by Prof. Ide Sachiko at Japan Women's University) supported by the *Grant-in-Aid for Scientific Research* program from the *Japanese Society for the Promotion of Science (JSPS).*

References

J. Arguello and C. Rosé. 2006. Museli: A multi-source evidence integration approach to topic segmentation of spontaneous dialogue. In *HLT/NAACL*, pages 9–12.

M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing. 2003. Discourse segmentation of multi-party conversation. In *ACL*.

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

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

R. Ochiai, Y. Nomura, and K. Ueno. 2005. Overview of Mister O corpus. In *9th International Pragmatics Conference*. In Panel: Exploring the relationship among culture, language and interaction: Cross-linguistic perspectives.

A. Olney and Z. Cai. 2005. An orthonormal basis for topic segmentation in tutorial dialogue. In *HLT/EMNLP*, pages 971–978.

J. C. Reynar. 1994. An automatic method of finding topic boundaries. In *Proc. of ACL*, pages 331–333.

K. Ries. 2001. Segmenting conversations by topic, initiative, and style. In *SIGIR Workshop: Information Retrieval Techniques for Speech Applications*, pages 51–66.

M. Stede and D. Schlangen. 2004. Information-seeking chat: Dialogue management by topic structure. In *Proc. CATALOG '04*, pages 117–124.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 175–176.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Collaboration in Peer Learning Dialogues

Cynthia Kersey and **Barbara Di Eugenio**
Computer Science, University of Illinois at Chicago, USA
{ckerse2,bdieugen}@uic.edu

Pamela Jordan and **Sandra Katz**
Learning Research and Development Center, University of Pittsburgh, USA
{pjordan,katz+}pitt.edu

Abstract

Our project seeks to enhance understanding of collaboration in peer learning dialogues, to develop computational models of peer collaborations, and to create an artificial agent, KSC-PaL, that can collaborate with a human peer via natural language dialogue. We present some initial results from our analysis of this type of dialogues.

1 Introduction

Peer tutoring and collaboration strongly promote learning (Cohen et al., 1982; Rekrut, 1992; van Boxtel et al., 2000); however, there are no models of collaboration in dialogue that can fully explain why collaboration between peers engenders learning for all the peers involved more than other learning situations, even when one peer is more "expert" than the other. There is general consensus that working together encourages students to generate new ideas that would probably not occur to them if working alone; mechanisms that support such exchanges include co-construction (Hausmann et al., 2004) and knowledge sharing (Soller, 2004). We will refer to all these mechanisms as KSC, or "Knowledge Sharing and Construction". To contribute to an increased understanding of peer learning, we have started to apply our *balance-propose-dispose* model of negotiation (Di Eugenio et al., 2000) to this type of learning dialogues. In that model, partners first balance their knowledge distributions, then propose a possible next step and lastly decide to commit to a proposal or postpone it in order to further balance the knowledge needed for problem solving. We expect this model will be affected by (a) the knowledge distribution, (b) a collaborator's estimates of what types of knowledge the partner has, (c) decisions on what knowledge to share and (d) the detection of proposals and of problem solving or collaboration impasses. The initial model was based on the Coconut dialogues, collected in a setting where the task was simple (furnishing a two room apartment) and knowledge was equally distributed. Our new domain is the fundamentals of data structures and algorithms in Computer Science, and the task is finding conceptual mistakes in simple code. Not only is knowledge much more complex, but it is of different kinds – e.g., one collaborator may know (more) about null pointers and the other about loops.

In this poster, we briefly outline some preliminary results from our data collection.

2 Collaborating on Data Structures Tasks

We have developed a set of data structures tasks for peers to solve and pre/post tests to measure whether the interaction is beneficial (a beneficial collaboration is one in which at least one student learns); we pilot tested both in a face to face setting; we then proceeded to collect data in a computer mediated environment. The specific task is debugging or explaining easy routines for fundamental data structures such as linked lists, stacks and binary search trees. We are interested in *conceptual*, not *syntactic* mistakes, and we inform our subjects of this.

We have chosen a computer mediated environment to more closely mimic the situation a student will have to face when interacting with KSC-Pal, the artificial peer agent we intend to develop based on our *balance-propose-dispose* model and our empirical findings. In addition, in (Di Eugenio et

```
14:01:56 C:  unless the "first" is just a dummy node
14:02:20 D:  i don’t think so because it isn’t depicted
             as a node in the diagram
14:02:28 C:  OK
14:03:13 C:  so you would draw something like...
14:03:24 D:  i believe it will make the list go like this:
             bat, ant, cat
14:03:40 C:  draw: add pointer second (n100)
14:03:44 C:  draw: move n100
14:03:46 C:  draw: link n100 to
14:03:47 C:  draw: link n100 to n002
```

Figure 1: An excerpt from one of our dialogues

al., 2000), we had shown that such a setting affects the length of turns and turn taking, but does not change the nature of collaboration. Our computer-mediated environment supports typed natural language dialogue, task-specific drawing tools and menu-based code mark-up. These features were based in part on observations on the face to face interactions: the peers frequently drew data structures and deictically referred to the code they were diagnosing or explaining. In addition they collaboratively marked up the code under discussion.

We have collected dialogues using the computer mediated interface for 12 pairs thus far. Each dyad was presented with 5 exercises and all but two solved all 5 exercises. Figure 1 shows a short excerpt from one dialogue. Note that it includes drawing actions in addition to verbal exchanges.

These dialogues differ from the face-to-face dialogues collected in the pilot study in that the dyads appear to be more focused when using the computer-mediated environment. There is only a small amount of off-topic chat compared with the face-to-face dialogues. Also, there is less hedging and hesitation in making problem-solving suggestions. The drawing appeared to be more purposeful as well, although this could be the result of the constraints of the drawing tool instead of the environment itself. Interestingly for our *balance-propose-dispose* model, proposals can be conveyed by drawing, as in Figure 1. C. announces he will propose a solution at 14:03:13, and then proceeds to draw it starting at 14:03:40. We have observed at least 5 instances in which a problem solving proposal was made by drawing in our dialogues. In addition, the drawing tool wasn’t consistently used by the dyads. We have analyzed in more detail 18 of the debugging dialogues, i.e., 9 dyads each solving exercise 3 on linked lists and 4 on stacks. 7 dyads (78%) drew something for problem 3, but only 4 dyads (44%) did for problem 4; two of the four dyads use the tool just once to place a single object on the screen. This could be related to the nature of the problem since exercise 3 involved linked-lists which are generally believed to be more confusing than stacks.

Another interesting observation was that 8 of the 18 dialogues do not appear to follow a recursive, stack-based dialogue structure (Rosé et al., 1995). In these 8 dialogues, the dyads separately identify the errors in the programs and then return later to discuss and correct them. However, the topics were not revisited according to recency of mention but by the order in which problems were identified. Additionally, the dyads occasionally revisit errors, not to reopen the discussion, but rather to reaffirm corrections that have already been made. Not only does this not follow the recursive, stack-based dialogue structure, it also creates difficulties in identifying the point of disposition.

References

P. A. Cohen, J. A. Kulik, and C. C. Kulik. 1982. Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal,* 19(2):237–248.

Barbara Di Eugenio, Pamela W. Jordan, Richmond H. Thomason, and Johanna D. Moore. 2000. The agreement process: An empirical investigation of human-human computer-mediated collaborative dialogues. *International Journal of Human Computer Studies*, 53(6):1017–1076, December.

Robert G. M. Hausmann, Michelene T.H. Chi, and Marguerite Roy. 2004. Learning from collaborative problem solving: An analysis of three hypothesized mechanisms. In *COGSCI05, 26th Annual Conference of the Cognitive Science Society*, Chicago, IL.

M.D. Rekrut. 1992. Teaching to learn: Cross-age tutoring to enhance strategy instruction. In *American Educational Research Association,* San Francisco, CA.

Carolyn Penstein Rosé, Barbara Di Eugenio, Lori Levin, and Carol Van Ess-Dykema. 1995. Discourse processing of dialogues with multiple threads. In *ACL95, Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics,* pages 31–38.

