Dialor'05

Proceedings of the ninth workshop on the semantics and pragmatics of dialogue

9–11 June 2005
Loria

Claire Gardent and Bertrand Gaiffe
Editors

Foreword

Steered by an informal group of researchers, the SEMDIAL Workshops on the Semantics and Pragmatics of Dialogue aim at bringing together researchers working on the semantics and pragmatics of dialogues from different perspectives such as artificial intelligence, formal semantics and pragmatics, computational linguistics, and psychology.

As these proceedings clearly show, the initiative has become reality and DiaLor'05 is indeed a meeting where dialog will be discussed from many distinct perspectives.

Pure linguistics is well represented with in particular, several papers from the formal semantics tradition. Maria Aloni gives an interesting new account of the semantics and pragmatics of imperatives and of their effect on the occurrence of free choice "or" and "any" in imperative sentences; Henk Zeevat considers the issues raised by modals, corrections and attitudes in Discourse Representation Theory and Yafa Al-Raheb presents an extension to Speaker/Hearer representation in DRT. In the phonology field, a paper by Safarova, Muller and Prévot addresses the discourse functions of final rises in French dialogue. And several papers are concerned with empirical issues such as establishing a taxonomy of dialogue acts (Bunt and Girard), classifying dialogue acts in task oriented information seeking dialogues (Geertzen and Girard), analysing belief transfer in information dialogues (Morante and Bunt), classifying errors in human-robot communication based on a web based experiment (Gieselmann and Waibel) or annotating corpora to learn dialogue strategies (Georgila, Lemon and Henderson).

In the areas of Artifical Intelligence and Computational linguistics, the contributions reflect the evolution of the fields with emphasis on multimodality, multiagent systems, computer aided learning systems and the use of statistical techniques and of ontologies. Thus Sandewall, Lindblom and Husberg report on a multimodal system integrating videos, speech and a robot; Bringert, Cooper, Ljungloef and Ranta present a grammar adapted for multimodal processing and several posters/demos describe multimodal systems (Löckelt and Norbert Pfleger; Kruijff-Korbayová et al.; Manchón, Pérez and Amores) or multimodal annotation tools (Müller). Further afield, Ginzburg and Fernandez discuss the issues arising from modeling multilogue rather than dialogue and two contributions deal with the specific issues arising from dialogues in teaching systems (Slabbers and Knott; Michel and Lehuen). Two further contributions explore the use of machine learning and statistical techniques for classifying dialogue acts (Webb, Hepple and Wilks) and for turn taking modeling (Eliasson). Finally, several papers/posters un-

derline the need/utility of ontologies in dialogue processing (Romanelli, Backer and Alexandersson; Loos and Porzel) and Larsson addresses the question whether current research in dialogue modeling can be viewed as cognitive modeling or more as an engineering task.

Psychology is also present thanks to a contribution by Bard and colleagues exploring the psychological plausibility of various predictions made by the theories of common ground underlying most of dialogue theories.

The three fields (Linguistics, Computational linguistics and AI, psychology) are also represented by each of the three invited speakers namely, Justine CASSELL (Northwestern University, USA) who will talk about the verbal and non verbal behaviour of virtual agents; Gerhard JAEGER (University of Bielefeld, Germany) who will present a game theoretic account of implicatures in dialog and Arthur GRAESSER, (University of Memphis, USA) whose talk on AutoTutor lies at the intersection of discourse processing, cognitive science, computational linguistics and the learning sciences.

Following on MunDial'97 (Munich), Twendial'98 (Twente), Amstelogue'99 (Amsterdam), Gotalog'00 (Gothenburg), Bidialog'01 (Bielefeld), Edilog'02 (Edinburgh), Diabruck 2003 (Saarbruecken) and Catalog'04 (Barcelona), this year SEMDIAL workshop will be held in Nancy (France). "Dia" is the compulsory dia-log component, Nancy is in Lorraine, hence Dialor'05!

As always, the realisation of such a workshop has depended on the good will, voluntary work and financial support of many individuals and institutions.

I would like to thank the members of the programme committee for their speedy and competent reviews. In alphabetical order : Jan Alexandersson, Ellen Bard, Johan Bos, Francis Corblin, Matthew Crocker, Raquel Fernandez, Jonathan Ginzburg, Rodger Kibble, Alistair Knott, Ivanna Kruijff-Korbayová, Nicolas Maudet, Philippe Muller, Martin Pickering, Manfred Pinkal, Massimo Poesio, Hannes Rieser, Laurent Romary, Laurent Roussarie, Robert van Roy, David Traum, Mats Wirén, Enric Vallduvi and Henk Zeevat.

For their invaluable help concerning all organisational aspects of the conference, I am grateful to Bertrand Gaiffe who took over the role of local organisation chair at a very short notice and at a time where family matters were perhaps more pressing than a conference; to Yannick Parmentier for handling the website; to Armelle Demange for running the workshop administration and to Mathieu Quignard, Eric Kow, Jean-Marie Pierrel, Laurent Romary and Christine Fay-Varnier for accepting to be part of the local organisation committee.

The LORIA is providing the infrastructure (secretarial support, computers, conference rooms, publicity services, etc.) for the conference and I would like to

particulary thank its director, Hélène Kirchner for supporting DiaLor'05.

Finally, I would like to acknowledge the financial support of the following institutions : the CNRS (Centre National de la Recherche Française), the INRIA (Institut National pour la Recherche en Informatique et ses Applications), la Communauté Urbaine du Grand Nancy (CUGN), la Région Lorraine, le Conseil Général de Meurthe et Moselle, le LORIA, l'Université Nancy 2, l'Université Henri Poincaré et l'Institut National Polytechnique de Lorraine.

Claire Gardent
Dialor'05 Programme chair

Programme commitee

Claire Gardent (chair)

Jan Alexandersson	Ellen Bard	Johan Bos	Francis Corblin
Matthew Crocker	Raquel Fernandez	Jonathan Ginzburg	Rodger Kibble
Alistair Knott	Ivana Kruijff-Korbayova	Nicolas Maudet	Philippe Muller
Martin Pickering	Manfred Pinkal	Massimo Poesio	Hannes Rieser
Laurent Romary	Laurent Roussarie	Susanne Salmon-Alt	Robert Van Rooy
David Traum	Mats Wirén	Enric Vallduví	Henk Zeevat

Invited speakers

Justine CASSELL	Northwestern University (USA)
Gerhard JAEGER	University of Bielefeld (Germany)
Arthur GRAESSER	University of Memphis (USA)

Contents

Invited talks

Making (Virtual) Friends and Influencing (Virtual) People 1
Justine Cassel

Let's pretend to agree. A game theoretic reconstruction of M-implicatures 2
Gerhard Jaeger

AutoTutor: Learning while Holding a Conversation with a Computer 3
Arthur Graesser

Papers

Utility and implicatures of imperatives 5
Maria Aloni

A system for generating teaching initiatives in a computer-aided language learning dialogue 13
Nanda Slabbers Alistair Knott

What makes Human-Robot Dialogues struggle? 21
Petra Gieselmann , Alex Waibel,

Integrating a Discourse Model with a Learning Case-Based Reasoning System 29
Karolina Eliasson

Designing An Open, Multidimensional Dialogue Act Taxonomy 37
Harry Bunt, Yann Girard

Dialogue Systems: Simulations or Interfaces? 45
Staffan Larsson

Multimodal Dialogue System Grammars 53
Björn Bringert, Robin Cooper, Peter Ljungloef, Aarne Ranta

Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations 61
Kallirroi Georgila, Oliver Lemon, James Henderson

Integration of Live Video in a System for Natural Language Dialog with a Robot 69
Erik Sandewall, Hannes Lindblom, Bjoern Husberg

The Discourse Function of Final Rises in French Dialogues 77
Marie Safarova, Philippe Muller, Laurent Prévot

Action at a distance: the difference between dialogue and multilogue 85
Jonathan Ginzburg, Raquel Fernandez

Empirical determination of thresholds for optimal dialogue act classification . 93
Nick Webb, Mark Hepple, Yorick Wilks

On Plurals and Default Unification 101
Massimo Romanelli, Tilman Becker, Jan Alexandersson

Conditional anaphora 109
Henk Zeevat

Let's you do that: carrying the cognitive burdens of dialogue 115
E. G. Bard, A. H. Anderson, Y. Chena, H. Nicholson, C. Havard

Posters and demonstrations

Robust Semantic Interpretation and Dialog Management in the Context of a CALL Application 123
Johan Michel and Jérôme Lehuen

Extensions to Speaker/ Hearer Representation in DRT 127
Yafa Al-Raheb, University of East Anglia (UK)

WOZ Experiments in Multimodal Dialogue Systems 131
Pilar Manchón, Guillermo Pérez, Gabriel Amores

Micro-analysis of the belief transfer in information dialogues 135
Roser Morante, Harry Bunt

Multi-Party Interaction With Self-Contained Virtual Characters 139
Markus Löckelt, Norbert Pfleger

A new Metric for the Evaluation of Dialog Act Classification 143
Stephan Lesch, Thomas Kleinbauer, Jan Alexandersson

Automatic analysis of elliptic sentences in the Thetos system 147
Nina Suszczanska, Julia Romaniuk, Przemyslaw Szmal

Simplified MMAXQL: An Intuitive Query Language for Corpora with Annotations on Multiple Levels 151
Christoph Müller

Presentation Strategies for Flexible Multimodal Interaction with a Music Player 155
Ivana Kruijff-Korbayová, Nate Blaylock, Ciprian Gerstenberger, Verena Rieser, Tilman Becker, Michael Kaisser, Peter Poller, Jan Schehl

DJ GoDiS: Multimodal Menu-based Dialogue in an Asynchronous ISU System 159
Staffan Larsson, David Hjelm

Towards Ontology-based Pragmatic Analyses 163
Berenike Loos, Robert Porzel

Invited talk: Making (Virtual) Friends and Influencing (Virtual) People

Justine Cassell
Northwestern University (USA)

Abstract

Harmony or rapport between people is essential for relationships as diverse as seller-buyer and teacher-learner. In this talk I describe the kinds of discourse behaviors – such as common ground and other interactional structures and narrative resonance – and non-verbal behaviors– such as attention, positivity, and coordination – that function together to establish a sense of rapport between two people in conversation. These studies are used as the basis for the implementation of virtual peers - adults, but also more recently embodied conversational virtual children who are capable of acting as friends and learning partners with real children from different ethnic traditions, collaborating to tell stories from the child's own cultural context, and aiding children in making the transition between home and school language.

Invited talk:
Let's pretend to agree. A game theoretic reconstruction of M-implicatures

Gerhard Jaeger
University of Bielefeld (Germany)

Abstract

Levinson (2000) classifies conversational implicatures into Q-, I-, and M-implicatures. While the former two can straightforwardly be analysed as consequences of speaker economy and hearer economy, a derivation of M-implicatures (non-stereotypical meanings are expressed by complex expressions) from rational economy principles is less straightforward. Inspired by recent work of Stalnaker on game theoretic pragmatics, I will show in the talk how all three types of implicatures are predicted to arise in situations in which semantic conventions are common knowledge between the interlocutors, but following those conventions would not be rational.

Invited talk:
AutoTutor: Learning while Holding a Conversation with a Computer

Arthur Graesser
University of Memphis (USA)

Abstract

AutoTutor is a learning environment on the Internet that helps students learn by holding a conversation in natural language. The system integrates computational mechanisms that were inspired by the fields of discourse processing, cognitive science, computational linguistics, and the learning sciences. More specifically, AutoTutor's design was inspired by explanation-based constructivist theories of learning, intelligent tutoring systems that adaptively respond to student knowledge, and research on dialogue patterns in tutorial discourse. AutoTutor presents challenging questions on topics such as Newtonian qualitative physics or introductory computer literacy and then engages in mixed initiative dialogue that coaches the student in building an answer. It provides feedback to the student on what the student types in (positive, neutral, negative feedback), pumps the student for more information, prompts the student to fill in missing words, gives hints, fills in missing information with assertions, identifies and corrects erroneous ideas, answers the student's questions, and summarizes answers. The recent versions of AutoTutor attempt to adapt to learners' emotions and to guide the learner through discourse in interacting with 3D simulations.

Utility and implicatures of imperatives

Maria Aloni
ILLC/Department of Philosophy
University of Amsterdam
M.D.Aloni@uva.nl

Abstract

The article defines the relevance or utility of an imperative in terms of how far it can help in increasing the probability of the occurrence of a desirable future world. In terms of this notion, we account for (i) the potential of imperatives to license free choice *any* in their scope; and (ii) the free choice effects of disjunctive and *any*-imperatives.

1 Choice-offering imperatives

1.1 *Or* in imperatives

It is a well known fact that *or* in imperatives can give rise to a free choice effect, see (Ross, 1941; Åquist, 1965; Hamblin, 1987) and more recently (Aloni, 2003).

(1) $!(A \vee B) \Rightarrow \Diamond A \wedge \Diamond B$

As an illustration of (1), consider the following example:

(2) SMITH: Take her to Knightsbridge or Bond Street!

JONES STARTS TO LEAVE.

SMITH: (?) Don't you dare take her to Bond Street!

Intuitively the most natural interpretation of Smith's first imperative is as one presenting a choice between two different actions. Smith's subsequent imperative can be regarded as negating this choice, and, therefore, strikes us as out of place here.

The free choice inference in (1), however, is not always warranted as illustrated by the following example from Rescher and Robinson (1964):

(3) TEACHER: John, stop that foolishness or leave the room!

JOHN STARTS TO LEAVE.

TEACHER: Don't you dare leave this room!

Examples like (3) suggest to treat free choice effects as pragmatic implicatures, rather than semantic entailments. In the classical literature (notably (Åquist, 1965)), examples like (3) has been presented as evidence in favor of an ambiguity between choice-offering and alternative-presenting disjunctive imperatives. On a pragmatic approach, the failure of the free choice inference in example (3) can be explained as an implicature cancelation without multiplying the senses of imperative sentences.

A further indication that free choice effects of disjunctive imperatives are conversational implicatures is the fact that they disappear in negative environments (e.g. Gazdar 1979).

(4) Don't post this letter or burn it!

If free choice inferences had the status of logical entailment, then (4) could be used in a situation in which one wants the letter to be posted or burnt, but doesn't want to leave the choice to the hearer. This is clearly not so.

1.2 *Any* in imperatives

Another example of a 'choice-offering' imperative is (5) with an occurrence of free choice *any* which is licensed in this context.

(5) Take any card!

Like disjunctive imperatives, *any*-imperatives should be interpreted as carrying with them the inference that a choice is being offered.

(6) $!(any\ x\ \phi) \Rightarrow \forall x \Diamond \phi$

As in the case of disjunctive imperatives, the free choice effect in (6) disappears under negation. One needs a special stress to retain it, as in (8).[1]

(7) Don't take any card!

(8) Don't take just ANY card!

Contrary to disjunctive imperatives, however, in a positive environment, the inference in (6) is hard to cancel. Contrast (9) with (10).

(9) MARIA: Take any card!

YOU START TO TAKE A CARD.

MARIA: # Don't you dare take the ace!

(10) MARIA: Take a card!

YOU START TO TAKE A CARD.

MARIA: (?) Don't you dare take the ace!

Imagine a context in which it is well known that aces cannot be taken. In such a context, Maria's second imperative in (10) would be natural. In (9), however, it would be still out of place. By using *any*, in (9), rather than *a*, Maria conveys that no exceptions apply to her prescription: even aces must be permissible options.

This reduced tolerance of exceptions typical of uses of *any* has been discussed in (Kadmon and Landman, 1993). On their account, *any* has the effect of WIDENING the domain of quantification compared to a standard use of an indefinite noun phrase. Furthermore, domain widening should be for a reason. *Any* is licensed only in those cases where widening the domain is functional, i.e., leads to a STRENGTHENING of the statement made.

Domain widening and strengthening (defined in terms of *entailment*) explain the following distribution facts:

(11) a. John did not take any card.

$\neg\exists x\phi$

b. # John took any card.

$\exists x\phi$

Enlarging the domain of an existential in the scope of negation does create a stronger statement (example (11a)). In an episodic sentence, it doesn't (example (11b)).

It is easy to see, however, that this sort of explanation does not extend directly to non-declarative cases. Let us assume Groenendijk and Stokhof's (1984) notion of entailment for interrogatives, and the standard notion of entailment for imperatives defined in terms of inclusion of their compliance conditions.[2] Then, widening the domain of an existential in an interrogative or an imperative does not create a stronger sentence, still *any* is licensed in (12) and (13).

[1] The use of *any* illustrated in (8) have been called anti-indiscriminative in (Horn, 2000) and anti-depreciative in (Haspelmath, 1997). On the present account, sentences like (8) must be taken to involve a metalinguistic use of negation.

[2] Imperative I entails I' iff each way of complying with I is a way of complying with I'. See e.g. (Hamblin, 1987).

(12) Did John take any card? $?\exists x\phi$

(13) Take any card! $!\exists x\phi$

To explain (12), (van Rooij, 2003) proposed to interpret strength in terms of *relevance* rather than entailment, and provided a perspicuous characterization of the relevance of a question in terms of the decision theoretic notion of *expected utility*.

In this article I would like to extend van Rooij's (2003) proposal to imperatives. In order to do this, I will define a notion of the relevance or utility of an imperative in a context as a function of the probability of its compliance and its desirability. According to this notion, in example (13), domain widening can lead to an interpretation with a higher expected utility because it can increase the probability of a positive response from the hearer. In this sense, I would like to suggest, imperatives meet Kadmon and Landman's requirement that domain widening should be functional. Intuitively, by enlarging the domain of an existential quantifier in an imperative the speaker indicates that she will be pleased by more ways of complying with her wishes. This increases her chances that the hearer will comply. Note that domain widening increases utility only in a situation in which no element in the enlarged domain is ruled out as an option. This allows us to derive from (13) the permission to take any card as an implicature. Since *any* can be used only in situations where domain widening increases utility, this explains why this implicature is hard to cancel. Since existential sentences can be seen as generalized disjunctive sentences, the free choice implicatures of disjuntive imperatives follow by the same reasoning. In this case, however, these implicatures can be canceled, like in Rescher & Robinson's example where the implicated material was in conflict with shared assumptions in the common ground.

2 Expected utility of imperatives

In this section I define the expected utility value of an imperative I in terms of how far I can help in increasing the probability of the occurrence of a desirable future world. Expected utility values will be calculated with respect to a state representing the speaker's beliefs and desires about the future.

2.1 States

A *state* σ is a pair (p, u) consisting of a probability function p on the set W of possible worlds and a utility function u.

The *probability* function p maps worlds to numbers in the interval $[0, 1]$, with the constraint that $\sum_{w\in W} p(w) = 1$. Probability distributions can be extended to subsets C of W as follows: $p(C) = \sum_{w\in C} p(w)$. In this context, a world represents a way in which things might turn out to be in the near future. The probability function p represents the belief of the agent with respect to the probability of the occurrence of a world w. The value $p(w)$ may depend on a number of factors, like physical possibility (relative to the laws of nature), temporal possibility (possible in the time), and, most important, active possibility (relative to the willingness of the other people to co-operate). If $p_\sigma(w) \neq 0$ we will say that w is possible in σ.

The *utility* function u is a mapping from W to the set $\{0, 1\}$ and expresses the desirability of a world w. Desirable worlds obtain value 1, undesirable worlds, value 0.

As an illustration of these notions, consider the following examples of a state (for simplicity we are considering only four worlds, where each world is indexed with the atomic propositions holding in it. For example, in w_q, only q holds, and in w, no atomic proposition holds):

(14) a.

	p	u
w_q	1/2	1
w_r	1/2	1
w_{qr}	0	0
w	0	0

b.

	p	u
w_q	0	1
w_r	3/4	0
w_{qr}	0	0
w	1/4	0

c.

	p	u
w_q	1/6	1
w_r	1/6	1
w_{qr}	0	0
w	2/3	0

In order to understand this notion it might be useful to ask ourself in which of these states one would rather be. Intuitively, (14a) is the best choice. Each world which is still possible there, is also desirable. State (14b) is the worst choice, none of the possible worlds is a desirable one. Finally, in (14c), which is probably the most realistic option, some of the possible worlds are desirable, some are not. The notion of the value of a state defined in the following paragraph is meant to capture these intuitions.

2.2 The value of a state

We can think of a state $\sigma = (p, u)$ as a degenerate decision problem in which the set of alternative actions has just one element. Following the standard notion of expected utility in Bayesian decision theory, I define the *value* of a state as follows:

(15) $V(\sigma) = \sum_{w \in W}(p_\sigma(w) \times u_\sigma(w))$

The value of a state σ expresses the probability in σ of the occurrence of a desirable world. A state with value 1 is one in which each possible world is also desirable, e.g. (14a) above. A state with value 0 is one in which none of the possible worlds are desirable, e.g (14b). More realistic states are those in which the value lies between 0 and 1, like (14c) above with value $(1/6 + 1/6) = 1/3$.

In order to increase the value of a state, an agent may do different things. She might change her desire or, better, she might act in order to change her probability function, for example, by using an imperative. Declaratives do not have the power to change the probability of a future world, imperatives do. The goal of a declarative is to update an information state. The goal of an imperative is to enlarge the chance of the occurrence of a desirable world.

In what follows I will characterize the expected utility of an imperative in a state σ in terms of how far it can help in increasing the value of σ. More precisely, the expected utility value of an imperative I will be defined in terms of the *utility value* and the *probability* of the proposition C_I expressing the *compliance conditions* of I.

2.3 Compliance conditions

Declaratives have truth conditions, interrogatives have answerhood conditions, imperatives have *compliance conditions*. Someone cannot be said to understand the meaning of an imperative I unless she recognizes what has to be true for the command (or request, advice, etc.) issued by an utterance of I to be complied with. I shall identify the compliance conditions $C_{!\phi}$ of imperative $!\phi$ with the proposition expressed by ϕ.[3] For example,

(16) I: 'Kill Bill!'

C_I: 'That the hearer kills Bill'

(17) I: 'Kill Bill or John!'

C_I: 'That the hearer kills Bill or John'

[3] But see (Mastop, 2005) or (Portner, 2004) who, among others, have argued that imperatives are better analyzed in terms of actions or properties rather than propositions.

2.4 Utility value of a proposition

Following (van Rooij, 2003), we define the *utility value* $UV(C, \sigma)$ of a proposition C in a state σ as the difference between the value of σ after updating with C and before updating with C, where updates are defined in terms of Bayesian conditionalizations.

(18) $UV(C, \sigma) = V(\sigma/C) - V(\sigma)$

where $\sigma/C = (p_C, u)$ and p_C is the old probability function p conditionalized on C, that is, for each world w:

(19) $p_C(w) = p(w \,\&\, C)/p(C)$

The utility value of a proposition C in a state σ expresses how much an update with C can enlarge the value of σ.[4]

As an illustration, let us calculate the utility value of the following three propositions in the state (14c) above.

(20) $q \vee r$, q, $\neg q$

In order to do this we need to update (14c) (rewritten as τ in (21)) with the propositions in (20) and calculate the value of the resulting states.

(21) τ

	p	u
w_q	1/6	1
w_r	1/6	1
w_{qr}	0	0
w	2/3	0

(22) a. $\tau/(q \vee r)$

	p	u
w_q	1/2	1
w_r	1/2	1
w_{qr}	0	0
w	0	0

b. τ/q

	p	u
w_q	1	1
w_r	0	1
w_{qr}	0	0
w	0	0

c. $\tau/\neg q$

	p	u
w_q	0	1
w_r	1/5	1
w_{qr}	0	0
w	4/5	0

States (22a) and (22b) have value 1. State (22c) has value $1/5$. Since $V(\tau) = 1/3$, we obtain for our three propositions the following utility values:

(23) a. $UV(q \vee r, \tau) = 1 - 1/3 = 2/3$

b. $UV(q, \tau) = 1 - 1/3 = 2/3$

c. $UV(\neg q, \tau) = 1/5 - 1/3 = -2/15$

We can now define the expected utility value of imperatives.

2.5 Expected utility of imperatives

The *expected utility value* of an imperative I is defined as the product of the utility value and the probability of its compliance conditions C_I.

(24) $EUV(I, \sigma) = UV(C_I, \sigma) \times p_\sigma(C_I)$

The expected utility of imperative I in σ depends not only on the utility value of C_I, $UV(C_I, \sigma)$, formalizing how much closer to your goal the imperative would lead you, if accepted, but also on the probability of its acceptance, $p_\sigma(C_I)$.

As an illustration consider again our state τ, with value $1/3$:

[4] This notion is different from the *value of sample information* of statistical decision theory, e.g. (Raiffa and Schlaifer, 1961).

τ

	p	u
w_q	1/6	1
w_r	1/6	1
w_{qr}	0	0
w	2/3	0

Suppose one wants to increase $V(\tau)$ by using an imperative. The notions defined above can help us in making predictions on which imperative one should choose. We have three reasonable options:

(25) a. $!q$ 'Post this letter!'

b. $!r$ 'Burn this letter!'

c. $!(q \vee r)$ 'Post this letter or burn it!'

To see which is the best choice let us calculate their expected utility. In order to do so we need to determine the utility values and the probabilities of the propositions expressing their compliance conditions, namely q, r, and $q \vee r$.

As we have already seen, these three propositions obtain equivalent utility values since updating τ with any of them leads to a state of value 1.

(26) a. $UV(q, \tau) = UV(r, \tau) = 2/3$

b. $UV(q \vee r, \tau) = 2/3$

The probabilities, however, of the three propositions crucially differ, giving for the three imperatives the following expected utilities:

(27) a. $EUV(!q, \tau) = 2/3 \times 1/6 = 1/9$

b. $EUV(!r, \tau) = 2/3 \times 1/6 = 1/9$

c. $EUV(!(q \vee r), \tau) = 2/3 \times 1/3 = 2/9$

Among the options which have the potential to maximally increase the value of τ, $!(q \vee r)$ is the one with the highest probability of being accepted. Therefore, $!(q \vee r)$ is recommended as the best choice in this case.

3 Applications

In this section we discuss two applications of the previously defined notions. The first application concerns the potential of imperatives to license free choice *any*. The second concerns the free choice effects of *or* and *any* imperatives.

3.1 *Any* in imperatives

The utility value of a disjunction $UV(A \vee B)$ can never be higher than the utility values of both its disjuncts.

(28) For *no* state σ:

$$UV(A \vee B, \sigma) > UV(A, \sigma), UV(B, \sigma)$$

In declaratives, disjunctions cannot increase relevance. The use of *or*, in declaratives, usually signals either lack of information (it is unknown which of the disjuncts is true) or lack of relevance (none of the disjuncts would be strictly more relevant).

In imperatives, however, disjunctions can be used to increase relevance. The example discussed in the previous section, has shown that the expected utility of a disjunctive imperative $EUV(!(A \vee B))$ can be higher than the expected utility value of any of its disjuncts:

(29) There is a state σ:

$$EUV(!(A \vee B), \sigma) > EUV(!A, \sigma) \;\&$$
$$EUV(!(A \vee B), \sigma) > EUV(!B, \sigma)$$

Since existential sentences can be treated as generalized disjunctions:

(30) $\exists x \phi \equiv \phi(a) \vee \phi(b) \vee \phi(c) \vee ...$

we can then conclude that domain widening can increase the relevance of an existential imperative ($!\exists x \phi$), but not of an existential declarative ($\exists x \phi$). This explains why *any* is licensed in (31a), while it is out in (31b).

(31) a. Take any card!

b. # John took any card.

In (31a), domain widening can increase relevance because it can increase the probability that the hearer will comply. In (31b), it cannot. The utility of a declarative is not a function of its probability.

With imperatives, but not with declaratives, a weaker option can be more relevant than a stronger alternative.

3.2 Free choice implicatures

On this account, free choice effects are derived as implicatures arising from the following Gricean reasoning (again for ease of exposition we only consider the case of disjunction):

(32) The speaker used $!(A \vee B)$ rather than the shorter $!A$ or $!B$. Why? $!A$ and $!B$ must have had a lower expected utility. A disjunctive imperative $!(A \vee B)$ has a higher expected utility than $!A$ and $!B$ only in a situation in which both disjuncts are allowed. Then A and B must both be allowed.

To formalize (32), I first define the following semantics for deontic $\Diamond$, to be read as 'It is allowed', and $\Box$, to be read as 'it is obligatory':

(i) $\sigma \models \Diamond\phi$ iff $\exists w : u(w) = 1 \;\&\; w \in [\phi]$;

(ii) $\sigma \models \Box\phi$ iff $\forall w : u(w) = 1 \Rightarrow w \in [\phi]$.

ϕ is allowed in σ iff there is at least one desirable world in σ in which ϕ is true. ϕ is obligatory in σ iff in each desirable world in σ, ϕ is true.

Building on ideas from (Schulz, 2003), I then define the *implicatures* of an imperative I as the sentences not entailed by I holding in all σ/I where σ is an optimal states for I.

(33) I implicates ϕ, $I \mid\!\approx \phi \;\Leftrightarrow$
$I \not\models \phi \;\&\; \forall \sigma \in opt(I) : \sigma/I \models \phi$

An optimal state for I is one in which I is the choice with highest expected utility among a set of alternatives.

(34) $opt(I) = \{\sigma \mid \forall I' \in alt(I) : EUV(I) > EUV(I')\}$

Now, it is easy to prove that a disjunctive imperative $!(\phi_1 \vee \phi_2)$ has a higher expected utility than any of its disjuncts $!\phi_i$ only in a state in which each ϕ_i is possible, $p([\phi_i]) \neq 0$, and allowed, $\exists w : u(w) = 1 \;\&\; w \in [\phi_i]$.

If we assume as set of alternatives for a disjunctive imperative $!(A \vee B)$, the set $\{!A, !B\}$, and for an existential imperative $!\exists_D x\phi$ the set $\{!(\exists_Z x\phi) \mid Z \subset D\}$, it then follows that choice-offering imperatives implicate that each alternative way of complying with them is allowed:

(35) a. $!(A \vee B) \mid\!\approx \Diamond A \wedge \Diamond B$

b. $!\exists x\phi \mid\!\approx \forall x \Diamond \phi$

On this account, all disjunctive and indefinite imperatives induce a free choice effect. Like all implicatures, this effect disappears in the scope of negation. As it is easy to see, reconstructing the optimal state for $!\neg(A \vee B)$ or $!\neg\exists x\phi$ does not yield any free choice inference. In the case of positive disjunctive or *a*-imperatives, free choice effects can be canceled depending on the circumstances of the utterance (examples (1), (3) and (10)). In the case of positive *any*-imperatives, free choice effects cannot be canceled. This fact can be explained if we assume that *any* is felicitous only in contexts in which domain widening is functional, i.e. it increases relevance. In a context in which not all elements in the enlarged domain are permitted options, domain widening would be unjustified and *any* would be infelicitous.

4 Conclusion

I have defined the expected utility of an imperative in terms of how far it can help in increasing the probability of the occurrence of

a desirable world. This notion has been then applied to explain: (i) the potential of imperatives to license *any* in their scope; and (ii) the free choice effects of disjunctive and *any*-imperatives.

Any is licensed in imperatives, because enlarging the domain of an existential quantifier in an imperative can increase its expected utility. In this sense, imperatives meet Kadmon and Landman's requirement that domain widening should be for a reason.

Free choice effects have been derived as implicatures defined in terms of what must hold in a state in order for the used imperative to have maximal expected utility in that state.

References

Maria Aloni. 2003. On choice-offering imperatives. In Paul Dekker and Robert van Rooij, editors, *Proceedings of the 14th Amsterdam Colloquium*. ILLC-University of Amsterdam.

Lennart Åquist. 1965. Choice-offering and alternative-presenting disjunctive commands. *Analysis*, 25:182–184.

Jeroen Groenendijk and Martin Stokhof. 1984. *Studies on the Semantics of Questions and the Pragmatics of Answers*. Ph.D. thesis, University of Amsterdam.

Charles L. Hamblin. 1987. *Imperatives*. Basil Blackwell.

Martin Haspelmath. 1997. *Indefinite Pronouns*. Oxford University Press, Oxford.

Lawrence Horn. 2000. Pick a theory (not just *any* theory): Indiscriminatives and the free-choice indefinite. In L. Horn and Y. Kato, editors, *Studies in Negation and Polarity*. Oxford U. Press.

Nirit Kadmon and Fred Landman. 1993. Any. *Linguistics and Philosophy*, 16:353–422.

Rosja Mastop. 2005. *What can you do? Imperative mood in semantic theory*. Ph.D. thesis, University of Amsterdam.

Paul Portner. 2004. The semantics of imperatives within a theory of clause types. In Kazuha Watanabe and Robert B. Young, editors, *Proceedings of Semantics and Linguistic Theory 14*, Ithaca, NY. CLC Publications.

Howard Raiffa and Robert Schlaifer. 1961. *Applied statistical decision theory*. MIT Press, Cambridge MA.