Amy Soller. 2004. Understanding knowledge-sharing breakdowns: a meeting of the quantitative and qualitative minds. *Journal of Computer Assisted Learning,* 20(3):213–223.

C. van Boxtel, J. van der Linden, and G. Kanselaar. 2000. Collaborative learning tasks and the elaboration of conceptual knowledge. *Learning and Instruction*, 10(311–330).

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 177–178.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

The BoB IQA System: a Domain Expert's Perspective

Manuel Kirschner
KRDB Center, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
kirschner@inf.unibz.it

Introduction

We present BoB, a multilingual Interactive Question Answering system we have been developing to be deployed on the web-site of our university library. While being rather simplistic regarding the underlying theories of language and dialogue, it is an adequate baseline system that could be developed in around one year's time, and is easy to tailor and maintain by library domain experts. In this paper, we describe the current development version of BoB from a domain expert's perspective, giving an overview of the ways in which they can enhance the system, and what tools they use to make these modifications. With the tools presented here, our domain experts can autonomously extend BoB's knowledge base with new question topics, dialogue features and additional languages. Currently, BoB is running (although not publicly accessible) and can be tested in the German version; also, the tools described here have been fully implemented and are regularly used.

1 BoB's Search Algorithm

Like typical "chatterbots", BoB uses a stimulus-response loop for mapping a user utterance to some corresponding "canned-text" response. Answering a user question thus becomes a problem of retrieving the best response. The mapping from user input to system response is done on the basis of regular expression patterns; for every system response, there is a pattern that is supposed to match a specific class of user input. In BoB, these patterns and the corresponding system answers are stored in pairs. Unlike in most chatterbots that have no representation of state, these pairs are stored in a focus tree that represents the dialogue context. In the course of a dialogue, the current topic switches between the focus nodes of the tree, depending on what regular expression patterns the current user utterance matches, and on the previously active node. In this simple model, the current focus node thus represents the dialogue state.

2 Jump-starting the Focus Tree

Through a cooperation with the library of the University of Hamburg, we acquired the knowledge base of Stella, a relatively sophisticated German chatterbot application for the library domain[1]. Our main goal for using these data was to extract a part of the encoded library application-specific information. We jump-started the creation of BoB's focus tree by extracting the regular expression patterns and corresponding system responses (both for German), as well as the topic hierarchy in which these pairs were organized. In this way, we got a topic hierarchy consisting of 230 topics[2], containing an overall of over 2000 pairs of regular expression patterns and system responses.

There are two problems with re-using the data from Stella in our project. First, many of the topics are unique to the University of Hamburg library and have to be removed from BoB's focus tree, while other topics are obviously missing; a similar problem concerns the regular expression patterns that often contain Hamburg-specific parts that have to be changed. The second problem is related to the seemingly ad hoc way that the Stella topic hierarchy is organized. In practice, this makes it potentially difficult for the domain experts to decide under which topic nodes to insert new pattern-response pairs. While in principle a topic hierarchy could be used as an interesting model for tracking dialogue focus, only few of the existing regular expression patterns in Stella actually happen to require a specific location within the hierarchy structure (these are patterns for underspecified user questions that can only be inter-

[1] http://www.sub.uni-hamburg.de/informationen/projekte/infoass.html

[2] Examples from the 22 main topics: library buildings, organization, services, catalog query, books, journals, topics, articles, lending, inter-library loan, web site.

preted by knowing the dialogue context: *context-dependent follow-up questions*). So far, our domain experts are using the Stella topic hierarchy to cluster and organize the pattern-response pairs; in the absence of concise criteria for how the hierarchy should be built, they are free to add, move and delete topics as they see fit. In practice, the situation in which BoB's search algorithm critically depends on the tree structure is when faced with a context-dependent follow-up question: in this case, the algorithm begins by searching focus nodes with a "follow-up" tag (see below) among the children of the last active node.

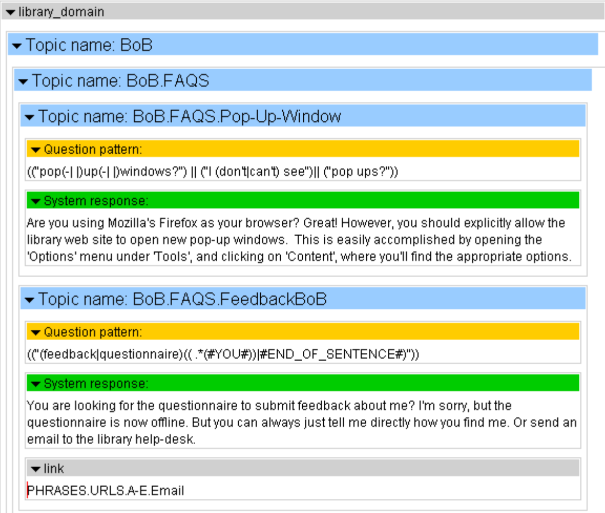
3 Controlling BoB's Dialogue Features

As mentioned above, one of BoB's dialogue-related features is its ability to handle context-dependent follow-up questions. This requires the domain expert to foresee possible ways in which users might follow up on a topic encoded in BoB's hierarchy, and add new focus nodes (with the "follow-up" tag) as children of the respective topic node. We are currently exploring principled ways in which to support the domain expert in this task (so that catering for follow-ups becomes less dependent on human intuition).

Another dialogue feature of BoB which is under the domain expert's control are sub-dialogues. They are used to implement system-initiated clarification requests, or more generally, to guide the user through the library domain via system initiative. In the first case, an ambiguous user question would trigger a clarification sub-dialogue, while in the second case, some system response might provide the user with a choice of related topics about which the user could then ask in more detail. Technically, both cases are implemented with sub-dialogues, by assigning special "link" elements to certain nodes in the tree. If in a dialogue BoB reaches a node with a "link" element, the normal tree search algorithm is suspended and resumes processing only after moving to the sub-dialogue node referenced in the link element.

4 Tools for the Domain Experts

The central tool used by the domain experts to edit BoB's knowledge base is a free, off-the-shelf XML editor[3] in conjunction with three style sheets for providing different views of the focus tree. This is the information that the respective views convey: (i) the BoB topic hierarchy, with fields for temporarily deactivating certain topic nodes; (ii) the focus nodes in detail, including regular expression pattern, system response, and possibly a "link" element (cf. Fig. 1); (iii) for each focus node, the (German) regular expression patterns and system responses along with fields for the Italian and English translations (used by the domain experts in charge of BoB's translation).



(("pop(-| |)up(-| |)windows?") || ("I (don't|can't) see")|| ("pop ups?"))

Are you using Mozilla's Firefox as your browser? Great! However, you should explicitly allow the library web site to open new pop-up windows. This is easily accomplished by opening the 'Options' menu under 'Tools', and clicking on 'Content', where you'll find the appropriate options.

(("(feedback|questionnaire)((.*(#YOU#))|#END_OF_SENTENCE#)"))

You are looking for the questionnaire to submit feedback about me? I'm sorry, but the questionnaire is now offline. But you can always just tell me directly how you find me. Or send an email to the library help-desk.

PHRASES.URLS.A-E.Email

Figure 1: "Focus node" view of the BoB focus tree

The "follow-up" tag is realized as an XML attribute that the domain expert can assign to any focus node, using the XML editor's standard methodology for changing attributes. Besides the customized XML editor, the domain experts also use a tool for checking BoB's extended regular expression syntax for correctness and matching against possible user questions. Additionally, they constantly test their latest changes to the focus tree on their own development version of BoB.

5 Future Work

We are exploring several ways of adding more intelligence to BoB; we are currently concentrating on follow-up questions, and how to accommodate them in the focus tree in a more systematic way. For the domain experts, this will mean less "guessing" possible user questions, since a theory should be able to automatically predict likely follow-ups. Also, once BoB goes public, we will collect user log files, which could further guide the domain experts in adjusting the focus tree. The data will also prove useful for validating our approach to IQA in general.