Nicholas Rescher and John Robinson. 1964. Can one infer commands from commands. *Analysis*, 24:176–179.

Alf Ross. 1941. Imperatives and logic. *Theoria*, 7:53–71.

Katrin Schulz. 2003. You may read it now or later. A case study on the paradox of free choice permission. Master thesis. University of Amsterdam.

Robert van Rooij. 2003. Negative polarity items in questions: Strength as relevance. *Journal of Semantics*, 20:239–273.

A system for generating teaching initiatives in a computer-aided language learning dialogue

Nanda Slabbers
Dept of Computer Science
University of Twente

Alistair Knott
Dept of Computer Science
University of Otago

Abstract

This paper describes an extension made to a bilingual human-machine dialogue system, to allow the system to take initiatives in a language-learning dialogue. When the user concedes the initiative to the system, the system generates a set of 'possible initiatives', and chooses the best of these based on a number of criteria. These criteria relate firstly to the *formal* goal of generating an initiative which is appropriate in the current context, and secondly to the *substantive* goal of teaching the student a set of targeted syntactic constructions.

1 Introduction

A system engaging in a dialogue with a user has to generate two quite different kinds of utterances: **responses** (such as acknowledgements, answers to questions, and clarification questions) and **initiatives** (such as assertions of new material, or new questions à propos of nothing). When we consider what is involved in these two kinds of utterance, there are some interesting differences. It is common to analyse the task of natural language generation (NLG) as a pipeline involving **content selection**, **sentence planning** and **syntactic realisation** (see e.g. Reiter, 1994). For the generation of responses, the task of content selection is normally simple; the burden of the work is in sentence planning and syntactic realisation. For instance, to generate answers to questions or clarification questions, we typically need to construct sentences whose syntax and semantics echo that of the sentence being responded to. For the generation of initiatives, on the other hand, content selection is a key process: the issue of 'what to say' is much less constrained for such utterances. In this paper, we describe a system for generating initiatives in a particular register of dialogue: **computer-aided language learning** (or **CALL**) dialogue. The main innovation in our system is its adaptation of some standard content-selection techniques from NLG (traditionally used to produce utterances in monologue) to the task of generating initiatives in such dialogues.

We will begin in Section 2 by surveying some existing systems which generate teaching initiatives. Section 3 describes the initiative module and its goals. Section 4 describes the dialogue system in which the initiative module is embedded and provides some

results.

2 Existing work in generating teaching initiatives

There has been a great deal written about the role of initiative in tutorial dialogue systems; see e.g. Haller and McRoy (1997). But comparatively little of this work has considered the situation where the topic being taught is a foreign language. A CALL dialogue need not resemble a tutorial interaction at all; in many cases, it simply looks like a (somewhat stilted) conversation between two speakers on a particular topic. Of course, either participant can also ask or answer explicit questions about the language being taught. But when the topic being taught is the language itself, simply advancing the conversation has educational merit in its own right. The initiatives made by the tutor thus have a dual function: to continue a natural-sounding conversation, and to do so in a way which scaffolds the student's current language learning.

Surprisingly, most dialogue systems specialising in language-learning do not focus on generating initiatives. The systems we have reviewed (e.g. Desmedt, 1995; Seneff *et al.*, 2004; Raux and Eskenazi, 2004) typically involve a scenario where the user has to accomplish some task, and in which therefore most initiatives come from the student. In these scenarios, it is hard for the student to learn by adapting utterances made by the teacher. In our system, we focus on more symmetrical dialogues where the student and tutor can make the same kinds of utterances (e.g. 'Where's your Mum from?'... 'Where's *your* Mum from?'). In these dialogues, the tutor's initiatives can provide the student with models of the constructions to be learned, as well as fleshing out the content of the dialogue. The question is: how to generate appropriate initiatives in such contexts? We believe that some standard content selection techniques from NLG can be usefully applied to the problem.

3 NLG content-selection methods for initiative generation

The process of content selection in NLG is typically defined in relation to two goals: firstly the **formal** goal of generating a coherent text, and secondly the **substantive** goal of achieving a certain effect on the hearer. If the text being generated is a monologue, the formal goal will be expressed in terms of a theory of discourse structure, such as RST or one of its many competitors. The substantive goal is typically expressed using the vocabulary of AI planning. In one common architecture for content selection (see e.g. Marcu, 1996; O'Donnell *et al.*, 2001), the process involves two passes. In the first pass, a large set of **candidate messages** is created, using heuristics designed to maximise the likelihood of achieving the system's formal and substantive goals. In the second pass, these candidate messages are evaluated more systematically, and the one which best achieves the goals is chosen to be generated. This more systematic process often involves 'look-ahead' to the sentence planning and realisation stages, so that the evaluation can take into account syntactic factors as well as semantic ones.

To adapt the model just outlined to the generation of initiatives in a CALL dialogue, we must first specify formal and substantive goals for the system, and then we must specify a procedure for generating and evaluating initiatives in relation to these goals. These topics will be considered in the remainder of this section.

3.1 Formal goal: dialogue coherence

The formal goal of our CALL system will be to maintain a coherent dialogue. Modelling di-

alogue coherence is hard; it is not possible to define coherence at the level of dialogue acts (e.g. 'a question begets an answer'), because in the general case, the semantic content of a dialogue act is as relevant as its type. Current models of dialogue coherence typically use some brand of update semantics to formalise different dialogue acts and to provide definitions of grounding, the relationship between questions and answers, and subdialogues (c.f. e.g. Traum *et al.*, 1999). However, while these complexities are necessary in order to constrain *response* dialogue acts, they do not seem so necessary for initiatives. If we restrict ourselves to contexts where an initiative must be taken, it seems possible to define a coherent dialogue move simply by enumerating the types of dialogue act which can be taken at this point. In our case, we introduce two special dialogue acts which can only be used to make initiatives: a **new assertion** (which we distinguish from assertions which provide the answers to questions) and a **new question** (which we distinguish from clarification questions and follow-up questions). For our CALL domain, we decompose new questions into **genuine questions** (which fill in gaps in the system's knowledge base) and **teaching questions** (which ask the student about information already in the common ground, to check whether it has been understood).

In addition to this restriction to particular dialogue act types, we posit two weaker formal criteria for initiatives. The first relates to the **topic** of the new utterance. We suggest there is a preference for initiatives which maintain the current topic of the dialogue. At some points topic *changes* may be preferable instead (especially when the dialogue is on the same topic for a long time), but we assume the student will change the topic when he wants to. We do not see topic continuity as essential for maintaining coherence, but certainly if there is no continuity, there are obligations to mark this textually in the utterance generated. In our model, the topic of an utterance is the set of individuals and predicates which it introduces, and the degree of topic continuity between two utterances is defined in terms of the overlap between the two relevant sets; see Slabbers (2005) for details. There is also a higher-order preference for **strategic** initiatives, which move onto a topic which the system knows a lot about. The system is configured to prefer assertions which introduce topics which appear frequently in the its private knowledge base of facts. The second weaker criterion relates to the **mix** of dialogue acts; we suggest there is a preference for interleaving dialogue acts of different types, rather than producing several acts of the same type. Dialogue act mix is a **global** constraint on dialogue coherence (in the sense of Hovy, 1988; Piwek and van Deemter, 2003) but nonetheless it is one which we can try and optimise locally. In cases where several candidate utterances score equally as regards topic continuity, we can give preference to those which realise dialogue acts which have not been recently used.

3.2 Substantive goal: language-learning

In a CALL dialogue, any initiative made by the system should further its goal of teaching the student the language. Since our dialogue system creates complete syntactic representations both when parsing student input and when generating teacher output, we can specify the system's educational goal very precisely, as a set of **target syntactic rules**. We assume that the system will deliver a sequence of dialogue-based lessons, beginning with dialogues featuring simple syntactic constructions and progressing in each subsequent dialogue to more complex constructions. The substantive goal of each lesson is for the student to show evi-

dence of understanding the rules 'featured' in the lesson; utterances which involve featured rules (or which are likely to elicit them) can then be scored higher than those which do not.

3.3 The initiative generation algorithm

Our algorithm for generating initiatives has four steps. First, we identify a set of possible topics for the new initiative. During the second step of the algorithm we generate a set of **candidate messages** of each dialogue act type: new assertions, genuine questions and teaching questions. A separate algorithm is used in each case, comprising content selection and sentence planning phases, but stopping short of syntactic realisation. (The algorithms for generating new assertions and genuine questions require the system to have a private knowledge base of facts and question-generation rules; see Section 4.1 for how this is created.) The algorithm for generating teaching questions selects a fact from the common ground and turns it into a yes-no question or a wh-question by manipulating its logical form. The result of these algorithms is a set of candidate messages, each represented as a logical form. We then consult a history of previous system utterances, and discard any initiatives which have previously been generated by the system, whether as initiatives or responses, so that the system never repeats itself when taking an initiative.

The third step of the algorithm consists of scoring the remaining initiatives on a range of different criteria: all initiatives get scores for the suitability of the dialogue act (based on the mix of the previous dialogue acts), the degree of topic maintenance, and finally a dialogue-act-specific score determined in different ways for each different dialogue act. Assertions get a score based on the strategy criterion (e.g. initiatives about topics which the system knows a lot about receive a higher score); teaching questions get a score based on the complexity of the question, with more complex questions being preferred; and genuine questions get a score based on the order in which question-formation rules were entered by the author (see Section 4.1), which reflects the author's view of their importance. The scores are normalised, summed and ranked to create a **shortlist** of initiatives. Finally, each initiative on the shortlist is passed to the sentence generator, and a second evaluation is carried out which assesses to what degree sentences use syntactic rules which have not yet been assimilated by the student. The winning initiative is delivered to the user.

4 Initiative generation in the Kaitito dialogue system

Our dialogue system, called Te Kaitito[1], supports bilingual written human-machine dialogues in English and Māori, the indigenous language of New Zealand. The Te Kaitito CALL system is originally meant to teach Māori, but it has a modular design, and can work to teach any language for which a grammar is specified. In this paper we will use the English grammar, so the system should be viewed as a CALL system for English.

The user and the system alternate in generating contributions to a dialogue. When it is the user's turn to contribute, she enters a sentence in English. The sentence is first parsed, using the LKB system (Copestake *et al.*, 2000) and the ERG grammar (Copestake and Flickinger, 2000). The parser produces a set of syntactic analyses, each of which can have several semantic interpretations after its presuppositions have been resolved against the common ground. One interpretation is then selected, using a combination of disambiguation tech-

[1] Online demos of Te Kaitito can be found at `http://tutoko.otago.ac.nz:8080/teKaitito/`.

niques (see Lurcock *et al*, 2004). The dialogue manager then determines how to create a message in reply—either using a 'response' dialogue act, or by invoking the initiative module. In either case, the response message is passed to a sentence planner for computing referring expressions and discourse signals, and then to a sentence generator. The generator consults the same grammar used by the parser to create the text which is returned to the user.

4.1 Authoring mode dialogues

In order to be able to generate initiatives, a dialogue system needs to be given a knowledge base of private information, on which to draw to create assertions and questions, and a set of substantive goals in relation to which candidate initiatives can be evaluated. In our system, both the knowledge base and the substantive goals are created during a special kind of dialogue with the system called an **authoring dialogue**. In authoring mode, the user is assumed to be a teacher, creating a lesson plan for the system. An example dialogue is given in Figure 1. The system begins with an empty common ground. The teacher authors a character by telling the system facts about itself (e.g. Utterance 1), and by entering **question-generation rules** specifying what kinds of question to ask about different types of objects (e.g. Utterance 2). (The assumption is that the author will enter rules in order of decreasing priority. When ranking alternative candidate initiatives, therefore, the system will prefer a question derived from application of a rule authored earlier during authoring mode.) At the end of an authoring dialogue, the system saves the set of facts in its common ground into a private knowledge base, and saves the set of question-generation rules into a separate private knowledge base. It also automatically creates a set of target syntactic rules for the lesson (see Section 3.2), by traversing the parse trees for every utterance in the authoring dialogue and recording all the rules which are used in this dialogue but not in the authoring dialogues for previous lessons.

4.2 Student mode dialogues

The start of the dialogue

When the system enters **student mode**, its common ground is initialised to empty, and it loads a private knowledge base and agenda of rules created by one of the authoring dialogues. It then enters a conversation with the student.

During the dialogue

Once a dialogue has been initiated, the system and the student alternate in making contributions to the dialogue. The dialogue consists of pairs of forward-looking and backward-looking dialogue acts—for instance, assertions and (possibly implicit) acknowledgements, or questions and answers. At the end of any such pair is a **transition relevance point—**a point where either participant can take an initiative. At such points, the system always passes the initiative to the student. However, the student can concede the initiative, simply by hitting <return>.

An example of a mixed-initiative dialogue based on the facts and goals created in the authoring dialogue in Figure 1 is given in Figure 2. The system begins by generating all possible initiative messages in the null context, and computing their normalised scores. In the initial context, with an empty common ground, genuine questions are preferred over assertions, and teaching questions are impossible. The candidate initiatives whose score exceeds a given threshold are passed to the sentence generation module, and they receive a syntactic score, which is combined with their original scores to generate a total score, as shown in Figure 3. The possible initiatives are

Auth	Your name is John. (1)	Auth	Ask: how is a person? (2)
TK	Okay.	TK	Okay.
Auth	You are from Dunedin.	Auth	Ask: what is a person's name?
TK	Okay.	TK	Okay.
Auth	You are fine.	Auth	Ask: where is a person from?
TK	Okay.	TK	Okay.

Figure 1: A simple authoring dialogue

TK	How are you? (1)	TK	I am John.
S	I am fine.	S	<return>
TK	Okay.	TK	Where are you from? (6)
S	How are you? (2)	S	I am from Dunedin.
TK	I am fine.	S	<return> (7)
S	<return> (3)	TK	I am also from Dunedin. (8)
TK	What is your name? (4)	S	<return>
S	My name is Ann.	TK	Let's see if you remember: Who am I? (9)
TK	Okay.	S	You are John.
S	Who are you? (5)	TK	Correct, well done! (10)

Figure 2: A simple mixed-initiative dialogue

```
DIAL-ACT SCORE TOPIC SCORE QN-ORDER SCORE  SYN. SCORE  TOTAL  SENTENCE
100            100         100              78         378    how are you
100            100          67             100         367    what is your name
100            100         100               3         303    you are how
100            100          67               3         270    your name is what
100            100          33              78         311    where are you from
100            100          33              56         289    from where are you
```

Figure 3: Normalised scores of initiatives in the shortlist for Utterance 1

all genuine questions, so the initiatives receive scores for the following criteria: a dialogue-act score based on the mix of the different dialogue acts, a topic score based on the similarity with the current topics, a question order score based on the order in which the genuine questions were entered by the author, and a syntax score based on the syntactic rules used in the initiative. On this basis, the first initiative generated (Turn 1) is *How are you?*. (Note that an alternative realisation of this sentence, *You are how?* scores badly at a syntactic level, because it involves several rules not used in the authoring dialogue.) If the student does not answer this question as expected the initiative will be repeated. However, in this example the student does answer the question as expected, so the dialogue continues normally. Next, the student is offered the initiative again, and she decides to ask the system a similar question (Turn 2), which the system answers. Then the student concedes the initiative (Turn 3), and the system asks the next-best genuine question (Turn 4). The student answers this, and then asks a similar question in response (Turn 5). The system then asks its last genuine question (Turn 6). When the student responds, and then concedes the initiative again (Turn 7), the system generates a new assertion on the current topic (Turn 8). Finally, when the student again concedes the initiative, the system opts to generate a teaching question (Turn 9), and when the student answers correctly, it provides some positive feedback (Turn 10).

The end of the lesson

The dialogue continues until the system has evidence that each of the target constructions has been assimilated by the user. This evidence comes in a number of forms; for instance, if the user correctly answers a teaching question, the system increments the assimilation score for each rule in both the question and its answer. When all rules have been assimilated, the lesson ends successfully, and the student is allowed to proceed to the next lesson. Sometimes it may happen that a student does not learn a new rule even when (s)he is shown an instance of it being correctly applied. Given that the system never repeats itself when taking an initiative, it might therefore happen that there are no candidate initiatives which will help assimilate any of the remaining unassimilated target rules. In such a situation the lesson ends unsuccessfully, and the student is asked to consult the teacher.

5 Conclusions and future work

Informal evaluations suggest that Te Kaitito's teaching dialogues provide a useful environment in which a student can practice conversational skills in the language being learned. The system's grammar, and repertoire of dialogue moves, are naturally very simple. But in a language-learning environment, particularly for novice language learners, this limited coverage is not as harmful as it normally is. The student's own grammar and vocabulary are similarly limited, and if we know which textbook is being used, and what stage in the book (s)he is at, we have a good chance of being able to build a grammar which can handle all the constructions (s)he is likely to attempt.

We believe that addding initiatives to the language-learning dialogue is very beneficial. If the student is lost, hitting <return> is a simple way of progressing the dialogue (though naturally there are still some places where the student has to respond with something other than <return>). And the initiatives taken by the system create models of well-formed sentences which the student can modify and try out him/herself. In a forthcoming evaluation, we will test more formally whether this is the case.

Naturally there are many aspects of CALL dialogues which we are not yet simulating in the current work. Most obviously, while we are generating teaching *initiatives*, we do not yet generate teaching *responses*—i.e. utterances whose aim is to alert the student to a mistake that (s)he has made, and to provide assistance in correcting the mistake. This is something we are considering in current work.

Acknowledgements

This research was funded by the NZ Foundation for Research, Science and Technology (FRST) grant UOOX0209.

References

Copestake, A. (2000). The (new) LKB system. CSLI, Stanford University.

Copestake, A. and Flickinger, D. (2000). An open-source grammar development environment and broad-coverage English grammar using hpsg. In *Proceedings of LREC 2000*, Athens, Greece.

Desmedt, W. (1995). Herr Kommissar: An ICALL conversation simulator for intermediate German. In M. Holland, J. Kaplan, and M. Sams, editors, *Intelligent language tutors: Theories shaping technology*. Lawrence Erlbaum Associates, Mahwah, NJ.

Haller, S. and McRoy, S. (1997). Papers from the AAAI Spring Symposium on Computational Models for Mixed Initiative Interactions. AAAI Technical Report SS-97-04.

Hovy, E. (1988). *Generating natural language under pragmatic constraints*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Lurcock, P., Vlugter, P., and Knott, A. (2004). A framework for utterance disambiguation in dialogue. In *Proceedings of the 2004 Australasian Language Technology Workshop (ALTW)*, pages 101–108, Macquarie University.

Marcu, D. (1996). Building up rhetorical structure trees. In *Proceedings of the AAAI annual meeting*.

O'Donnell, M., Mellish, C., Oberlander, J., and Knott, A. (2001). ILEX: an architecture for a dynamic hypertext generation system. *Natural Language Engineering*, **7**.

Piwek, P. and van Deemter, K. (2003). Dialogue as discourse: Controlling global properties of scripted dialogue. In *AAAI Spring Symposium on Natural Language Generation in Spoken and Written Dialogue*, Stanford, CA.

Raux, A. and Eskenazi, M. (2004). Using task-oriented spoken dialogue systems for language learning: Potential, practical applications and challenges. In *Proceedings of the InSTIL/ICALL Symposium*, pages 147–150.

Reiter, E. (1994). Has a consensus nl generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the 7th. International Workshop on Natural Language generation (INLGW '94)*.

Seneff, S., Wang, C., and Zhang, J. (2004). Spoken conversational interaction for language learning. In *Proceedings of the InSTIL/ICALL Symposium*, pages 151–154.

Slabbers, N. (2005). A system for generating teaching initiatives in a computer-aided language learning dialogue. Technical Report OUCS-2005-02, Department of Computer Science, University of Otago, Dunedin, New Zealand.

Traum, D., J, B., Cooper, R., Larsson, S., Lewin, I., Matheson, C., and Poesio, M. (1999). A model of dialogue moves and information state revision. TRINDI project deliverable.

What makes Human-Robot Dialogues struggle?

Petra Gieselmann
Interactive Systems Lab
Universität Karlsruhe
Germany
petra@ira.uka.de

Alex Waibel
Interactive Systems Lab
Carnegie Mellon University
Pittsburg, PA 15221
ahw@cs.cmu.edu

1 Introduction

During the last few years, humanoid robots became very popular in the robotic research community and some humanoid robots are already commercially available, such as Asimo from Honda or Qrio from Sony. Comparing the currently possible human-robot communication with the human-human communication we can see that in human-human communication we have efficient strategies to avoid errors and also to recover from them, such as for example *grounding* new information (Traum, 1999; Traum and Dillenbourg, 1998; Poesio and Traum, 1998). This is still one of the biggest challenges for human-robot communication to develop a system which can cope with real world situations and is error tolerant so that it can react in a reasonable way even when something has been misunderstood or not understood at all. Therefore, in this paper we want to evaluate problematic situations in human-robot-communication and how they can be resolved.

Our target scenario is a household situation, in which the user can ask the robot questions related to the kitchen, such as "What's in the fridge ?", "How do I cook Spaghetti Napoli?", ask the robot to set the table, to switch certain lights on or off, to bring some objects, such as cups, dishes, etc. (Gieselmann et al., 2003; Stiefelhagen et al., 2004). In this context which is specifically tailored for unexperienced and older users, it is important that the user can talk to the robot in the same way as to a human servant. This means that the communication should be as natural and as comfortable as possible for the user and therefore, errors should be avoided or at least easy to correct, if they cannot be avoided beforehand.

We can distinguish two kinds of errors: *Non-understanding* vs. *misunderstanding*. Non-understanding means that the dialogue manager cannot find any information in the user utterance. This can be due to the fact that the grammar does not cover the user utterance which cannot be parsed therefore. Also on the pragmatic level, non-understanding is possible, when the user utterance is inconsistent with the current discourse. Misunderstanding means that a user utterance can be parsed and the semantic interpretation is integrated in discourse, but does not correspond to the user's intention. This is above all due to speech recognition errors which means that a word has been misrecognized. But also a semantic misunderstanding might be possible, if some information from the user utterance has been integrated wrongly in the existing discourse.

Therefore, in this paper we want to classify the different kinds of errors which occur in human-robot communication. Section two gives an overview of related work on errors in human-machine dialogues and error classifications. Section three deals with our dialogue

system: The household robot, the dialogue manager, and the web-based interface for user tests of human-robot dialogues are described. Section four gives experimental details and results, and section five gives a conclusion and outlook.

2 Related Work

2.1 Errors in Dialogues

The problems caused by errors in spoken dialogue systems are well known and can result in user frustration and task failure. Most of the research dealing with errors only take speech recognition errors into account until now. For example, Xu et al. and also Gorrell (Xu and Rudnicky, 2000; Gorrell, 2003) use different methods for dialogue state adaptation to the language model to improve speech recognition. Also different stages and language models are used to reduce word error rates and perplexity in error dialogues: A general n-gram language model is used at the beginning and in underspecified situations and a specialized language model which can be an n-gram language model or a grammar-based one is used in specific situations based on the preceding system prompt (Fosler-Lusier and Kuo, 2001). In (Solsona et al., 2002), the state-independent n-gram language model is also combined with a state-dependent finite state grammar by comparing the acoustic confidence scores. Furthermore, work on hyperarticulation concludes that speakers change the way they are speaking when facing errors in principle so that the language model has to be adapted therefore (Stifelman, 1993; Hirschberg et al., 2004).

Choularton (Choularton and Dale, 2004) examines different repair strategies of the users and how these strategies can be generalized to be domain-independent. Also Stifelman explains the user reactions to errors and how repair utterances can be automatically detected on the acoustic side (Stifelman, 1993). Both of them are looking for general strategies on error recognition and repair to prepare the speech recognizer better to the special needs of error communication.

Our concern, however, is with slightly different analyses in order cope with errors more efficiently: We want to concentrate on semantic errors and how they can be classified. We avoided speech recognition errors by using an interface with keyboard input to our robot, as explained in section 3.3. We want to find out the reasons for errors in order to avoid them as far as possible. Furthermore, we want to have a look at repair dialogues in order to be able to perform efficient error handling strategies in the future so that it is easier for the user to correct errors which could not be avoided.

2.2 User Tests and Error Classification

At the moment, there exist only very few error classifications based on the semantics of user utterances. Most of the researchers use the Levenstein distance (Levenstein, 1996) which gives the cheapest way to transform one string into another one by combining the following steps:

- **Substitution** of one symbol by another one
- **Deletion** of one symbol by another one
- **Insertion** of a new symbol

But since this is not useful in our case to find out, why the dialogue failed, we made a new error classification which is based on the semantics of the user utterance and possible reasons why it cannot be understood by the system.

3 The Dialogue System

3.1 Our Household Robot

We developed a rapid prototype system with approximately 33 dialogue goals, 190 dialogue moves and more than 140 ontology concepts. Furthermore, we developed more than

650 grammar rules and the lexicon has now more than 250 entries. By means of this prototype we started user tests and interactively develop now new versions of the robot grammar and domain model.

The robot can accomplish different tasks in the household environment. The user can for example ask it to get something from somewhere, put something somewhere else, set the table, switch on or off different lamps, to give him information about some recipes, make a cup of coffee or tea, etc.

3.2 Dialogue Management

We use the TAPAS dialogue tools collection based on the approaches of the language and domain independent dialogue manager ARIADNE (Denecke, 2002) which is specifically tailored for rapid prototyping, so that can interactively develop new versions relying on the same base technology. We developed the domain and language dependent components, such as an ontology, a specification of the dialogue goals, a data base, a context-free grammar and generation templates.

The dialogue manager uses typed feature structures (Carpenter, 1992), to represent semantic input and discourse information. A context-free grammar enhanced by information from the ontology defining all the objects, tasks and properties about which the user can talk parses the user utterance. The parse tree is converted into a semantic representation and added to the current discourse. If all the information necessary to accomplish a goal is available in discourse, the dialogue system calls the corresponding service, such as "getting the cup from the table to the user". Otherwise, the dialogue manager generates clarification questions to the user by means of generation templates.

3.3 Web-based User Interface

An internet user test has the advantage that lots of users all over the world can participate whenever they like to so that the costs in time and money are lower than in other user studies (Schmidt, 1997). Also Reips explains these advantages of web-based experiments, such as "speed, low cost, experimenting around the clock, and a high degree of automation" (Reips, 2002). Therefore, we made the rapid prototype accessible via the internet, as you can see in figure 1 and posted the link to different news groups and added it to some experimental portals in the web to get as much user data as possible.

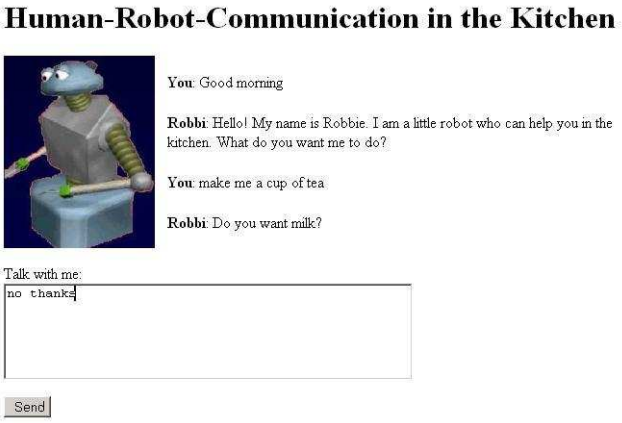


Figure 1: The web-based Internet Interface for our Humanoid Robot

One drawback of web-based experiments is that users might dropout quite easily because there is no experimenter available who forces them to stay (Reips, 2002). But at the same time especially in our case this resembles much more the real world situation where the user has the robot in his own home and can decide whether he wants to use it or not. Therefore, we carefully evaluate all the situations when the users dropped out to avoid them in the future for a more comfortable use of the robot.

4 Experimental Details & Results

4.1 Details

The data are collected with about 70 test persons. All together, we have about 1000 turns;

```
1. Ask Robbi to make you a cup of tea with milk and sugar.
2. Ask Robbi to get you some water.
3. Ask Robbi to get you the blue cup.
4. You would like to cook Spaghetti Napoli.
Ask Robbi, how to do this.
5. You invited some friends for diner.
Ask Robbi to set the table for all of you.
6. Ask Robbi to make you a cup of coffee without milk, but with sugar.
7. Ask Robbi to get you some coke.
8. Ask Robbi to switch on the small lamp.
9. Imagine that you come home after work and are very hungry.
Now you want Robbi to cook something for you.
10. Imagine that you are sitting on your sofa thinking what you might cook
this evening. Since you are too lazy to go to the kitchen, you ask Robbi
to have a look at the fridge, what is still there.
```

Figure 2: Tasks for the User Test

on average, there are 15 turns per user. All the users talked to our robot via the webinterface and got the instruction to make the robot do five of the predefined tasks you can see in figure 2.

	on Average
Accomplished Tasks	2.65
New Tasks	0.81
New Objects	0.53
New Words	3.34
Overall Turns	14.48
	Rates (in %)
Parsability	74.62%
Turn Error Rate	56.2%
Finalized Goal Rate	25.3%
Dropout Rate	1.22%

Table 1: Detailed Results

4.2 Results

As you can see in table 1, the users managed to let the robot do more than half of the predefined tasks. The turn error rate was quite high because the system was only a prototype which did not cover all the utterances the users invented. Furthermore, some users did not read the instructions carefully and entered punctuation marks and digits which could not be parsed by the current dialogue manager because it expects input similar to the one from a speech recognizer. Therefore, we want to integrate a small component which can delete all the punctuation marks in the future.

Since the grammar was only a prototype, it did not cover all the user utterances, but some new concepts were used. In addition, we also found some new goals which were not covered by our dialogue manager. These new goals concern above all meta-communication, such as ”what can you cook?” or ”do you know the word coffee?”. Since the users got predefined tasks to accomplish, most of the other goals are already covered by the grammar. For the same reason, the users refer only to very few new objects, such as new recipes for example. They used some new words for known objects, such as ”cream” instead of ”milk”.

Since a conversation which consists of less than five turns means that the user talked to the robot less than a minute, we determined five turns as a limit for a conversation. Only very few users dropped out given this limit of five user turns, but most of the users seemed to have acquired a taste for the robot communication and went on talking with it for quite some time. All the users who dropped out did not manage to make the robot understand them at all during these first few turns which was most of the time due to the problem with punctuation marks mentioned above.

About three fourth of the user utterances can be parsed, but some of them cannot be transformed to the complete, correct semantic representation which explains the slightly higher turn error rate. We now want to have a closer look at all the utterances which cannot be understood correctly and results in errors. Therefore, we manually tagged all the utterances by means of the reasons why they failed, as you can see in table 2.

4.3 Error Analysis

We noticed that the main reason for errors were new ontological or grammatical concepts (cf. Table 2). Lots of new syntactical constructions were used, such as ”prepare a salad” instead of ”make a salad”, ” i want you to cook spaghetti for me” instead of ”please make spaghetti napoli”. Sometimes the participants used also new words for known objects, such as ”icebox” instead of ”fridge” or ”soda” instead of ”water”. This might be due to the fact that we only had a small prototype grammar. It is possible that a more complete grammar would result in lesser errors in this area. This could be explored in future studies.

Also some new goals were used by the participants, such as ”switch yourself off”, ”can you wash the dishes”. But above all most of the new goals can be defined as meta-communication and clarification questions from the user as already described in the previous section. When the robot did not understand the user, he tried to detect what went wrong by asking questions such as ”are you making the coffee?” or ”can you understand me?”. Therefore, we want to integrate a component in the future which can deal with all this kind of meta-communication and has access to the context model and the discourse to include the previous user utterances.

Very few new objects were used such as ”cupboard”, ”dustbin”. The small grammar seems to already cover most of them because we have such a fixed set of tasks the user should accomplish. It would be an interesting topic for future studies to see whether more complex task sets also require a bigger variability within the vocabulary.

Sometimes, the context to resolve an utterance is missing and also elliptical utterances and anaphora can be found quite often. As you can see in Figure 3 in the first example, where the users refers to the ”lamp” by saying ”the small one”, we need to include context management issues in future versions to resolve elliptical and anaphoric utterances.

On the other hand, we also have some utterances which are too complex and contain concatenated sentences which cannot be resolved at the moment, such as for example ”I need a cup of coffee that has about a quarter cup of milk in it”, ”I want you to cook spaghetti for me and a coke”, etc. In theses cases, we want to make sure that at least one part of the utterance can be understood so that the user can repeat the other one later.

In addition, lots of users entered punctuation marks and digits which could not be parsed by the current dialogue manager, such as ”set the table for 4 people”, ”How do I cook Spaghetti Napoli?”, ”Yes.”. All the digits and words with punctuation marks at the end are simply ignored by the parser and result in errors therefore.

And we have some grammatically wrong utterances, such as ”can you put the light on”, ”you bring coffee”, ”cook a salad for me”, ”i want drink”, etc. Since these seem to be due to the fact that some of the user were nonnatives, we want to integrate therefore possibilities to parse at least parts of ungrammatical sentences so that the robot can ask clarifications questions to the user to understand what he wants to say.