[3] XMLMind Standard Edition, `http://www.xmlmind.com/`

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 179–180.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

AMBROSIO: THE MIMUS TALKING HEAD

Pilar Manchón, Antonio Ávila, David Ávila, Guillermo Pérez, Gabriel Amores
University of Seville
{pmanchon, aavila, davila, gperez, jgabriel}@us.es

ABSTRACT

In a technology-driven world where HCI is developing fast and the research on Multimodal Dialogue Systems is substantial, personality endowed virtual characters are acquiring importance. Socio-psychological research [4] indicates that users are in general more willing to interact with technology when the latter is 'humanized', that is, when the interaction is closer to that between humans. In this paper, the integration of a talking head in MIMUS, a home-control multimodal dialogue system, is presented. Also known as 'Ambrosio', the MIMUS talking head in synchronized with state-of-the-art synthesizers and provides a full range of facial expressions and motions. A preliminary evaluation of the talking head impact on the overall system perception seems to confirm the benefits of integrating virtual characters in this type of systems.

***Index Terms—Multimodal** dialogue systems and **interfaces, HCI,** human factors, talking heads, personality,*

1. INTRODUCTION

During the last couple of decades at least, a great deal of research around Multimodal Dialogue Systems is being conducted. Among other interesting issues, the importance of introducing virtual entities to interact with users has been analysed from different perspectives.

Some authors advocate for the benefits of human-like interaction, endowing virtual characters with human characteristics to make human-machine interaction as close as possible as human-human interaction [4]. Other authors however are reluctant to believe that simulating human communication is the best alternative. Moreover, they express serious concerns about negative social and cognitive issues, pointing at a rather dramatic clash of the human-computer social order in cognitive terms [9].

In the development of the MIMUS system, we have opted for the human-centered design approach and implemented a talking head with the ability to talk, change its facial expressions and perform some motions. The overall purpose of the MIMUS system is to become a practical and valuable tool in the smart home scenario, and more in particular, an everyday tool for the specific focus group the data was gathered from in a set of WoZ experiments [10]. It is therefore understandable that with that objective in mind, different sub--objectives gain importance: human--like interaction must be not only efficient, but may and/or should also include additional human features. In order to endow the system with sufficient capabilities to fulfill these requirements, the MIMUS system has been furnished with the hereby described talking head that complements the system's personality, and confers an appearance of human—like communication on the interaction.

2. THE MIMUS SYSTEM

MIMUS is a multimodal dialogue system for the control of a smart home. It relies of a flexible architecture that allows for the integration of multiple input and output modalities. The current design and configuration responds to the requirements of the selected focus group of users (wheel-chair bound users), although there is no reason why it could not be reconfigured for different user profiles. As a matter of fact, one of the main advantages of the flexile architecture above mentioned is the possibility of configuring the system's behavior in terms of the information available at user profile level.

As in [5], MIMUS offers a pseudo-symmetric architecture: graphical and voice modalities are available both at input and output, although written text is provided as output but it is not a current input option. Nonetheless, MIMUS offers additional advantages since any functionality can be achieved by mixing modalities, using only voice, or only graphically. This is particularly important to allow for different user profiles to take full advantage of the system, especially those with special needs.

MIMUS consists of a set of collaborative OAA agents. It consists of an ISU-based dialogue manager [6], a knowledge manager, a device manager, ASR and TTS managers, several graphical agents and the talking head. This Talking Head complements previous implementations, and endows the system with additional features and communicative intensity.

The literature [4] illustrates how different experiments show that computers are indeed social actors, and also that the users' conduct is quite different when interacting with a virtual character as opposed to when they interact with a faceless computer. The overall user satisfaction is greater when interacting with a virtual character. MIMUS seeks to be a clear example of user--centered design, and with the user always in mind, the MIMUS talking head has been integrated into the main system architecture. For more information about the overall system architecture, please see [7].

3. AMBROSIO'S DESIGN AND IMPLEMENTATION

Endowing the character with a name has a manifold purpose: Personalization (users can give him a name of their choice), Personification (they will address the system at personal level, reinforcing the sense of human--like communication) and Voice activation (Ambrosio will remain inactive until called for duty).

Ambrosio has been implemented in 3D to allow for more natural and realistic gestures and movements. The graphical engine used is OGRE, a powerful, free and easy to use tool.

3.1. 3D FACIAL ANIMATION

Modeling: The modeling methodology chosen is based on the facial muscular structure [2], which determines the basic modeling lines and areas that will in turn allow for the generation of facial expressions.
Expressiveness: In a 3D real--time application, facial expressions are generated by means of pre--defined poses. Once each expression is modeled, the 3D vertex variation for each pose is recorded separately, so that several expressions can be simultaneously generated. This is a widely used method called lineal interpolation animation.
Animation: It is achieved throughout a skeleton system: each bone has an impact on the neighboring vertexes. Each vertex has an associated list where each bone has an associated value ranging from 0 to 1.
Texture: For picture-like realism, the light has been integrated in the texture in order to achieve better performance and good graphical quality.

3.2. ARCHITECTURE & EXPRESIVENESS

As in [3], the system consists of four different subsystems: Input, Synchronization, Speech synthesis and Face management. The current talking head is integrated with Loquendo, which allows for lip synchronization.

According to the literature [1], gestures and expressions are almost or quite as important as speech itself: it reinforces the overall communicative act. It is therefore mandatory to determine the different behavior layers, and establish which gestures are conscious and which unconscious. Ambrosio’s conscious motions are nodding and shaking; his unconscious motions are breathing and blinking. His expressions coincide mostly with the standard [1] (happiness, sadness, anger, fear and surprise). However, “disgust” has been substituted by “doubt”, being the latter more useful in the dialogue context.

4. CONCLUSIONS & FUTURE WORK

As conclude in previous research [4] and our own experiments, a human-like talking head has a significant positive impact on the subjects' perception and willingness to use MIMUS. Although no formal evaluation of the system has taken place yet, MIMUS has already been presented successfully in different forums, and as expected, ``Ambrosio" has always made quite an impression, making the system more appealing to use and approachable. In the opinion of the potential users enquired, Ambrosio’s motions and expressions were helpful and communicative. Ambrosio’s desing and architecture are the result of the integration of different theoretical and practical approaches to avatars, personality and dialogue, all applied to a smart home system. Its flexible architecture will allow for very interesting extensions in the future such as user-taylored personalities. The final objective is to generate a range of different users’ profiles, detect the user’s personality direct or indirectly, and establish their direct mapping with compatible virtual characters that will match the users’ personalities and preferences.

Future developments necessarily entail a formal overall system evaluation, but also a specific human factors and usability evaluation of Ambrosio.

5. ACKNOWLEDGEMENTS

This work was carried out under the “TALK” research project, funded by EU´s FP6 [ref. 507802] and the “Generación de Lenguaje Natural para Sistemas de Diálogo” project, funded by the Spanish Ministry of Education under grant TIN2006-14433-C02-02.

6. REFERENCES

[1] Fischer, A, Ostermann J., Beutnagel M. and Wang, Y. Integration of talking heads and text.to.speech synthesizers fot visual tts. In International Conference of Speech and Language Processing, Sydney, Australia, 1998. ICSLP98.

[2] Moreaux, A. Anatomie Artistique de l'Homme. Maloine, Paris, 2nd edition, 1990.

[3] Gattass, M. .Lucena P. S. and Velho L.. Expressive talking heads: A study on speech and facial expression in virtual characters. Scientia, 13(2):1.12, 2002.

[4] Reeves, B. and Nass. C. The Media Equation. CSLI.Cambridge University Press, 1996.

[5] Wahlster, W. “SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell”, in Proceedings of the HCI Status Conference, Berlin, Germany, pp. 47-62., 2003

[6] Traum, D. Bos, J., Cooper, R. Larsson, S. Lewin, I. Matheson, C. and Poesio, M.. A model of Dialogue Moves and Information State Revision. Technical Report D2.1, Trindi Project. 1999.

[7] Pérez, G., Amores G. and Manchón, P.. A Multimodal Architecture for Home Control by Disabled Users. (2006) Proceedings of IEEE/ACL Workshop on Spoken Language Technology (SLT), Aruba. December 2006.

[8] P. Manchón, D. Ávila & A. Ávila. Modality-specific Resources: Extension. Deliverable 3.3 Extension. TALK Project. December 2006.