4.4 User Behaviour with Respect to Problems and Errors

The behaviour of the users when facing problems is very interesting: Lots of users tend to

	Rate of All Utterances	Rate of not Understood Utterances
New Concepts		
New Formulation	3.34%	39.82%
Meta Communication	0.76%	9.12%
New Goal	0.41%	5.65%
New Object	0.53%	6.32%
Ellipsis		
Elliptical Utterances / Anaphora	0.63%	7.54%
Missing Context	0.22%	2.63%
Complexity		
Concatenated Utterances	0.48%	5.79%
Input Problems		
Punctuation	0.97%	11.58%
Digits	0.22%	2.63%
Grammatically Wrong Utterances	0.37%	4.39%

Table 2: Error Analysis

use shorter utterances, sometimes even only a single word, when they notice that the robot did not understand them. For example, you can see in Figure 3 in the second example that a user who has not been understood by the robot, just repeated the object "blue cup".

In the third example (cf. Figure 3), you can even see a more condensed user utterance, after an error occurred: After the robot did not understand "I want to drink coffee." because of the punctuation mark at the end, the user just said "kitchen-coffee". Of course, the system would never understand this utterance.

```
User: switch on the lamp
Robbi: Which lamp do you want me
to switch on?
User: the small one

User: get me the blue cup
Robbi: Sorry. I didn't understand you,
please try again!
User: blue cup

User: I want to drink coffee.
Robbi: I am sorry.
I can't understand you.
User: kitchen-coffee
```

Figure 3: Example Dialogues from the User Test

In the event of an error, the users tried different words and just waited whether the robot understood them. The problem is that most of the words were correctly understood by the system, but since they could not be interpreted without context, there was no answer from the dialogue system. In the future, we will therefore implement methods to cope with single words and give the user better feedback so that he knows what had been understood.

In addition, it seems to be a general problem that the dialogue system understands some parts of the user utterance, but does not indicate this to the user so that he thinks that it did not understand. In the example below (cf. Figure 4), the user told the robot to get him the blue cup. But since this verbal construction has not been covered by the prototype grammar, the system only understood the object, namely "the blue cup" which the user did not know but thought that everything was misunderstood and mentioned again "blue cup". Better grounding mechanisms have to be implemented in the future for all these cases.

```
User: can you get me the blue cup please
Semantic Representation: [generic:object,NP,_] { [robbi:obj_cup,N,_]
{ THE BLUE CUP } [robbi:please_NT] { PLEASE } }
Robbi: Sorry. I didn't understand you, please try again!
User: blue cup please
Semantic Representation: [generic:object,NP,_] { [generic:object,NP,Sg]
{ [robbi:obj_cup,N,Sg] { CUP } [robbi:please_NT] { PLEASE } } }
```

Figure 4: Example Dialogues with semantic Representations from the User Test

5 Conclusion & Outlook

In this paper, we presented the results of an internet user test of the dialogue management component of our household. The results showed that most of the errors in human-robot communication are due to new formulations and missing mechanisms to deal with meta-communication and elliptical utterances.

Furthermore, the user test showed that lots of users tried to get the communication back on track by using shorter and shorter utterances. Unfortunately, even if these utterances had been understood correctly, the dialogue manager did not give any feedback to the user, but waited for more input. Therefore, we want to integrate a component which can handle these short utterances and adds them to the common ground. In this way, a clarification dialogue can be initiated to find out what the user wants to do. In addition, this component can also help avoiding errors resulting form elliptical utterances.

Acknowledgments

This work was supported in part by the German Research Foundation (DFG) as part of the SFB 588 and by the European Commission under project CHIL (contract #506909).

References

B. Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press.

S. Choularton and R. Dale. 2004. User responses to speech recognition errors: Consistency of behaviour across domains. *Proceedings of the 10th Australian International Conference on Speech Science & Technology*.

M. Denecke. 2002. Rapid prototyping for spoken dialogue systems. *Proceedings of the 19th International Conference on Computational Linguistics*.

E. Fosler-Lusier and H.K. J. Kuo. 2001. Using semantic information for rapid development of language models within asr dialogue systems. *Proceedings of ICASSP*.

P. Gieselmann, C. Fuegen, H. Holzapfel, T. Schaaf, and A. Waibel. 2003. Towards multimodal communication with a household robot. *Proceedings of the Third IEEE International Conference on Humanoid Robots (Humanoids)*.

G. Gorrell. 2003. Recognition error handling in spoken dialogue systems. *Proceedings of the 2nd International Conference on Mobile and Ubiquitous Multimedia*.

J. Hirschberg, D. Litman, and M. Swerts. 2004. Prosodic and other cues to speech recognition failures. *Speech Communication*, 43.

V. I. Levenstein. 1996. Binary codes capable of correcting insertion and reversals. *Cybernetics and Control Theory 10*.

M. Poesio and D. Traum. 1998. Towards an axiomatization of dialogue acts. *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology)*.

U.-D. Reips. 2002. Standards for internet-based experimenting. *Experimental Psychology*, 49(4).

W. C. Schmidt. 1997. World-wide web survey research: Benefits, potential problems, and solutions. *Behavior Research Methods, Instruments & Computers*, 29(2).

R. A. Solsona, E. Fosler-Lussier, H.-K. J. Kuo, A. Potamianos, and I. Zitouni. 2002. Adaptive language models for spoken dialogue systems. *Proceedings of the ICASSP*.

R. Stiefelhagen, C. Fuegen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. 2004. Natural human-robot interaction using speech, gaze and gestures. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

L. J. Stifelman. 1993. User repairs of speech recognition errors: An intonational analysis. *Technical Report, Speech Research Group, MIT Media Lab*.

D. R. Traum and P. Dillenbourg. 1998. Towards a normative model of grounding in collaboration. *Working notes of the ESSLLI-98 workshop on Mutual Knowledge, Common Ground and Public Information*.

D. R. Traum. 1999. Computational models of grounding in collaborative systems. *Psychological Models of Communication in Collaborative Systems - Papers from the AAAI Fall Symposium*.

W. Xu and A. Rudnicky. 2000. Language modeling for dialog system. *Proceedings of ICSLP*.

Integrating a Discourse Model with a Learning Case-Based Reasoning System

Karolina Eliasson
Department of Computer and Information Science
Linköping University
Linköping Sweden
karel@ida.liu.se

Abstract

We present a discourse model integrated with a case-based reasoning dialogue system which learns from experience. The discourse model is capable of solving references, manage sub dialogues and respect the current topic in a dialogue in natural language. The framework is flexible enough not to disturb the learning functions, but allows dynamic changes to a large extent. The system is tested in a traffic surveillance domain together with a simulated UAV and is found to be robust and reliable.

1 Introduction

For a dialogue in natural language to run smoothly, the participants have to know the history of it. If a computer dialogue system will be able to work properly in such a natural dialogue with a human user, it has to maintain a discourse model of the dialogue so far to be able to interpret the utterances of the user in the right context. The discourse model helps the system to interpret references to utterances earlier in the dialogue. The system also need to know if an utterance shall be interpreted in the earlier discourse or if it is a start of a new dialogue with a new discourse.

In this paper, we will describe a discourse model which is integrated in a case-based reasoning (CBR) system used for dialogue with a robot. Case-based reasoning is a form of machine learning where the system stores problems and their corresponding solutions in a case base. When a new target case enters the system, it searches the case base for similar cases. When the most similar case is found, its corresponding solution is adapted to the new target case and the new solution is returned. The new target case and its solution are then stored in the case base for future use. See for example (Aamodt, 1994) for an overview.

CBR provides our dialogue system with a simple and modular design. New functionality is directly added by writing new cases and storing them in the case base. New domain knowledge similar to existing knowledge can be added to the system in a simple manner. It can directly be used by the system without any additional changes to the case base, due to the flexible and adaptable nature of the CBR design. This provides us with the facility of letting the system incorporate new information, such as new words or knowledge about the physical world, into the system. This knowledge can then directly be used by the cases in the case base, hence giving the system mechanisms for updating its own knowledge and increasing its performance. The new information can be obtained from dialogue with an

operator. Because phrase matching is necessary both in CBR and in discourse modeling, in the latter to allocate incoming new phrases to the correct dialogue thread, it makes CBR and discourse modeling a suitable combination without producing any additional overhead.

We have chosen to work on the discourse model presented in (Pfleger et al., 2003) for the SmartKom project. Our structure of the discourse model as described in section 3 is highly inspired by their model. Our contribution to their work is mainly the integration of the model with CBR which is described in section 4 and 5.

2 Dialogue System

CEDERIC, Case-base Enabled Dialogue Extension for Robotic Interaction Control, is a dialogue system designed for dialogue with a physical robot, in particular the WITAS autonomous unmanned aerial vehicle (UAV). The WITAS project focuses on the development of an airborne computer system that is able to make rational decisions about the continued operation of the aircraft, based on various sources of knowledge including prestored geographical knowledge, knowledge obtained from vision sensors, and knowledge communicated to it by data link (Doherty et al., 2000). The UAV used in the project is a Yamaha RMAX helicopter which an operator can control by high level voice commands or by written commands. The operator can ask the UAV to perform different tasks and answer questions.

CEDERIC consists of a *case base*, *domain knowledge*, a *discourse module* and a *case-base manager* as shown in Figure 1. The domain knowledge contains an ontology of the world as the robot knows it, a categorization of the world items, and a grammar. The purpose is twofold. It serves as a world representation which gives CEDERIC knowledge about which buildings there are in the known

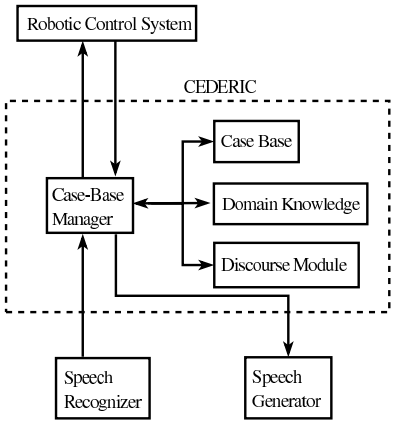


Figure 1: Architecture of CEDERIC.

world, what kind of buildings they are, where they are placed, and their attributes such as color and material. It also gives CEDERIC fundamental knowledge about which items that can be called buildings in the dialogue and which can not and provides CEDERIC with a grammar so that the system can interpret natural language. The ontological information is then used to measure the similarity of two different knowledge items. Items belonging to the same ontological class is considered similar.

The operator can choose to use either speech or text for the input to the dialogue system. The speech recognizer used is the off-the-shelf product Nuance and the speech generator used is one of the off-the-shelf products Festival or Brightspeech. When learning a new word using speech recognition, one can choose between having a considerably bigger grammar for the speech recognizer than the dialogue manager and only consider learning in the dialogue manager, or provide the new word in text form in the learning phase and then compile it into the speech recognition grammar at runtime. We have chosen the second approach where the unknown words are provided in text and the learning phase extends the grammar.

When a new sentence arrives from the op-

erator CEDERIC looks for cases similar to the new target case. The solution to it is either an utterance in return to the user or a request to the robotic control system. The robot acts upon the request and produces a response that is catched by CEDERIC, who searches its case base and returns a message to the user. The system can manage simple cases of dialogue such as a command from the user that directly produces an answer even without a discourse model, but to be able to handle a more natural and sophisticated dialogue such as references to earlier objects and clarifying questions (where?, what?, which?, why?), a discourse model is necessary. This paper is particulary focused on the discourse model implemented in CEDERIC and how it can be used in a case-based system. For a description of the total system, see (Eliasson, 2005).

The following dialogue problems are addressed in the paper:

Anaphora references. The discourse model should be able to solve references to objects which have occurred in an earlier stage of the dialogue.

Sub dialogues. It should be able to recognize if an utterance is a sub dialogue to the present dialogue and hence should be interpreted within the limits of the current discourse or if it is the start of a new dialogue. It should also recognize a dialogue as completed which makes the old discourse no longer applicable. It should be possible to return to older non-completed dialogues which is not presently in focus.

Topic management. The discourse model should be able to figure out if it is a good moment to mention e.g. an observed event or if that utterance should wait for a better occasion when it does not disturb the present dialogue.

3 Discourse Model Design

The discourse model we have chosen to implement in CEDERIC is very similar to the one presented in (Pfleger et al., 2003). It is built up of four different objects, which is linked to one another in a hierarchical manner which constitutes the meaning of the dialogue.

The linguistic objects. These objects are furthest down in the chain of objects and thus most specific on the word level. They contain information of how the nouns in the dialogue where uttered. They could for example have been references by the word `it` or by a noun and a determinant.

The discourse objects. These objects contain the different nouns together with their attributes mentioned in the dialogue. A discourse object can also be composite. An enumeration of several objects can be seen as a discourse object representing the enumeration as such and this object contains the enumerated objects as its children. This gives CEDERIC the opportunity to understand references referring to the order of the enumerations, e.g. `the first one`. The discourse objects have a link to the corresponding linguistic object.

The dialogue objects. These objects groups the sentences and their information together which have the same direct goal. The sentence `fly to the hospital` gives for example, when it is executed, a dialogue object which groups the sentences `fly to the hospital`, `ok` and `I am at the hospital now` together. If any sub dialogues come up, they will be saved in a new dialogue object with their direct goal to clarify some matter in the

dialogue. Dialogue objects contain information about the topic of the dialogue, which discourse objects that were created due to the utterances, and which future utterances this dialogue object expects to consider the dialogue or the sub dialogue completed. These expectations on future dialogue are saved in a modified *initiative-response (IR) unit* (Ahrenberg et al., 1991). IR-units in our context can, unlike the original IR-units described by Ahrenberg, contain more than two sub elements. That is because they shall also be able to represent the response from the robot when the system sends a request. The `fly to the hospital` example above shows such an example.

The global focus space. The different objects in the dialogue layer which belongs to the same dialogue, including sub dialogues, are grouped together in a top object called the global focus space. It contains information about the main topic of the dialogue, if it is ok to interrupt the dialogue and which dialogue objects that belongs to it. Each global focus space also keeps track of the discourse object last mentioned, to be able to resolve references such as `it`. This is known as the *local focus stack*. The last mentioned discourse object is said to be in focus.

To keep track of the current dialogue in focus, CEDERIC saves the different global focus spaces in a stack called the *global focus stack*. The global focus space on top of the stack is said to be the one in focus. If every IR-unit belonging to a global focus space is closed, that is, has received all its subelements, the global focus space is marked as closed and removed from the stack. Several dialogues can be open and ongoing at the same time and are thus members of the stack but only one dialogue can be in focus at the same time.

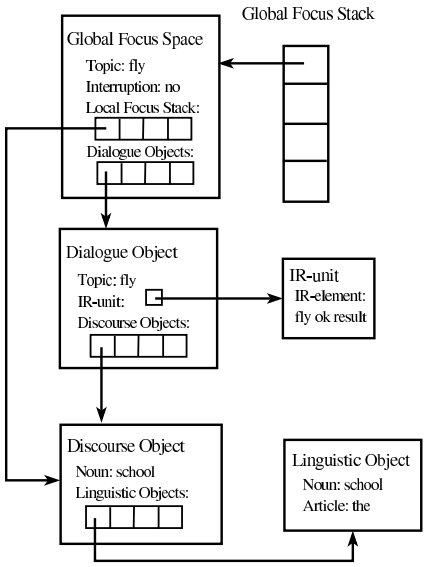


Figure 2: An example of a discourse model.

Figure 2 shows an example of how the discourse model looks like when the utterance `Fly to the school` has been executed.

4 Discourse Information in the Cases

When a new utterance enters the system, it is not only the utterance itself, but also the dialogue discourse, that tells the system how it should be interpreted. The simple answer `yes` to a question is an illustrative example of this. Without knowing the question, the answer carries no information at all. Therefore, to match a case in the case base, not only the utterance by itself but also the discourse needs to match. When a matching case is found, the system knows which information the new utterance carried and the discourse has to be updated accordingly to reflect this new information.

A case in our approach is divided into five different parts:

The problem. The problem is a description of the utterance. It contains the words and their classification according to the grammar in the domain knowledge.

The discourse information. This part describes how the global focus space in focus and its discourse object in focus should look like. It makes sure that utterances such as answers to questions are executed with the correct case.

The update according to problem. Depending on the problem, the discourse model has to be updated with the new information. This information is stored in this part.

The solution. This part contains the reaction to the problem. It can be a request to the robot to perform an action or an answer in natural language to the operator.

The update according to solution. When the solution has been executed, the discourse model has to be updated to reflect it.

If a new dialogue is started, a new global focus space with one or more dialogue objects with corresponding IR-units, one or more discourse objects, and one or more linguistic objects are created. This newly created global focus space is put on top of the global focus stack and the local focus stack of the new global focus space is populated with the new discourse objects. Possible old open global focus spaces on the global focus stack are left in the stack as they are and are still reachable although not in focus.

If the new problem is an expected continuation of an ongoing dialogue, the case returns the newly satisfied elements of the IR-unit and CEDERIC updates the above IR-units accordingly. In case all elements in the IR-unit have been satisfied, the IR-unit is closed and CEDERIC checks if the global focus space of that IR-unit only consists of closed IR-units. In that case the whole global focus space is marked as closed.

In case CEDERIC needs to ask a clarifying question to a given problem to be able to unambiguously interpret the meaning of the operator's utterance, a new dialogue is created. The new dialogue object is created in the same global focus space that matched the case, because the new dialogue is only a sub dialogue to the main one. A new IR-unit is created and possible discourse and linguistic objects are created as well. If a new discourse object is created, it is put on top of the local focus stack.

If the solution to the case is a request to the robot, the discourse model notices it and starts to expect a response from the robot.

5 Case Matching

When a new utterance from the operator or a message from the robot enters the system, it starts by classifying the included words according to the grammar. Then the case base is searched for cases with similar utterances. The current discourse in focus is matched with the discourse information saved in the case, hence a match implies that the utterance can be evaluated in the current discourse in focus.

If no case matches the new problem and the discourse currently in focus, one of the following scenarios has happened:

- The operator or the robot returns to an older open discourse.
- The operator or the robot changed topic and started a new dialogue.
- CEDERIC did not understand the new utterance either because the utterance as such is not represented in the case base or it is totally out of context and no suitable open discourse is found.

The operator is free to change subject of the dialogue at any time by starting a new dialogue or return to an old open one. If no matching case is found using the present discourse in focus and the utterance origins from

O: Fly to the school.
C: I have two schools to choose between. Which one do you mean?
O: Take off.
C: Ok.
O: Which can I choose between.

CEDERIC gets a message from the robot saying that the action take off has been successfully completed

C: You can choose between the one on Harborroad and the one on Mainstreet.
O: Fly to the hospital.
C: Ok.
C: I have taken off now.
O: What is your altitude?
C: It is 20 meters.
C: I am at the hospital now.

Figure 3: An example of dialogue topic changes and topic management between the operator and CEDERIC.

the operator, CEDERIC investigates if it is possible to match the utterance with a discourse from an earlier open dialogue. These dialogues are stored in the global focus stack. If a match is found using an old global focus space the solution to the case will be evaluated in the discourse represented by that old global focus space and it will be put in focus. It will also be updated so it correlates with the result of the evaluation of the solution. If no matching case is found this way either, the operator did not return to an earlier dialogue and the utterance is again matched against the cases in the case base. This time with no discourse in focus at all. If matching, this indicates that the operator has started a new dialogue. If a match is found, the adapted solution is evaluated with an empty discourse and a new global focus space is put on the global focus stack. If there is no match with an empty discourse either, CEDERIC matches with a default error case.

If it, on the other hand, is a message from the robot that does not match with the present discourse in focus, CEDERIC has to take the topic management into consideration. A report of a result of a performed command shall for example not be mentioned right away if the operator waits for an answer to a question. CEDERIC decides what to do by investigating the global focus space currently in focus and checks if it is ok to interrupt in the present discourse or not. If it is ok to interrupt, the same algorithm as the one for an utterance from the operator is performed, but if it is not, the message is put in a queue and is evaluated as soon as it is ok to interrupt or the present dialogue is closed.

6 Result

CEDERIC has been tested connected to a simulated UAV situated in a simulated environment. The simulated UAV can perform high level actions such as flying to a building with a certain unique identity, flying in a certain direction, take off, land, ascend and descend. It reports the result of an action and also report if it observes any buildings as it flies. It is able to answer status questions such as the current altitude, velocity and heading etc. A number of dialogues have been implemented where both references, dialogue topic changes, and topic management during the shifts have been carefully tested. Figure 3 shows an example where the operator switches dialogue topic back and forth. CEDERIC is able to keep the different discourses in mind and use the correct discourse model for every new utterance. It is also an example of a case where CEDERIC gets a message from the robot, but due to the current dialogue, the message is restrained until a suiting moment in the dialogue appears. The operator's utterances are never restrained and he or she is free to lead the dialogue as he or she wishes.

O: Fly to the church.
C: I do not know what the noun church means. Is it a building?
O: Yes.
C: Ok, where is the church you want me to fly to?
O: East of the hospital.
C: Ok, I will start by flying to the hospital.
C: I am at the hospital now.
O: Fly east and look for a white building.
C: Ok, I'll stop when I see such a building.
C: I see the church now and hover over it.
O: Fly to the school on Mainstreet.
C: Ok.
C: I am at the school now.
O: Fly to the church.
C: Ok.
C: I am at the church now.

Figure 4: An example of a teaching situation between the operator and CEDERIC.

More complex examples where the operator teaches CEDERIC new information have also been tested. An example of such a dialogue is provided in Figure 4, where the operator teaches CEDERIC a new word and a domain item, in this case a new building. The result of this dialogue is, besides the movement of the robot, an update of the grammar and the domain knowledge. As seen in the end of the dialogue, CEDERIC can successfully fly to the church after the explaining dialogue.

The tests have proven the discourse model to be robust with ability to cope with various types of operator input as partly shown above. Due to the flexibility of the information stored in the discourse model it is well suited for CBR systems and various learning strategies.

7 Related Work

Dialogue in CBR is mainly used in conversational CBR (CCBR) where the system asks the user questions which guides the search for the most similar case in the case base. Unlike in CEDERIC, the dialogue acts are not by themselves items in the case base, but rather a help to discriminate the cases. For an overview, see (Aha et al., 2001). Some work has been done in integrating a discourse model with CCBR, e.g. Branting's discourse model for conversational CBR (Branting et al., 2004). Brating's discourse model is however not integrated with the cases in the case base.

Because our CBR-system for dialogue with a robot is not a pure conversational CBR system, but has with respect to its use of dialogue more in common with non-learning dialogue systems such as (Allen et al., 2001; Rosset and Lamel, 1999), we have integrated a discourse model built on the traditional principles with CBR.

Within the WITAS project, several dialogue systems with various capabilities have been developed. The first WITAS Dialogue System (Lemon et al., 2001) was a system for multi-threaded robot dialogue using spoken I/O. The DOSAR-1 system (Sandewall et al., 2003) was a new implementation using another architecture and a logical base. This system has been extended into the current OPAS system (Sandewall et al., 2005). Our work takes a rather different approach to discourse modeling, compared to these predecessors, as we are integrating CBR techniques, but it reuses major parts or the OPAS implementation for other aspects of the system. For additional information, please refer to the WITAS web site at `http://www.ida.liu.se/ext/witas/`.

8 Conclusion and Future Work

We present a discourse model called CEDERIC which is integrated with a CBR-system for communication with a robot. We have shown how the cases updates the discourse model which gives scope for learning of new dialogues and dialogue structures within the

loose framework the discourse model defines. This way, we can control which cases matches the new problem not just by comparing the problem statements but also by comparing the discourse, which gives us the opportunity to solve problems such as references, sub dialogues and topic management in a learning system.

Our implementation has been tested connected to a simulated UAV operating in a simulated environment. The resulting system is robust and allows the operator to take the initiative in the dialogue at any time without loosing track of the discourse. It has also proven easy to work with and new cases can easily be automatically generated from new target case problem, the adapted discourse description, the adapted solution and the adapted discourse update. In fact, the adapted discourse description is generated per se because it is the same discourse as the one currently in focus.

The integrated discourse model is an aid for our primary goal to design a dialogue system not only capable of learning in a restricted area but to be able to handle a large amount of utterances and advanced dialogue both from the operator and from the robot. The advanced dialogue features provides a platform for further research regarding giving the operator the opportunity to explain new domain and dialogue knowledge to the system and the ability for the system to ask for confirmation to a solution.

Acknowledgement

This research work was funded by the Knut and Alice Wallenberg Foundation, Sweden.

References

Agnar Aamodt. 1994. Case-based reasoning; foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59.

David W. Aha, Leonard A. Breslow, and Hector Munoz-Avila. 2001. Conversational case-based reasoning. *Applied Intelligence*, 14(1):9–32.

Lars Ahrenberg, Arne Jönsson, and Nils Dahlbäck. 1991. Discourse representation and discourse management for a natural language dialogue system. Technical report, Institutionen för Datavetenskap, Universitetet och Tekniska Högskolan Linköping.

James Allen, George Ferguson, and Amanda Stent. 2001. An architecture for more realistic conversational systems. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 1–8. ACM Press.

Karl Branting, James Lester, and Bradford Mott. 2004. Dialogue management for conversational case-based reasoning. In *Proceedings of the Seventh European Conference on Case-Based Reasoning*.

Patrick Doherty, Gösta Granlund, Krzysztof Kuchinski, Erik Sandewall, Klas Nordberg, Erik Skarman, and Johan Wiklund. 2000. The witas unmanned aerial vehicle project. In *Proceedings of the 12th European Conference on Artificial Intelligence*.

Karolina Eliasson. 2005. Towards a robotic dialogue system with learning and planning capabilities. In *Proceedings of the 4th Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Oliver Lemon, Anne Bracy, Alexander Gruenstein, and Stanley Peters. 2001. The WITAS multi-modal dialogue system. In *Proceedings of EuroSpeech*.

Norbert Pfleger, Jan Alexandersson, and Tilman Becker. 2003. A robust and generic discourse model for multimodal dialogue. In *Workshop Notes of the IJCAI-03 Workshop on Knowledge and Reasoning in Practical Dialogue Systems*.

Sophie Rosset and Samir Bennacef Lori Lamel. 1999. Design strategies for spoken language dialog systems. In *Proceedings of EuroSpeech*.

Erik Sandewall, Patrick Doherty, Oliver Lemon, and Stanley Peters. 2003. Words at the right time: Real-time dialogues with the witas unmanned aerial vehicle. In *Proceedings of the 26th Annual German Conference in AI*.

Erik Sandewall, Hannes Lindblom, and Björn Husberg. 2005. Integration of live video in a system for natural language dialog with a robot. In *Proceedings of DIALOR-05*.

Designing an Open, Multidimensional Dialogue Act Taxonomy

Harry Bunt and Yann Girard
Tilburg University
{bunt|y.girard}@uvt.nl

Abstract

This paper discusses issues in the design of a rich taxonomy of dialogue acts that is hierarchically structured in such a way that a notion of 'dimension' is reflected, corresponding to the intuition that communication is a complex activity, with multiple aspects that can be addressed simultaneously. The taxonomy is also 'open' in the sense that it is based on clear criteria for including dialogue act types and for how they fit into the taxonomy, allowing easy addition of other act types.

1 Introduction

To describe what is happening in a dialogue from semantic and pragmatic points of view, it has become common to consider dialogues in terms of communicative actions, called 'communicative acts' or 'speech acts' or 'dialogue acts'. In recent years the term 'dialogue act'' has become particularly popular among researchers involved in the design of computer dialogue systems or in dialogue annotation, see e.g. Jurafsky & Martin (2000).

This paper is concerned with the definition of dialogue acts and especially with the definition of taxonomies of dialogue acts, inspired by the goal to build a public registry of dialogue act specifications, as undertaken by the Task Domain Group on Semantic Content Representation within the International Standards Organisation ISO (ISO/TC 37/SC 4/TDG 3).[1] We outline a number of fundamental and practical issues that have to be addressed in developing a repository of dialogue acts, focusing on the following issues:

- How are dialogue acts defined? How do dialogue acts relate to speech acts, communicative acts, utterances, turns, etc.?
- What uses of dialogue acts do we envisage, that should be supported by a repository of dialogue acts? What requirements on dialogue act specification follow from potential uses of dialogue acts, such as manual or automated annotation?
- What exactly does it mean for a dialogue act annotation system to be 'multidimensional' and/or 'layered'? How are 'dimensions' (and 'layers') defined, and why?

[1] Some of the ideas presented in this paper have been introduced in a presentation at the 4th Joint ISO-SIGSEM Workshop on the Representation of Multimodal Semantic Information, Tilburg, January 10-11, 2005, and appear in an unpublished discussion paper prepared for that workshop - see Bunt (2005).

- What criteria are relevant for identifying a particular class of dialogue acts? In designing a system of dialogue act types, what are the criteria for structuring the system?

2 The dialogue act concept

2.1 Defining dialogue acts

The term 'dialogue act' is sometimes used in a rather loose sense, to mean 'speech act, used in dialogue'. There are also more formal approaches, where dialogue acts are considered as concepts in the description or annotation of dialogue utterance meanings, and have a well-defined formal semantics. For instance, Bunt and Romary (2002) have proposed to view the meaning of an utterance as *the way in which the utterance is meant to change the information state of an interpreting system upon understanding the utterance.*

When analysing the meaning of a dialogue utterance, we can distinguish two fundamental aspects: (1) the semantic (or 'referential', 'propositional') content: the objects, events, situations, properties, relation, etc. that the utterance is about; and (2) the communicative function or purpose that the utterance has in the communication. Using these two aspects, a formal interpretation can be given to a dialogue act by viewing the combination of a communicative function and a semantic content as an operation that updates the information states of the dialogue participants in a certain way. This approach is known as the *information-state* or *context-change* approach to dialogue acts (see e.g. Traum and Larsson, 2003; Bunt, 2000; Cooper et al. 2003).

The use of update operations on information states (or contexts) does not mean that any logically possible type of update operation corresponds to a dialogue act. The whole idea of dialogue acts is that they are a way to characterize dialogue behaviour; therefore, dialogue acts should have an empirical basis: every dialogue act type should have some reflection in observable features of communicative behaviour. In other words, for every dialogue act type there are behavioural (linguistic) devices which a speaker can use in order to indicate the communicative function(s) of his contribution. This means that we have two criteria for distinguishing a particular type of dialogue act: (1) it corresponds to a specific context-changing effect; (2) the intended context-changing effect can be indicated by means of certain observable features of communicative behaviour.

2.2 Dialogue act types

There are often alternative possible ways to characterize the type of dialogue act performed by a given utterance. For example, the utterance *What did you say?*, can be characterized either as a feedback act, providing information about the speaker's understanding of the previous utterance, or as a question, and as such as different from the statement *I didn't hear what you said*, which may also be characterized as a feedback act.

Characterizations as a question or an inform relate more closely to the surface form of the utterance than the characterization as a FEEDBACK ACT. Characterizing these utterances as feedback acts takes into account what the question and the statement are about. Rather than choosing between these alternative characterizations, it seems more attractive to combine the two and characterize these utterances as FEEDBACK QUESTION and FEEDBACK INFORM, respectively.

It is common to speak of dialogue act *types* (or speech act types) as synonymous with: dialogue (speech) acts with a certain communicative function (illocutionary force); the case just considered shows that this may be inaccurate, for characterizing utterances as feedback acts is saying something about the type of their semantic content, rather than about their communicative function. Also, characterizing an utterance as a feedback

question says something both about semantic content type and communicative function.

Indirect speech acts may also be considered as allowing more than one characterization. An utterance such as *It's rather chilly in here* can be seen as intended to inform the addressee of something, but also as a request - to lit a fire, for instance. On the standard view, an indirect speech act occurs when a speaker uses an utterance to perform an additional speech act to the one that is 'directly' associated with the utterance in view of its appearance, as illustrated by *Do you know what time it is?* (as a request to tell what time it is) or *What time do you think it is?* as a reproach for being late.

To understand an utterance as being used to perform an indirect speech act, the addressee must reason with his understanding of the utterance as 'surface speech act', including its semantic content, and his knowledge of the context in order to construe an indirect interpretation as a speech which is appropriate in the given context.

When dialogue acts are viewed as context-changing operations, however, the notion of an indirect dialogue act comes to stand in a different light. Consider, for example, the direct and indirect questions *What time is it?* and *Do you know what time it is?* In both cases we may assume that the speaker wants to know what times it is, but when using the direct question the speaker makes the assumption that the addressee knows the answer to the question, whereas the indirect question does not carry this assumption - the utterance in that case expresses precisely that the speaker does not know whether the addressee knows the answer. If we follow the traditional analysis of indirect speech acts where the speaker is taken to perform the same speech act as an *extra* act, in addition to what is expressed directly, then we have to say that the indirect question creates in the addressee, among other things, the effect of the 'direct' question where the speaker wants to know whether the addressee knows what time it is, plus the effects of the indirectly expressed question where the speaker wants to know what time it is. This combination of beliefs would clearly be inconsistent, however. It would therefore be wrong to analyse the indirect question as the direct question plus an additional question. Instead, the indirect question associated with should be analysed as expressing the speaker's wish to obtain the information what time it is, *without* also expressing the expectation that the addressee is able to tell that. This makes the indirect question a (slightly) different type of dialogue act than the direct question.