[9] Schmidt, C. T. A. (2005). Of Robots and Believing. Minds and Machines. Kluwer Academic Publishers, Hingham, MA, USA, 15: 195-205.

[10] P. Manchón, C. del Solar, G. Amores & G. Pérez (2006) "The MIMUS Corpus." LREC 2006 International Workshop on Multimodal Corpora From Multimodal Behaviour Theories to Usable Models. Genoa, Italy, pp. 56-59

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 181–182.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

SEMI-AUTOMATED TESTING OF REAL WORLD APPLICATIONS IN NON-MENU-BASED DIALOGUE SYSTEMS

Pilar Manchón, Guillermo Pérez, Gabriel Amores, Jesús González
University of Seville
{pmanchon, gperez, jgabriel, jesusgm}@us.es

ABSTRACT

Real-world dialogue systems as opposed to demo systems need in-depth logical testing to ensure robustness. This may indeed be a cumbersome task when dealing with non-menu-based dialogue systems, since the number of possible combinations is unmanageable. In this paper, a new logical testing methodology is described. Its main objective is to reach a manageable compromise between coverage and feasibility, in order to ensure robustness while keeping the amount of testing down to an affordable level. Since the number of test cases grows exponentially as applications become more complex and industry-oriented, it is fundamental to device a methodology to determine which cases should be tested and what level of robustness is to be expected with such amount of testing.

Index Terms— User Interface, Testing, Computer interface human factors.

1. INTRODUCTION

One of the main challenges of real-world applications as opposed to most showcase or research applications is the in-depth logical testing and evaluation of the application design and implementation. Proof-of-concept systems, whose main purpose is to proof and demo a specific set of strategies and/or functionalities in fully controlled environments, do not require the level of robustness of real world applications; therefore, they do not really entail so exhaustive an evaluation as applications which will be "out in the open", exposed to users and circumstances far from laboratory conditions.

Although it is widely agreed in the literature that menu-based systems imply significant drawbacks with respect to more sophisticated non-menu based systems, it is quite evident that the former do present a very important advantage with respect to the latter: predictable and manageable logical testing.

When it comes to non-menu-based systems, the scenario changes dramatically: this approach has a very positive impact in the flexibility and naturalness of the dialogue, and a very negative impact in the amount of time and resources that must be invested on each application to ensure robustness.

The same flexibility and naturalness that makes these systems more appealing to use originates the testing problems: any possible combination of events is allowed and no formal main dialogue flow is defined. It is true that there is usually a conceptual main flow that seems more likely or ideal. Nonetheless, it is a much more subjective notion than that of finite state-based or frame-based systems.

In this paper, the focus will be placed on Information State Update based systems (ISU-based) [1]. These systems consist of an information state, a formal representation of the information state, a set of dialogue moves, a set of update rules and an update strategy. Some ISU-based dialogue systems are Godis [2], Dipper [3] or Delfos NCL[4], the latter being the base system for the development of this methodology. Delfos NCL has been designed and implemented to deal with Natural Command Language Dialogues.

An ISU-based system can work with several Dialogue Moves within the same turn (e.g. "switch on the light and open the door") which can be theoretically infinite. Furthermore, these systems do not behave as finite-state automata: given a current dialogue phase and a new utterance, the next phase is not univocally determined: it also depends on the context (dialogue history). These two factors make the universe of possibilities infinite in two dimensions: by the number of Dialogue Moves per utterance, and by the number of utterances per dialogue.

Even though it is sensible to assume that some restrinctions on both dimensions will not affect dramatically on the system performance, the figures are still unmaneageble.

3. OBJECTIVES

The overall objective of this methodology is the formalization of a reliable testing procedure in the described environment that will ensure a reasonable degree of robustness. For this purpose, several issues must be taken into account: the methodology must be semi-automated, must allow for several testers to work simultaneously, must generate a pre-deployment Logical Flow Score (***LFS***), must determine the precise set of test cases to be used and must take into account all special natural language dialogue phenomena.

4. THE COVERAGE- FEASIBIITY TRADE-OFF

One of the main challenges here is the determination of the precise set of test cases that will ensure a high LF-Score. In Delfos, there are several configuration files that contain all the relevant information to define a new Natural Command Language application: a natural language grammar, a lexicon and the dialogue rule specification. Given that the information in these files is insufficient to undertake the task at hand, additional information must be generated: a. The dialogue "***hot zone***", which is somewhat similar to the dialogue flow of a finite-state or frame-based dialogue system, but defining a set of possibilities; b. The list of natural language dialogue phenomena handled by the system.

4.1. Dialogue Rule Unit-Testing

The formal representation of the information state in Delfos is the DTAC structure, which is a set of attribute-value pairs: DMOVE, generic type of dialogue move, TYPE, specific type of dialogue move, ARGS, complementing arguments to complete the dialogue move, linked by logical operators and

CONT, the actual content of the move. Each DTAC in the grammar triggers a rule in the dialogue management specification file. All possible triggering scenarios for each dialogue rule must be generated. This is equivalent to software unit-testing since rules are tested in isolation. The result is a full list of high level grammar productions that must be tested. It must be notice that this is quite different from just listing all the grammar productions in the grammar file, since the correct grammatical parsing does not guarantee the appropriate system behavior. Once the tester has gone through all these productions, the first testing phase will have been completed, ensuring the correct system behavior inside each independent dialogue rule.

4.2. Inter-Rule Testing

The second testing phase will necessarily entail the correct system inter-rule behavior, which means ensuring that the logical dialogue flow involving different rules in any order is also correct. In order to accomplish this, a hand-made matrix of possibilities granting scoring the likelihood of the first DMove being followed by the second DMove has been generated. Given that matrix, let us define the right terminology:

$$P(path) = \prod_{\forall arch} P(arch)$$

$$W(graph \mid depth = N) = \sum_{\forall (path|depth=N)} P(path)$$

$$LFS(path \mid depth = N) = \frac{P(path)}{W(graph \mid depth = N)}$$

$$LFS(K \mid depth = N) = \frac{\sum_{\forall path \in K} P(path)}{W(graph \mid depth = N)}$$

Where $P(path)$ is the probability of a path within a graph, $W(graph \mid depth = N)$ is a graph weight given a maximum path depth = N, $LFS(path \mid depth = N)$ is the Logical Flow Score given a maximum path depth = N, and K a given set of paths:

From the matrix and by means of the algorithm, an ordered set of test cases will be obtained. Given the full set of cases, the above-mentioned formulae can then be applied in two different ways: to determine the ***LFS*** (Logical Flow Score) that can be achieved by testing the top ***X*** percentage of the full set, or to determine the percentage of the ordered set of cases that must be taken into account in order to achieve a fixed ***LFS***. In either case it is quite clear that the testing will be thorough and will achieve the intended degree of robustness, while minimizing the testing effort.

The process however does not end here. This methodology also enables us to compare the baseline hand-made matrix with real data collected once the application is deployed. A corpus of real user interactions with the system will make it possible to generate a new matrix that will be compared to the baseline matrix. As more and more applications are developed, tested, launched and then tuned after deployment, more and more corpora of cases will be collected, which will in time provide a measurement of the average proximity of the hand-made matrixes to the real ones. This of course will allow even further tuning in the test case generation process.

Testing a complex natural language application is usually a hairy and expensive issue; however, by optimizing the testing procedure we can ensure a very high level of robustness, an optimal use of resources and most likely, a significant reduction in testing costs.

In addition to the sets of test cases generated in phases 1 and 2, an additional number of random cases will also be selected in order to ensure the appropriate system behavior, even in rather odd or unpredictable circumstances. This set will be randomly selected from the remaining percentage of potential test cases.

5. CONCLUSIONS & FUTURE WORK

This methodology relays therefore in two main milestones: defining by hand the "***hot zone***" for the most likely flow/s to prioritize their exhaustive testing, and defining the properties and restrictions of the algorithm that will generate the testing scripts from the matrix to ensure a finite and valid number of cases. It also guarantees a well-defined level of testing that will include the full "**hot zone**", i.e., the most likely paths or flows the users will go through, allow for the test case distribution among an unrestricted number of testers, minimize the human error by providing an unambiguous methodology that can easily be followed, generate metrics to compare, learn and improve the testing procedure in subsequent cycles, optimize the amount of testing to be carried out y relation with the application size and complexity and facilitate the post-deployment tuning of the application, reduce de testing costs and therefore the overall application development costs.

The methodology hereby described represents a significant improvement with respect to previous situations with loosely defined or completely undefined methodologies. However, there is yet a considerable amount of work to be done by hand at this point. Future work must necessarily involve the automation of a number of human tasks, and the formalization of some of those tasks, such as the manual generation of probabilities for the baseline matrix.

7. ACKNOWLEDGEMENTS

This work was carried out under the "TALK" research project, funded by EU´s FP6 [ref. 507802].

8. REFERENCES

[1] Amores & Quesada, "Dialogue Moves in Natural Command Languages," *SIRIDUS Deliverable D1.1*, September 2000.

[1] Berstel et al., "A Scalable Formal Method for Design and Automatic Checking of User Interfaces," *ACM Transactions on Software Engineering and Methodology, Vol 14, No 2, Apri*l 2005.

[3] Hagerer et al., "Efficient Regression Testing of CTI-Systems: Testing a complex Call-Center Solution", *Annual Review of Communication Vol 55*, Int. Engineering Consortium (IEC), 2001.

[4] Larsson et al., "Evaluation of Contribution of the Information State Based View of Dialogue," *SIRIDUS Deliverable D3.4*, October 2002.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 183–184.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Enhancing System Communication through Synthetic Characters

Elena Not, Koray Balci, Fabio Pianesi and Massimo Zancanaro
Bruno Kessler Foundation (FBK-Irst)
Via Sommarive,18
38050 Povo-Trento, Italy
{not,balci,pianesi,zancana}@itc.it

Abstract

Synthetic characters are an effective modality to convey messages to the user, provide visual feedback about the system internal understanding of the communication, and engage the user in the dialogue through emotional involvement. We propose SMIL-AGENT as a representation and scripting language for synthetic characters, which abstracts away from the specific implementation and context of use of the character. SMIL-AGENT has been defined starting from SMIL 0.1 standard specification and aims at providing a high-level standardized language for presentations by different synthetic agents within diverse communication and application contexts.

1 Introduction

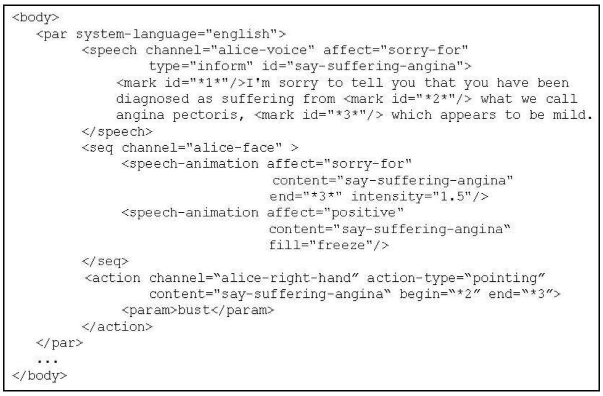
Synthetic characters are often integrated in multimodal interfaces as an effective modality to convey messages to the user. They provide visual feedback about the system internal understanding of the communication and engage the user in the dialogue through emotional involvement. However, avatars should not be considered as an indivisible modality, but as the synergic contribution of different communication channels that, properly synchronized, generate an overall communication performance: characters can emit voice and sounds, animate speech with lips and facial expressions, move eyes and body parts to realize gestures, express emotions, perform actions, sign a message for a deaf companion, display listening or thinking postures, and so on.

In this paper, we present SMIL-AGENT, a representation and scripting language that is intended to play as a sort of SMIL dialect for the specification of information presentations by a synthetic agent (Not et al., 2005).

2 SMIL-AGENT

With respect to other existing scripting languages (e.g., CML and AML (Arafa et al., 2004), APML (De Carolis et al., 2002), MPML), SMIL-AGENT pushes further the idea of having a separate representation for the various communication modalities of a synthetic character (e.g., voice, speech animation, sign animation, facial expressions, gestures,…) and their explicit interleaving in the presentation performance. Furthermore, SMIL-AGENT explicitly abstracts away from all data related to the dialogue management and the integration of the agent within larger multimodal presentations, thus assuring the portability of the language (and of the synthetic characters supporting it) to different task and application contexts.

The SMIL-AGENT formalism is certainly less compact and less discourse-oriented than, for example APML (see figure 1 for a sample presentation script written in SMIL-AGENT).



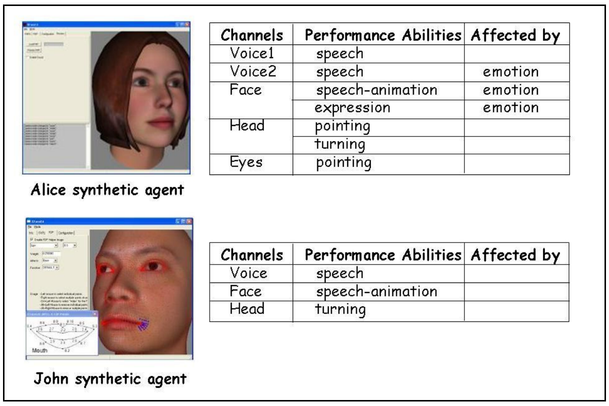
```
<body>
   <par system-language="english">
         <speech channel="alice-voice" affect="sorry-for"
                 type="inform" id="say-suffering-angina">
             <mark id="*1*"/>I'm sorry to tell you that you have been
             diagnosed as suffering from <mark id="*2*"/> what we call
             angina pectoris, <mark id="*3*"/> which appears to be mild.
         </speech>
         <seq channel="alice-face" >
              <speech-animation affect="sorry-for"
                                 content="say-suffering-angina"
                                end="*3*" intensity="1.5"/>
              <speech-animation affect="positive"
                                content="say-suffering-angina“
                                fill="freeze"/>
         </seq>
          <action channel=“alice-right-hand” action-type=“pointing”
                 content="say-suffering-angina“ begin=“*2” end=“*3”>
              <param>bust</param>
         </action>
   </par>
   ...
</body>
```

Figure 1. Sample SMIL-AGENT script

As an advantage, it allows plenty of flexibility in expressing which channel should realize a certain performance directive. For example, given a synthetic agent with sophisticated control of body motion and various channels corresponding to different body parts, a pointing in the direction of the character's bust could be realized by a hand, a finger, a hand plus the head in synchronization, etc…, according to which channel is specified in the <action> element of the script (in the script in Figure 1, for example, the agent's right hand is used). Furthermore, alternative voices could be easily selected at different stages of the presentation, or the face could support a wider set of emotions than voice.

2.1 Extending the language

SMIL-AGENT can easily be used with synthetic characters with different levels of sophistication (figure 2 shows two sample synthetic faces with different communication abilities[1]). The language formal syntax specification defines a separate language partition of attribute values that can be extended by expert authors to list the actual communicative channels and performance abilities supported by a certain synthetic agent. In practice, this is realized by a separate dtd file collecting the list of possible values for: (i) available types of *communicative channels* (e.g., voice, face, eyes, mouth, body, arm,…); (ii) supported *performance abilities* (e.g., verbal and animation abilities, emotions, speech acts, actions, languages); (iii) features that can be tested to include optional parts in the scripts.



Channels	Performance Abilities	Affected by
Voice1	speech	
Voice2	speech	emotion
Face	speech-animation	emotion
	expression	emotion
Head	pointing	
	turning	
Eyes	pointing	

Channels	Performance Abilities	Affected by
Voice	speech	
Face	speech-animation	
Head	turning	

Figure 2. Sample agents with different communication abilities

2.2 Playing scripts

At our institute, a SMIL-AGENT player has been implemented for MPEG-4 based synthetic faces which support: speech, speech-animation, affective facial expressions, gestures, head movements. As shown in Figure 3, for the sake of modularity, the player includes a core processing submodule to which different synthesizers (to get different languages or voice quality) and facial animation players can be plugged in (in the figure, the XfacePlayer and LUCIA[2] players are taken as examples). Visual speech, emotions and expressions are treated as separate channels where the timing is driven by the visual speech to be synchronized with the audio. For each channel, a sequence of morph targets (or FAPs) is created and then blended. An authoring tool for SMIL-AGENT scripts is also under development.