Similar analyses apply to other indirect dialogue acts, such as indirect requests.

Theories of dialogue acts or communicative acts often emphasize the multifunctionality of dialogue utterances, i.e., the phenomenon that an utterance can have several functions at the same time (se e.g. Allwood, 2000). This is also reflected in some dialogue act annotation schemas, such as DAMSL (Allen and Core, 1997), which allow the assignment of multiple dialogue act tags to an utterance. One of the reasons for the multifunctionality of utterance is that it can have an effect related to various dimensions of the communication process, such as exchanging task-related information, giving feedback, and managing the interaction.

2.3 Uses of dialogue acts

Dialogue acts (DAs) have been used for several different purposes: to support conceptual analysis of natural human dialogue; as building blocks in the interpretation and generation of utterances in a dialogue system; to annotate dialogues, either manually or automatically; or to define the inter-agent communication between software agents; see e.g. FIPA (2002). Each of these applications brings specific constraints and requirements. Here, we

only consider the use of dialogue acts for tagging, and its implication for to the design of a well-structured system of dialogue acts.

When very small sets of tags are used, such as the LINLIN tag set (Ahrenberg, Dahlbäck & Jönsson, 1995) or the HCRC tag set (Carletta et al., 1996; Isard & Carletta, 1995), then there is little need to be concerned with its organization, but larger tag sets, such as those of DAMSL or DIT (see e.g. Keizer, 2003), call for a well-motivated structure to support annotators' work. For the ISO effort to develop a registry of standardized concepts for semantic annotation, it is moreover worth taking into account that the specification an *exhaustive* tag set for all domains and all purposes is hard to imagine. Explicit performatives, for instance, form an open class of dialogue communicative functions. Also, degrees of granularity in dialogue act distinctions are often possible. It therefore seems best to design a structured set of tags, with a clear, well-motivated structure, containing a number of obviously needed instances in the various categories, and with clear principles for how to add tags to the set as may be needed for specific domains or specific purposes. Such a system is what we suggest to call an 'open taxonomy'. Moreover, we propose to structure such a taxonomy according to the intuitive notion of 'dimensions of communication', mentioned above in relation to the multifunctionality of dialogue contributions.

3 Dimensions in dialogue act assignment

3.1 Formal concepts

Dimensions of communication are, intuitively, different aspects of the communication process that can be addressed independently and simultaneously by means of dialogue acts.

As an example of a dimension, consider the turn-taking system. For a dialogue participant A, the following situations may arise:

1. A has the turn, i.e. he is in a position to make a contribution to the dialogue. The following cases may arise:
   (a) A uses the turn and makes his contribution. In this case he does not have to perform any turn management action.
   (b) His turn is contested: dialogue partner B is trying to get the turn. The following situations may occur:
      i. A wants to keep the turn. The efforts that he makes in order to achieve that, constitute a TURN KEEPING act.
      ii. A is willing to concede the turn. The act of indicating to B that B may take the turn, constitutes a TURN GIVING act.

2. (a) B has the turn and is using it. If A is happy with that, he does not have to perform any turn management action.
   (b) B has the turn and is using it. If A wants to get the turn, without waiting until B concedes it, A's efforts to get it constitute a TURN GRABBING act.
   (c) B is offering A an opportunity to take the turn.
      i. If A seizes the opportunity and takes the turn, then that constitutes a TURN TAKING act.
      ii. If A is not willing to accept the turn, his behaviour that indicates that is a TURN REFUSAL act.

This example shows that a dialogue agent may perform one of five possible turn management acts, but never more than one: the alternatives within a dimension are mutually exclusive.

In general, dimensions are independent sets of features such that per dimension only one value may be assigned for an object that is characterized in the multidimensional space. To formalize this notion, we clearly need a formal device for assigning values to the objects to be characterized; in the case of dialogue annotation, that is a formal device of assigning annotation tags to the 'markables' to be annotated. We therefore introduce a *dialogue act assignment system* as follows.

Definition 1: A **Dialogue act assignment system** is a 4-tuple $A = < D, f, C, T >$ where D is a set of (simple) dialogue act tags, f is a function assigning tags to utterances (which may be simple elements of D, or complex structures built from D elements), C is a set of constraints on admissible combinations of tags, and T is a set of additional labels that f may assign to utterances – T contains such labels as *inaudible* and *abandoned*.

It may be noted that the DAMSL annotation system speaks of 'layers' in annotations as well as of multidimensionality, and seems to use these terms as synonyms. One of these layers/dimensions is called Communicative Status, and contains such tags as *uninterpretable* and *abandoned*, which seems better modelled as part of the annotation system than as a dimension in a set of dialogue act tags. (And perhaps DAMSL's 'Other Level' tags are best treated in this way as well.)

To reflect the multifunctionality of dialogue contributions, the DA assignment function should be allowed to assign sets of tags to utterances, where the elements of the set correspond to different dimensions of communication. To this end, the DA tag set may be organized as a taxonomy, i.e. as partitioned into named subsets such that the assignment function associates at most one tag per dimension with any given utterance. More formally:

Definition 2: A **multidimensional dialogue act assignment system** is a 4-tuple $A = < D, f, C, L >$ where $D = \{D_1, D_2, ..D_m\}$ is a dialogue act taxonomy with 'dimensions' $D_1, D_2, ..D_m$ and where the combination constraints C allow a dialogue utterance to be assigned a tag in each of the dimensions, but never more than one tag per dimension.

We consider this definition as capturing the essence of a multidimensional system. Another aspect is the independence of the assignment of a tag in one dimension from the tags in other dimensions. This is captured by the following definition of independence:

Definition 3: Two dimensions in a multidimensional annotation system are **independent** if any pair of tags from the two dimensions is admissible.

Definition 4: If any two dimensions in a multidimensional dialogue act assignment system are independent, then the system is called **orthogonal**.

Orthogonality is not to be taken as a strictly necessary requirement of a multidimensional system (it does not seem realistically feasible for DA tagging), but it is desirable to be as much orthogonal as possible (and thus to keep the set of constraints C as simple as possible).

It may be noted that we defined a dialogue act taxonomy as simply a partitioned set ot tags, thereby excluding the possibility of a taxonomy to have several levels. The reason for this choice is that a set of dimensions is itself not a dimension, according to Definition 2, since it would give rise to multiple tags from that dimension set. Still, it is convenient to have more than one level in a DA taxonomy, for grouping a number of dimensions under a more general name, like 'interaction management'. To distinguish such a grouping from dimensions proper, we propose to use the term *layer* with this definition: as a set of dimensions or, recursively, a set of layers, thereby making a clear distinction

between layers and dimensions. We will incorporate this notion of layer in Definition 5 below.

3.2 Multidimensional dialogue act tags

We noted above that an attractive way to characterize an utterance may be as a pair like FEEDBACK QUESTION, consisting of the name of a dimension (FEEDBACK) and the name of a communicative function (QUESTION). This suggests that DA tags may be pairs. On the other hand, characterizing an utterance as a TURN KEEPING act does not require a second element, since the turn keeping function is necessarily concerned with the dimension of turn management. A question, by contrast, can be about any type of information and therefore relate to any interaction dimension. We therefore propose to classify communicative functions as being either *general-purpose* or *dimension-specific*. A DA tag is then either a pair, consisting of a general-purpose function and a dimension. or a single dimension-specific function. This leads to the following modified definition of a multidimensional dialogue act assignment system, to which we have also added the notion of layers:

Definition 5: A **layered multidimensional dialogue act assignment system** is a 7-tuple $A = < GP, DS, D, f, C, L, T >$ where GP is a set of general-purpose communicative function names, DS is a taxonomy of dimension-specific communicative function names, D is the taxonomy of dimension names that mirrors the DS taxonomy, L is a set of layers (i.e., set of (sets of...) dimensions of D, and where f, C and T are as before, except that f is now a function from utterances to sets of tags (or labels from T), each tag being either an element from DS or a pair $< g, d >$ with $g \in GP$ and $d \in D$.

3.3 The DIT taxonomy

We have applied the concepts defined here and redesigned the DA taxonomy of DIT, adding some of the dialogue types distinguished in DAMSL. It should also be noted that some of the DIT categories of communicative functions for feedback and interaction management have been inspired by the work of Allwood et. al. (1994). (For the complete resulting taxonomy see `http:pi1294.uvt.nl/dit`). Slightly simplified, the taxonomy of dimension-specific functions in DIT looks as follows:

Task-Oriented Functions

- *Task/Domain-Specific Functions*: Hire, Fire, Appoint,...; Acquit, Condemn, Appeal,..
- *Task Management Functions*: ...

Dialogue Control Functions

- *Feedback Functions*
  - *Auto-Feedback Functions*: Overall Positive, Execution Negative, Evaluation Positive, ..., Perception (= Overall) Negative
  - *Feedback Elicitation Functions*: Evaluation, Execution
  - *Allo-Feedback Functions*: Allo-Overall Positive, Allo-Execution Negative, Allo-Evaluation Positive, ..., Allo-Perception (= Overall) Negative
- *Interaction Management Functions*
  - *Turn Management*: Turn accepting, Turn giving, Turn grabbing, Turn keeping, Turn refusal
  - *Time Management*: Stalling, Pausing
  - *Contact Management*: Contact check, Contact indication
  - *Topic Management*: Topic shift, Topic shift announcement,..
  - *Own Communication Management*: Error signaling, Retraction, Self-correction
  - *Partner Communication Management*: Completion, Partner correction
  - *Dialogue structuring*: Opening, Closing, DA announcement
- *Social Obligations Management Functions*
  - *Greeting*: Init-greeting, React-greeting
  - *Self-introduction*: Init-self-introduction, React-self-introduction
  - *Apology*: Apologising, Apology-downplay
  - *Gratitude*: Thanking, Thanking-downplay
  - *Valediction*: Init-goodbye, React-goodbye

It may be noted that general-purpose communicative functions can also be put into a

(partial) hierarchy, but the hierarchical relation in this case has a different significance from that between dimension-specific ones, namely as an expression of degree of specificity. For example, a confirmation is more specific than an answer, and a check is more specific than a question.

The DIT taxonomy is being used for annotation in the DIAMOND project (see http://pi1294.uvt.nl/diamond/), and in the PARADIME project (PARallel Agent-based Dialogue Management Engine) as part of the Dutch national IMIX project on interactive multimodal information extraction (see http://www.nwo.nl/nwohome.nsf/pages/NWOP_653H9J). Inter-annotator agreement data are not yet available, and are not easy to obtain for multidimensional annotation, but are one of the aims of these activities. Another major aim is the establishment of annotation guidelines, of which there is only a beginning, and annotation tools.

4 Multidimensional dialogue act annotation

Using a layered multidimensional DA assignment system for annotation raises several issues, some of which have been discussed by Larsson (1998), such as the consequences of multidimensional tags for measuring inter-annotator agreement. One obvious suggestion, that follows from the intended orthogonality of the various dimensions, is to consider calculating inter-annotator agreement *per dimension*. But even within a single dimension the issue of inter-annotator agreement is not a simple one in a DA system with hierarchical relations among communicative functions. If one annotator marks an utterance as a YES/NO-QUESTION concerned with domain information, and another as a CHECK, these annotators do not agree completely but cannot be said to disagree completely either. A more dramatic inter-annotator disagreement occurs for instance when one annotator thinks that an utterance does not have a function in a certain dimension, while an another annotator thinks it has.

This brings us to another issue that deserves further study: should it be assumed that every utterance in principle has a function in every dimension, if only implicitly? Every utterance could conceivably be said to have a feedback function, for instance, since it can always be taken to provide some information about the processing of previous utterances. Similarly, if we assume the existence of a topic management function that corresponds to continuing the dialogue without a change of topic, so 'TOPIC CONTINUATION' WOULD BE A DEFAULT VALUE IN THIS DIMENSION, then every utterance could be said to have a topic management function. So it seems that one consistent strategy for multidimensional tagging could be to assume the existence of default values for every dimension (except the domain and task management dimension) and to annotate each utterance with an 11-tuple of functions in the dialogue control dimensions. This is to be contrasted with the alternative of only annotating non-default values, and assuming a variable multiplicity of the tags to be assigned to utterances.

5 Related and future work

Most closely related to the work discussed in this paper is the effort of the Discourse Research Initiative that has resulted in the DAMSL annotation scheme (Dialogue Act Markup in Several Layers; see Allen & Core, 1997). While presented as a layered, multidimensional scheme, the DAMSL scheme is not based on clearly defined notions of dimension and layer.

In the communicative functions that it contains, the DAMSL scheme has much in common with the DIT taxonomy. An important difference is the much more elaborate and fine-grained set of functions for feedback and other aspects of dialogue control functions

that is available in DIT. For a more detailed comparison of the contents of DAMSL and DIT see Keizer (2003). Other surveys and comparative discussions of dialogue act annotation schemes and taxonomies include Larsson (1998); Lendvai (2004) and the MATE survey (Mengel et al., 2000); discussions of issues in the definition and use of dialogue acts include, in particular, Core & Allen (1997); Traum (1999); Stolcke et al. (2000) and Popescu-Belis (2005).

The latest version of the DIT taxonomy has been designed to include most of what is found in DAMSL, organized in a more systematic way. This should make it possible to develop annotation tools that are simpler than those of DAMSL, since the (approximate) orthogonality of the DIT dimensions allows annotators to more freely assign combinations of tags in various dimensions than is the case in DAMSL.

Bibliography

Ahrenberg, L., N.Dahlbäck & A.Jönsson (1995) Codings Schemes for Studies of Natural Language Dialogue. In: *Working Notes from the AAAI Spring Symposium*, Stanford.

Allen, J. et al. (1995) The TRAINS project: A case study in building a conversational planning agent. *J. of Experimental and Theoretical Artificial Intelligence 7*, 7–48.

Allen, J. & M. Core (1997) Draft of DAMSL: Dialogue Act Markup in Several Layers.

Allwood, J., J. Nivre & E. Ahlsén (1994) Semantics and Spoken Language Manual for Coding Interaction Management. Report from the HSFR project Semantik och talsprak.

Allwood, J. (2000) An activity-based approach to pragmatics. In H. Bunt & W. Black(eds.) *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics.* Amsterdam: Benjamins, 47–80.

Bunt, H. (2000) Dialogue pragmatics and context specification. In H. Bunt & W. Black(eds.) *Abduction, Belief and Context in Dialogue. Studies in Computational Pragmatics.* Amsterdam: Benjamins, 81–150.

Bunt, H. (2005) A Framework for Dialogue Act Specification. Paper presented at the 4th Joint ISO-SIGSEM Workshop on the Representation of Multimodal Semantic Information, Tilburg, 10-11 January 2005. Available at http://let.uvt.nl/research/ti/sigsem/wg

Bunt, H. & L. Romary (2002) Towards multimodal content representation. In K. Lee & K.S. Choi (eds.) *Proc. LREC 2002 Workshop on International Standards in Terminology and Linguistic Resources Management.* Paris: ELRA, 54–60.

Carletta, J., A. Isard, S. Isard, J.Kowtko & G. Doherty-Sneddon (1996) HCRC dialogue structure coding manual. Technical Report HCRC/TR-82.

Cooper, R., S. Ericsson, S. Larsson & I. Lewin (2003) An information state approach to collaborative negotiation. P. Kuhnlein, H. Rieser & H. Zeevat (eds.) *Perspectives on Dialogue in the new Millenium.* Benjamin, Amsterdam, 271–287.

Core & J. Allen (1997) Coding dialogues with the DAMSL annotation scheme.

FIPA (2002) FIPA SL Content Language Specification. Geneva: Foundation for Intelligent Physical Agents, Document No. SC000081.

Isard, A. & J. Carletta (1995) Transaction and action coding in the Map Task Corpus. Research Paper HCRC/RP-65.

Jurafsky, D., E. Shriberg & D. Biasca (1997) Switchboard SWBD-DAMSL Shallow-Discourse-Function-Annotation Coders Manual, Draft 13.

Jurafsky, D. and J.H. Martin (2000) *Speech and Language Processing*. Prentice-Hall.

Keizer, S. (2003) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks.* PhD Thesis, Twente University, Enschede.