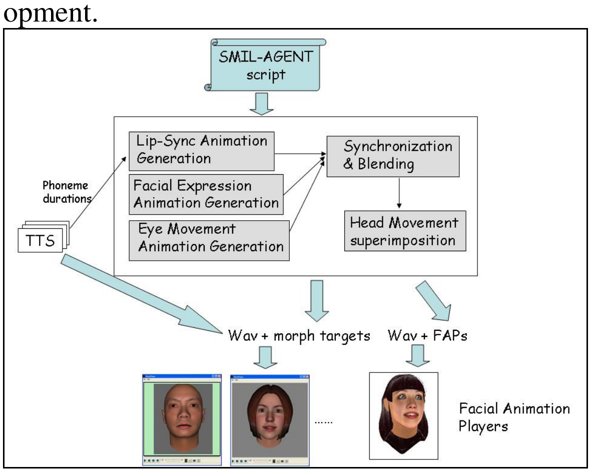

SMIL-AGENT script
Lip-Sync Animation Generation
Facial Expression Animation Generation
Eye Movement Animation Generation
Synchronization & Blending
Head Movement superimposition
Phoneme durations
TTS
Wav + morph targets
Wav + FAPs
……
Facial Animation Players

Figure 3. Sample processing of SMIL-AGENT scripts for MPEG-4 based synthetic faces

Reference

Arafa, Y., Kamyab, K. and Mamdani, E. Toward a Unified Scripting Language: Lessons Learned from Developing CML and AML. In H. Prendinger and M. Ishizuka (eds.) *Life-Like Characters. Tools, Affective Functions, and Applications*. Springer-Verlag, 2004, 39-63.

Balci, K. Xface: MPEG-4 based open source toolkit for 3d facial animation. In *Proceedings of AVI04, Working Conference on Advanced Visual Interfaces*, Gallipoli, Italy, 25-28 May, 2004.

De Carolis, B., Carofiglio, V., Bilvi, M. and Pelachaud, C. APML, a Markup Language for Believable Behavior Generation. In *Proceedings of the Workshop on "Embodied conversational agents – let's specify and evaluate them!"* at AAMAS02, 2002

Elena Not, Koray Balci, Fabio Pianesi and Massimo Zancanaro "Synthetic Characters as Multichannel Interfaces" In *Proceedings of ICMI05, Seventh International Conference on Multimodal Interfaces*, Trento, October 3-7, 2005

[1] These faces have been developed with Xface, a set of open tools for the creation of MPEG-4 based 3D Talking Heads (Balci, 2004)).

[2] http://www.pd.istc.cnr.it/LUCIA/home/default.htm.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 185–186.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

The LUNA Corpus: an Annotation Scheme for a Multi-domain Multi-lingual Dialogue Corpus

Christian Raymond◇, Giuseppe Riccardi◇, Kepa Joseba Rodríguez♣, Joanna Wisniewska♠
◇Department of Information and Communication. University of Trento.
{christian.raymond|riccardi}@dit.unitn.it
♣Piedmont Consortium for Information Systems (CSI-Piemonte)
KepaJoseba.Rodriguez@csi.it
♠Institute of Computer Science. Polish Academy of Science
jwisniewska@poczta.uw.edu.pl

Abstract

The LUNA corpus is a multi-domain multi-lingual dialogue corpus currently under development. The corpus will be annotated at multiple levels to include annotations of syntactic, semantic and discourse information and used to develop a robust natural spoken language understanding toolkit for multilingual dialogue services[1].

1 Introduction

LUNA is a project focused on the problem of real-time understanding of spontaneous speech in context of next generation dialogue systems[2].

Three steps will be considered for the Spoken Language Understanding (SLU) interpretation process: generation of semantic concept tags, semantic composition into conceptual structures and context-sensitive validation using information provided by the dialogue manager.

The SLU models will be trained and evaluated on the LUNA corpus and applied to different multilingual conversational systems in Italian, French and Polish.

The corpus is currently being collected with a target to collect 1000 human-human and 8100 human-machine dialogues in Italian, Polish and French. The dialogues will be collected in the following application domains: travel information and reservation, public transportation information, IT help desk, telecom customer care and financial information and transaction.

2 Segmentation and Transcription

The first step is the segmentation of the speech signal into dialogue turns. The turns will be annotated with time information, speaker identity and gender, and marked where speaker overlap occurs.

The next step is the transcription of the speech signal, using conventions for the orthographic transcription and for the annotation of non-linguistic acoustic events.

3 Multi-level annotation

Semantic interpretation involves several aspects, like the meaning of tokens referred to a domain or the relation between different semantic objects in the utterance and discourse level. In order to capture these different aspects we decided to implement a multi-dimensional annotation scheme. The annotation of some levels is mandatory for all the dialogues of the corpus. The annotation of the other levels is recommended.

The first levels of the annotation are related to the preparation of the corpus for the semantic annotation, and include segmentation of the speech signal in dialogue turns, transcription and syntactic pre-processing with Part of Speech (POS) tagging and shallow parsing.

The next level consist of the annotation of domain information using attribute value pairs. The annotation of this level is mandatory, as the annotation of the other levels depends on it.

The other levels of the annotation are the predicate structure, coreference and anaphoric relations and dialogue acts.

[1] This research was performed under LUNA project funded by the EC, DG Infso, Unit E1.

[2] The members of the consortium are: Piedmont Consortium for Information Systems (IT), University of Trento (IT), Loquendo SpA (IT), RWTH-Aachen (DE), University of Avignon (FR), France Telecom R&D Division S.A. (FR), Polish-Japanese Institute of Information Technology (PL) and the Institute for Computer Science of the Polish Academy of Sciences (PL). http://www.ist-luna.eu

4 POS-tagging and Chunking

The transcribed material will be annotated with POS tags, morphosyntaectic information and segmented based on syntactic constituency. For the POS-tags and morphosyntactic features, we will follow the recommendations made in EAGLES (EAGLES, 1996), which allows us to have a unified representation format for the corpus, independently of the tools used for each language.

5 Domain attribute level

Semantic segments are produced by concatenation of the semantic chunks. A semantic segment is a unit that corresponds unambiguously to a concept of the dictionary described bellow.

Semantic segments are annotated with attribute-value pairs following an approach similar to the used for the annotation of the French MEDIA corpus (Bonneau-Maynard and Rosset, 2003). We specify domain knowledge in domain ontologies that are used to build domain-specific concept dictionaries. Each dictionary contains:

- Concepts corresponding to classes of the ontology and attributes of the annotation.
- Values corresponding to the individuals of the domain.
- Constraints on the admissible values for each concept.

6 Predicate structure

For the annotation of predicate structure we decide to use a FRAMENET-like approach (Baker et al., 1998).

Based on the domain ontology, we define a set of frames for each domain. The frame elements are provided by the named entities, and for all the frames we introduce the negation as default frame element.

For the annotation first of all we annotate the entities with a frame and a frame element. If the target is overt realized we make a pointer from the frame element to the target. The next step is putting all the frame elements and the target (if overt realized) in a set.

7 Coreference

Coreference and anaphoric relations will be annotated in the LUNA corpus using an scheme close to the one used in ARRAU (Artstein and Poesio, 2006).

The first step is the annotation of the information status of the markables with the tags `given` and `new`. If the markables are annotated with `given` the annotator will select the most recent occurrence of the object and add a pointer to it. If the markable is annotated with `new`, we distinguish between markables that are related to a previously mentioned object, the so called associative references, or don't have such a relation.

If there is more that a unique interpretation, the annotator can annotate the markable as `ambiguous` and add a pointer to each of the possible antecedents.

8 Dialogue acts

In order to associate the intentions of the speaker with the propositional content of the utterances, the segmentation of the dialogue turns in utterances is based on the annotation of predicate structure. Each set of frame elements will be correspond with a utterance.

We use a multi-dimensional annotation scheme partially based on the DAMSL scheme (Allen and Core, 1997) and on the proposals of ICSI-MRDA (Dhillon et al., 2004). We have selected nine dialogue acts from the DAMSL scheme as initial tagset, that can be extended for the different application domains. Each utterance will be annotated with as many tags as applicable.

References

J. Allen and M. Core. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers.

R. Artstein and M. Poesio, 2006. *ARRAU Annotation Manual (TRAINS dialogues)*. Univerity of Essex, U.K.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of COLING-ACL*. Association for Computational Linguistics.

H. Bonneau-Maynard and S. Rosset. 2003. A semantic representation for spoken dialogues. In *Proceedings of Eurospeech*, Geneva.

R. Dhillon, S. Bhagat, H. Carvez, and E. Shriberg. 2004. Meeting Recorder Project: Dialog Act Labeling Guide. Technical report, TR-04-002 ICSI.

EAGLES. 1996. Recomendations for the Morphosyntactic Annotation of Corpora. EAGLES Document EAG-TCWG-MAC/R.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 187–188.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Adapting Combinatory Categorial Grammars in a Framework for Health Care Dialogue Systems

Lina M. Rojas-Barahona
Dipartimento di Informatica e Sistemistica, University of Pavia
Via Ferrata 1, 27100
Pavia, Italy
linamaria.rojas@unipv.it

1 Abstract

Dialogue systems have been extensively used to provide access to computer-based applications and services in several domains. Particularly, in the medical domain dialogue systems have been adopted as a wide reaching solution to complement traditional contact channels and have been used successfully(Young et al., 2001; Giorgino et al., 2004; Beveridge and Fox, 2006). Several studies have discussed the advantages of adopting dialogue systems for chronic symptoms monitoring, interviews, counselling, education, etc. (Migneault et al., 2006). Nevertheless, a widely diffused adoption of dialogue systems in the medical domain is still far from reality because of domain complexity and speech technology costs. Health dialogues have an additional complexity because they must confront social and relational issues through continuity over multiple interactions with patients, as well as, criticality in cases of chronic-disease management (Bickmore and Giorgino, 2006).

VoiceXML emerged as way to provide a standard solution to voice applications. However, most of the VoiceXML-based dialogues are system-driven because of VoiceXML shortcomings in supporting dynamic natural language processing (NLP) and discourse phenomena features (Mittendorfer et al., 2002). An extension of VoiceXML to support NLP in dialogues and to overcome its limitations is described in (Hataoka et al., 2004). However, a big effort should still be done in VoiceXML-generative frameworks to support the complex ontologies, guidelines and structured enquiry data collection tasks of the medical domain.

We present a platform for health-care dialogue deployment and for the incremental incorporation of well defined formalisms such as Combinatory Categorial Grammars (CCG) and enables the generation of different backends such as VoiceXML. This work is especially targeted to the health care context, where a framework for easy deployment of more "natural" dialogues could improve patient's perception of dialogues and allow a more widespread adoption of dialogue systems.

1.1 Proposed Approach

We developed a framework for easy dialogue development, motivated by our experience in building and validating a dialogue system for hypertensive patient home management HOMEY(Giorgino et al., 2004). AdaRTE (Adaptive Dialog and Runtime Engine) arises from this experience and the idea of offering a framework for efficient deployment of dialogue solutions in terms of time, cost, development effort and maintainability. This framework is mainly composed of an interpreter, a runtime-engine and an interface media realizer for backend generation (figure 1). AdaRTE is an extensible architecture for dialogue interpretation and representation which supports different backend formats (HTML and VoiceXML) and allows an easy implementation of external resources access such as databases or ontologies.

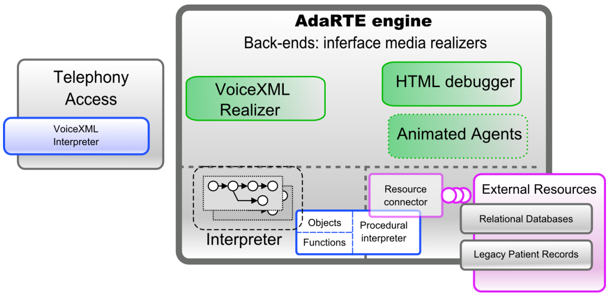


Figure 1: AdaRTE architecture

The AdaRTE framework has been beta-tested with two realistic health care dialogue systems. The first one is based on a prototype based on the TLC-COPD dialogue deployed in the past by the Boston MISU group(Young et al., 2001). This pilot's deployment demanded less than two weeks of man effort, is executed in English and uses DTMF interaction. The second test case is the partial reimplementation of the Homey dialogue system for the management of hypertensive patients. Re-engineering the system from the original proprietary dialogue manager (DM) to the AdaRTE architecture took approximately three weeks (eleven days of man effort), this system uses speech inputs and is executed in Italian. This framework yielded an important reduction of the time invested in developing these two prototypes whilst facilitating component reuse in each dialog.

Despite this time optimization, the generated dialogues are user-restrictive, that is to say, the user expressivity was extremely limited because of the restrictive grammar formats (CFG) supported by Voice Browsers. The semantic analysis supported by the VoiceXML standard is limited to complex ECMAScript objects processing. In order to support discourse phenomena features and mixed-initiative, the Voice Browser ASR should be extended to support either NLP-based grammar formalisms or n-grams.

We pursue the development of a framework that uses an ASR enriched with probabilistic grammars, NLP application and a flexible DM. The adoption of NLP in our framework allows us to support flexibility in dialogues. Thus, we chose the NLP library OpenCCG[1], which is based on CCG and MMCCG (Steedman, 2000; Baldridge and Kruijff, 2001). OpenCCG has been successfully used in two european projects (Foster and White, 2005; Wilske and Kruijff, 2006).

The strongest advantage of CCG is that it assigns categories enriched with meaning to expressions. Thus, a common-understanding in dialogues could be modelled by taking advantage of this meaning representation, together with the ontologies underlying the medical domain. Since the knowledge handled in the medical domain is complex, a tool for unification of meaning could provide a better representation of the domain and dialogue knowledge. Currently, we are making progress in the construction of an Italian grammar for the hypertensive patients management by using OpenCCG. In this grammar we references ontological definitions. Nevetheless, still a big effort should be done in order to adapt typical linguistic phenomena of romance languages such as Italian and Spanish.

1.2 Discussion and Future Work

We have presented an architecture for dialog representation and interpretation in which we built an engine for dialogue deployment. AdaRTE supports high-level dialog representations and supports VoiceXML generation as one of the generation backends. Even through this framework reduces the time invested in developing dialogue systems, we have found the dialogues being system-initiative.

This approach pursue building not only a reliable platform for health-care dialog deployment, but also a framework for the incremental incorporation of alternative formalisms in order to support features of discourse phenomena and best practices. For instance, we are working in adapt the CCG formalism, which allows a wide-lexicon that increase user expressivity in dialogues, by integrating the NLP library OpenCCG. In addition, we are working in the adoption of common understanding by using complex ontologies that describe the dialogue and the medical domain.

[1] http://openccg.sourceforge.net

References

Jason Baldridge and Geert-Jan M. Kruijff. 2001. Coupling ccg and hybrid logic dependency semantics. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 319–326, Morristown, NJ, USA. Association for Computational Linguistics.

Martin Beveridge and John Fox. 2006. Automatic generation of spoken dialogue from medical plans and ontologies. *J. of Biomedical Informatics*, 39(5):482–499.

Timothy Bickmore and Toni Giorgino. 2006. Health dialog systems for patients and consumers. *J. of Biomedical Informatics*, 39(5):556–571.

M.E. Foster and M. White. 2005. Assessing the impact of adaptive generation in the comic multimodal dialogue system. In *Proceedings of IJCAI-05 Workshop on the Knowledge and Reasoning in Practical Dialogue Systems*.

T. Giorgino, Azzini I., C. Rognoni, S. Quaglini, M. Stefanelli, R. Gretter, and D. Falavigna. 2004. Automated spoken dialogue system for hypertensive patient home management. *International Journal of Medical Informatics*, 74(1386-5056):159–167, apr.

N. Hataoka, Y. Obuchi, T Mitamura, and E Nyberg. 2004. Robust specch dialog interface for car telematics service. *ieeecnf*, (10.1109/CCNC.2004.1286882):331 – 335, jan.

Jeffrey P. Migneault, Ramesh Farzanfar, Julie A. Wright, and Robert H. Friedman. 2006. How to write health dialog for a talking computer. *J. of Biomedical Informatics*, 39(5):468–481.

M. Mittendorfer, G. Niklfeld, and W. Winiwarter. 2002. Making the voice web smarter-integrating intelligent component technologies and VoiceXML. *ieeecnf*, 2(10.1109/WISE.2001.996736):126–131, dec.

Mark Steedman. 2000. *The syntactic process*. MIT Press, Cambridge, MA, USA.

Sabrina Wilske and Geert-Jan Kruijff. 2006. Service robots dealing with indirect speech acts. In *International Conference on Intelligent Robots and Systems*, pages 4698–4703.

M. Young, D. Sparrow, D. Gottlieb, A. Selim, and R. Friedman. 2001. A telephone-linked computer system for COPD care. *Chest*, 119(5):1565–1575, May.

Decalog 2007: Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue, pages 189–190.
Trento, Italy, 30 May – 1 June 2007. Edited by Ron Artstein and Laure Vieu.

Automatic Discourse Segmentation using Neural Networks

Rajen Subba
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois 60607
rsubba@cs.uic.edu

Barbara Di Eugenio
Department of Computer Science
University of Illinois at Chicago
Chicago, Illinois 60607
bdieugen@cs.uic.edu

Discourse segmentation is the task of determining minimal non-overlapping units of discourse called elementary discourse units (EDUs). It can be further subdivided into sentence segmentation and sentence-level discourse segmentation. This paper addresses the latter, more challenging subtask, which takes a sentence and outputs the EDUs for that particular sentence.

(1) Saturday, he amended his remarks to say that he would continue to abide by the cease-fire if the U.S. ends its financial support for the Contras.

(1a) Saturday, he amended his remarks
(1b) to say
(1c) that he would continue to abide by the cease-fire
(1d) if the U.S. ends its financial support for the Contras.

In example (1), a sentence from a Wall Street Journal article taken from the Penn TreeBank corpus is further segmented into four EDUs, (1a), (1b), (1c) and (1d) (RST, 2002). Discourse segmentation, clearly, is not as easy as sentence boundary detection. The lack of consensus with regards to what constitutes an elementary discourse unit adds to the difficulty. Building a rule based discourse segmenter can be a tedious task since these rules would have to be based on the underlying grammar of the particular parser that is to be used. Therefore, we adopted a neural network model for automatically building a discourse segmenter from an underlying corpus of segmented text. We chose to use part-of-speech tags, syntactic information, discourse cues and punctuation. Our ultimate goal is to build a discourse parser that uses this discourse segmenter.

The data that we used to train and test our discourse segmenter is the RST-DT (RST, 2002) corpus. The corpus contains 385 Wall Street Journal articles from the Penn Treebank. The training set consists of 347 articles for a total of 6132 sentences, whilst the test set contains 38 articles for a total of 991 sentences. The RST-DT corpus provides us with pairs of sentences and EDUs. For the syntactic structure of the sentences, we have used both the gold standard Penn Treebank data and syntactic parse trees generated by (Charniak, 2000). As regards the discourse cues, we used a list of 168 possible discourse markers.

Problem formulation Like (Soricut and Marcu, 2003), we formulate the discourse segmentation task as a binary classification problem of deciding whether to insert a segment boundary after each word in the sentence. Our examples are vectors that provide information on POS tags, discourse cues and the syntactic structure of the surrounding context for each word in the sentence. The categories that we decided to use in our vector representation for each example are given in table 1. We used binary encoding of the values for each category in order to convert them into numeric values and compress our data. For all the 12 categories, we needed a total of 84 bits. After processing our data we obtained about 140,000 examples (vectors) to train the model. Each vector also indicated whether a segment boundary followed that particular word or not. We used a Multi-Layer Perceptron. The weights of the network were initialized using a random uniform distribution. Back-Propagation was used to update the weights. Each training run was limited to 50 iterations. We trained both a single model and a bagged model.

No.	category type
1	Prev. word POS
2	Prev. word Next Label
3	Prev. word Parent
4	Cur. word POS
5	Cur. word Parent
6	Next word POS
7	Next word Next Label
8	Next word Parent
9	Common ancestor CFG Rule for Cur. word and Next word
10	Cur. word CFG Non-Terminal
11	Next word CFG Non-Terminal
12	Is Next word a Discourse Cue ?

Table 1: Categories used for training the model.

Experiments and Results We evaluate our discourse segmenter against the test set of 38 articles with 991 sentences from the RST-DT corpus. We compare our results on the RST-DT test set with that of (Marcu, 2000) and (Soricut and Marcu, 2003). (Soricut and Marcu, 2003) used a probabilistic model (SynDS) and (Marcu, 2000) implemented a decision tree based model (DT). (Soricut and Marcu, 2003) measures the performance of the segmenter based on the it's ability to insert inside-sentence segment boundaries. Table 2 reports the results for the RST-DT test set for four systems using their metric. NNDS (Neural Network Discourse Segmenter) is our system. NNDS-B is the bagged model. SynDS is the best reported system that we are aware of. The results show that NNDS, a neural network based discourse segmenter can perform as well as SynDS. Bagging the model increases the performance of the segmenter. More importantly recall is higher since a bagged model is less sensitive to overfitting. The human segmentation performance as reported by (Soricut and Marcu, 2003) is 98.3% F-Score.

We also compare our system to (Huong et. al, 2004). (Huong et. al, 2004) is a symbolic implementation. Unlike (Soricut and Marcu, 2003), they used a flat-bracketing measure to compute performance. This measure accounts for both the start and end boundaries of a segment for precision and recall. They report an F-Score of 80.3% using the Penn TreeBank parsed trees. Our segmenter using bagging obtains a performace of 84.19% F-Score according to this measure. While our evaluation is based on the full test set of 38 articles, (Huong et. al, 2004) used only 8 articles for testing their symbolic segmenter.

System	Parse Tree	Precision	Recall	F-Score
DT	-	83.3	77.1	80.1
SynDS	C	83.5	82.7	83.1
SynDS	T	84.1	85.4	84.7
NNDS	C	83.66	80.17	82.03
NNDS	T	85.35	83.8	84.56
NNDS - B	C	83.94	84.89	84.41
NNDS - B	T	85.56	86.6	86.07

Table 2: Performance on the RST-DT corpus.
(Parse Tree: C - Charniak, T - Penn TreeBank)

Conclusion We have presented a connectionist approach to automatic discourse segmentation. Bagging the model yields even better performance. The performance of our discourse segmenter is comparable to the best discourse segmenter that has been reported. In the future, we intend to exploit additional features, namely lexical head features from the syntactic parse trees. We also plan to test our discourse segmenter on other discourse corpora, where segmentation decisions are based on a different coding scheme to test how well our model can generalize.

Acknowledgments

This work is supported by award IIS-0133123 from the National Science Foundation (NSF), and additionally by awards ALT-0536968 from NSF and N000140010640 from ONR.

References

Charniak, E. 2000. A maximum-entropy-inspired. In Proceedings of the NAACL 2000, pages 132139, Seattle, Washington, April 29 May 3.

Daniel Marcu. 2000. The Theory and Practice of Discourse Parsing and Summarization. The MIT Press, November 2000.

Huong Le Thanh, Geetha Abeysinghe and Christian Huyck. 2004. Automated Discourse Segmentation by Syntactic Information and Cue Phrases, In Proceedings of the IASTED, Innsbruck, Austria.

Radu Soricut and Daniel Marcu. 2003. Sentence Level Discourse Parsing using Syntactic and Lexical Information. In *Proceedings of the HLT/NAACL-2003*, Edmonton, Canada, May-June.

RST-DT. 2002. RST Discourse Treebank. Linguistic Data Consortium.

Author index