Larsson, S. (1998) Coding Schemas for Dialog Moves. Unpublished paper; see http://www.ling.gu.se/ sl

Lendvai, P. (2004) Extracting information from spoken input. PhD Thesis, Tilburg University.

Mengel, A., L. Dybkjaer, L.Garrido, J.M. Heid, V.Pirelli, M.Poesio, S. Quazza, A. Schiffrin & C. Soria (2000) MATE Dialogue Annotation Guidelines. http://www.ims.uni-stuttgart.de/projekte/mate/mdag/

Popescu-Belis, A. (2005) Dialogue Acts: One or More Dimensions? ISSCO Working Paper 62, ISSCO, Geneva.

Stolcke, A et al. (2000) Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics 26:3*, 339 – 373.

Traum, D. (1999) Twenty Questions for Dialogue Act Taxonomies. *Proc. of Amstelogue '99.*

Traum, D. & S. Larsson (2003) The Information State Approach to Dialogue Management. In. R. Smith & J. van Kuppevelt (eds.) *Current and New Directions in Discourse and Dialogue.* Dordrecht: Kluwer, 325–353.

Dialogue Systems: Simulations or Interfaces?

Staffan Larsson
Dept. of linguistics
Göteborg University
SE 40530 Göteborg, Sweden
sl@ling.gu.se

Abstract

This paper raises the question of the aim and scope of formal research on dialogue. Two possible answers are distinguished – the "engineering" and the "simulation" view – and an argument against the soundness of the "simulation" position is reviewed. This argument centres on the (im)possibility of formalising the context (or "background") needed for human-level language understanding. This argument is then applied to formal dialogue research and some consequences are discussed.

1 Introduction

Although perhaps nowadays many researchers would be wary of subscribing to the view that a complete simulation of human language use is possible, the precise extent to which this goal is feasible (and desirable) is still an open question. A premise of this paper is that this is an important issue to discuss, and that such a discussion could be useful as a backdrop for formulating goals and methods for research on the formal semantics and pragmatics of dialogue.

In this paper, I raise the question of the aim and scope of research on dialogue systems and the formal and computational semantics and pragmatics of dialogue. (I will refer to this area of research as "formal dialogue" research.) I distinguish two possible answers – the "engineering" (or "interface") view and the "simulation" view – representing the most extreme positions taken in response to this question. I then review an argument against the soundness of the "simulation" position, in order to give an impression of the deep difficulties involved in achieving this goal. This argument centres on the (im)possibility of formalising the context (or "background") needed for human-level language understanding.

The contribution of the present paper is the explicit application of this argument to formal dialogue research and an attempt to draw out some consequences of the argument for this area of research. I argue that an intermediate position closer to the "engineering" view on formal dialogue research is both more useful and realistic. However, knowledge of human language use (both formal and informal) is still essential in this endeavour. A further important consequence of the argument is that since the "simulation" and "interface" research programs are in fact very different, it is important to be clear about the goal in any given piece of work in formal dialogue research.

2 Engineering vs. Simulation

The first view (the "engineering" position) claims that the purpose of formal dialogue research is ultimately one of interface engineering; to enable the building of better human-computer interfaces by incorporating (spoken) dialogue. The second answer (the "simulation" position) claims that the ultimate goal is a complete computational (implementable) theory of human language use and understanding. In reality, there is of course a continuum where individual researchers may take intermediate positions, take different positions depending on the situation, and/or assume that both goals converge and so there is no reason to take any position (which is, of course, itself a position). For example, one intermediate position might be to regard formal semantics and pragmatics as capturing (although in a more or less simplified manner) some aspects of human language use while deliberately ignoring or oversimplifying other aspects, and to regard dialogue systems as a possible area of application for such theories. Still, the issue remains as to what the ultimate goal of the research is.

3 Dialogue systems as interfaces

There is, I believe, a consensus in the formal dialogue research community that (spoken) dialogue has the potential of vastly improving on, or even replacing, available human-computer interface technology. There are good reasons for this optimism, as spoken dialogue is perhaps the most natural way for humans to communicate. As technologies become more complex, previous interfaces such as the command-line or menu-based graphical interfaces become increasingly unwieldy and impractical, and an interface based on the metaphor of intelligent conversational agents becomes increasingly attractive.

A common idea in formal research on dialogue is that there is a extensive, if not complete, overlap between research on human language use and research on conversational interfaces. To build good conversational interfaces it is important to develop, extend, formalise and implement theories of human language use. A very influential way of thinking about this overlap is the idea that a dialogue system should, as far as possible, be a *simulation of human language use*.

4 Dialogue systems as simulations

Can a machine be intelligent? Turing offered an operational definition of intelligence in the form of a test, which goes roughly like this: Test person A has a dialogue (via a text terminal) with B. A:s goal is to decide whether B is a human or a machine. If B is a machine and manages to deceive A that B is a human, B should be regarded as intelligent and able to think.

According to the Turing test, human intelligent behaviour is equivalent to the ability to carry out a dialogue using natural language. This means that in order to make a computer use natural language in the same way and on the same level as a human, it needs to be endowed with human-level intelligence. Interestingly, this also means that research on semantics and pragmatics of natural language has a central role in AI as a whole. In fact, if one takes a simulation view on formal dialogue research, this field becomes in a way equivalent to AI. Below, I will review an argument against AI, and attempt apply it more explicitly to research on dialogue systems and formal semantics and pragmatics.

4.1 GOFAI and formal research on dialogue

A lot of formal research on dialogue has, by way of inheritance or common ancestry, some central ideas in common with the classical AI approaches (sometimes referred to as Good Old-Fashioned AI, or GOFAI (Haugeland, 1985)). For example, the Information

State Update approach to dialogue management (Traum and Larsson, 2003) has a lot in common with GOFAI approaches such as SOAR (Laird et al., 1987). Symbolic representation and symbol manipulation remain important cornerstones in the way that problems are formulated and in the form of the solutions given. Starting out from the assumption that sentences in natural language can be given a formal semantics, the realisation that context plays a central part in language use has led to the idea of formalising the context so that it can be related to formal semantic representations of sentences and utterances. One reason for the use of formal techniques is simply that, so far, representation and symbol manipulation it seems to be the most (or even the only) workable method for dealing with many of the complex problems of natural language dialogue, e.g. ellipsis resolution, pronoun resolution, dialogue act recognition, keeping track of multiple topics, etc..

Often, computational dialogue researchers implement their theories either in limited toy examples, or as semi-functional dialogue system interfaces for small domains [1]. Dialogue systems based on symbolic computation definitely appears to be useful for improving on current human-computer interfaces, although only a major breakthrough of natural language dialogue interfaces would prove this conclusively. But is it also a step on the way towards human-level natural language understanding in computers?

[1] A recent paper (Bos, 2005) boldly claims to show that "it is possible to have a robust and wide-coverage system that generates semantic interpretations with relevant background knowledge from texts and perform first-order inferences on the result." However, the correctness (accuracy and adequacy) of these resulting representations have not yet been evaluated. As I understand Dreyfus' argument it would predict that the correctness would be low in most domains, although some success is possible in systematic domains (see Section 5.1).

4.2 Arguments against GOFAI

The position of so-called "weak AI" is, roughly, that computers can be made to act as if they were intelligent (Russell S, 1995). Independently of any "strong AI" claims as to whether such a computer would also be conscious, Dreyfus (1992) and others (e.g. Winograd and Flores (1987)) have put forward arguments against the possibility of weak AI, based on the philosophies of Heidegger, Merleau-Ponty, and the later Wittgenstein. As these arguments centre on the possibility of human-level understanding of language in computers, they are also very relevant to the present discussion. This section briefly reviews Dreyfus' arguments; unfortunately, space restrictions make it hard to do justice to the argumentation. For the full account, see Dreyfus (1992).

Some well-known problems in GOFAI are computational complexity of logical inference in real-time resource-bounded applications, planning for conjunctive goals, plan recognition, incompleteness of general FOL reasoning (not to mention modal logic), and the frame problem (Haugeland, 1987). However, as humans we don't tend to encounter these problems in our everyday life (unless, of course, we happen to be AI researchers). Dreyfus asks rhetorically whether it is is possible that all these problems have a common cause? Well, they all seem to be related to symbolic representations and symbol manipulation. The idea that understanding and thinking is forming and using symbolic representations is an old one, going back at least to Descartes[2] and reformulated in (Newell and Simon, 1963) as the "physical symbol hypothesis". According to this idea, intelligent behaviour can be captured by a system that reasons logically from a set of formal and context-independent facts and rules[3].

[2] Dreyfus argues that the idea of formalising human reasoning goes back at least to Plato.

[3] Although facts in themselves may (purport to) represent

Against this, Dreyfus argues that human behaviour is essentially non-formal. Human behaviour based on our everyday common-sense background understanding, which allows us to experience what is currently relevant, deal with things and people in everyday situations, and understand natural language. The background involves (among other things) utterance situation, ongoing activities, relevant institutions, and cultural settings. In its widest sense, the background involves all of human culture and experience as it is passed down through generations in social interaction. Dreyfus argues that the background has the form of dispositions, or informal know-how. It is thus a form of skill rather than propositional knowing-that – inarticulate, and to some extent pre-conceptual.

To achieve GOFAI, this know-how, along with the interests, feelings, motivations, and bodily capacities that go to make a human being, would have to be conveyed to the computer as knowledge in the form of a huge and complex belief system. Indeed, work in this direction has been going on for several years, e.g. in the CYC project (Lenat and Guha, 1990). Dreyfus argues, however, that *the background cannot be formalised*; there are no reasons to think that humans represent and manipulate the background explicitly or that this is at all possible even in principle. To quote from Dreyfus (1992), p. 3: "...understanding requires giving the computer a background of common sense that adult human beings have by virtue of having bodies, interacting skillfully with the material world, and being trained into a culture.". This background enables humans to, among other things, skillfully cope with changing events and motivations, project understanding onto new situations, and understand social innovations – someone may do something that has not so far counted as appropriate, and have it recognized in retrospect as having been just the right thing to do

Even so, there is a grain of truth to the information-processing idea. When something goes wrong – when there is a *breakdown* in some activity – we need to reflect and reason, and may have to learn and apply formal rules. However, it is a mistake to read these rules back into the normal situation and appeal to such rules for a causal explanation of skillful behaviour. Similarly, when learning new skills we might start from a set of rules and facts, but as we progress from novice to expert, the rules are replaced by embodied skills.

4.3 Non-symbolic approaches to AI and dialogue

According to Dreyfus, since around 1986 GOFAI has become less popular, partly in response to arguments from critics such as Dreyfus. A widely-used textbook on AI acknowledges admits that "[m]any of the issues Dreyfus discusses (...) are now widely accepted as important aspects of intelligent agent design." (Russell S, 1995). Instead, there has been an increasing focus on non-symbolic or semi-symbolic methods such as connectionism, embodied interactive automata, reinforcement learning, probabilistic methods, etc.. Mirroring this move in cognitive science is an increased focus in computational linguistics, including formal dialogue research, on semi-symbolic statistical and machine-learning methods.

Space restrictions prohibit a thorough discussion of whether non-symbolic methods can be used to overcome the problem of equipping computers with the background necessary for human-level language understanding. Suffice to say that Dreyfus argues (convincingly, in my view) that non-symbolic approaches to AI face the same basic problem as the symbolic approach. True, non-

context, and rules may make reference to such facts, the facts and rules themselves are represented in a formal language or programming language which does not depend on context for its interpretation.

symbolic systems do not themselves contain a formal description of background. However, they cannot be built and trained without a pre-existing formalisation of background knowledge.

To put it very briefly, the reason is that even these approaches require a formalised context in order to set up the training data in a way that will allow a system to learn anything useful from it. This requires that humans interpret the context in terms of its relevant features before it can be fed to the computer. To quote from a recent conference call[4]:

> As experience with machine learning for solving natural language processing tasks accumulates in the field, practitioners are finding that feature engineering is as critical as the choice of machine learning algorithm, if not more so. Feature design, feature selection, and feature impact ... significantly affect the performance of systems and deserve greater attention.

This process of "feature engineering" is far from an innocent "preparation" of data; rather, it is a crucial step of *pre-digesting* the data by noting the relevant aspects of a situation to a problem at hand and embodying this interpretation in a formal description that the computer can then manipulate. The quote above indicates that there seems to be a growing realisation within the AI community that "feature engineering" is crucial for natural language processing in computers.

The ability to see the relevant features of a situation is not present in computers, Dreyfus argues, since it crucially requires a commonsense background. So, one might wonder, how do humans manage to learn this background? As already indicated in the quote above, they are able to do so by virtue of having bodies, interacting skillfully with the material world, and being trained into a culture. Language is, simply, very deeply interconnected with human life. Unless we are able to build computers which have (human or human-like) bodies, and which are trained into a culture through social practices of human society (involving being born by parents, going through childhood and adolescence and growing up and learning personal responsibility, social interaction, making friends, and establishing an identity, and all the other things that make up human life), the argument implies, no machine will ever pass the Turing test[5].

It must be stressed that this is not a "knock-down argument" proving conclusively that weak AI is impossible; no such claims are made by Dreyfus. For me personally, it served to point out that achieving human-level language understanding in computers might be much harder than I had previously thought, and that the research methods involved in pursuing this goal may be quite different from the methods appropriate for the design of dialogue systems as human-computer interfaces.

5 Formal dialogue research and dialogue systems design

If we accept the argument that "the background is not formalizable" and that computers will never achieve human-level language understanding, does it follow that formal and computational research on dialogue and dialogue systems is useless? Of course not; it provides (as already mentioned) a great potential for improving on human-computer interaction. But granted this, has theories of human language use now been shown to be of no use to research on human-computer dialogue? Again, of course not. For one thing, if we want dialogue systems that are reasonably human-like in their behaviour, these systems will need to be designed on the basis of theories of human language. But this does not require that these theories have to be formal descriptions on human language use and

[4] `http://research.microsoft.com/~ringger/FeatureEngineeringWorkshop/`

[5] Indeed, it could be argued that the power of the Turing test rests on this intuition that the ability to carry on a dialogue in natural language truly requires *human* intelligence.

cognition, nor of implementations of them as (even partial) simulations. Instead, we may use these theories as providing important clues about how to best build dialogue systems. Firstly, we may observe regularities in dialogue that can serve as the basis for formal representations. Second, non-formal theories of those aspects of language use which resist formalisation can be used as a basis for design of aspects of dialogue systems that do not need to be modelled by the system itself.

5.1 Formal theories as systematic domains

Arguing against the possibility of human-level intelligence and language understanding by computers (along similar lines as Dreyfus), Winograd points out that computers are nevertheless useful tools in areas of human activity where formal representation and manipulation is crucial, e.g. word processing. In addition, many practical AI-style applications do not require human-level understanding of language. In such cases, it is possible to develop useful systems that have a limited repertoire of linguistic interaction. There are regularities in conversational behaviour ("domains of recurrence"), and that on the basis of such regularities it is possible for e.g. a researcher to create[6] so-called *systematic domains*. That is, a set formal representations that can be used in a system and that embodies the researcher's interpretation of the situation in which the system will function.

Note that providing formal rules for describing behaviour does not necessarily imply that similar rules are represented in humans. If we accept that human behaviour is essentially non-formalizable, formal rules will always be, at best, rough representational approximations of the non-representational know-how embodied in humans[7].

[6]Winograd stresses that this is a creative process, rather than one of mere observation.

[7]Compare the case where a statistical speech recognition grammar is trained on the output of a formal grammar. The SLM can then be subjected to machine learning which will subtly modify its behaviour in ways that could not be expressed in the rules of the original grammar. The difference is only that in attempting to formally describe language use, we are abstracting the hard-edged rules from embodied behaviour, rather than starting with the rules. Humans may have learned their behaviour with or without starting from explicit rules; however, human behaviour is always shaped by biological factors and social interactions that are not available to computers for reasons already discussed.

Semantics[8] is not a focus of Winograd's formal analysis, presumably because Winograd believes that language understanding is not amenable to formal analysis (see also citewindograd:fulcrum). However, even if one accepts the arguments such as those above, I believe that the idea of systematic domains also applies to semantics. That is, for certain "semantically systematic" task domains it is indeed possible to provide a formal semantics, e.g. in the form of a formal ontology and formal representations of utterance contents. Arguably, semantics is more closely related to specific activities than is pragmatics, since semantics involves the entities and relations which are relevant in a given activity or task domain. This means that the question of whether a task domain can be usefully captured in formal semantic must be answered for each task domain or task domain type individually.

5.2 Non-formal theory and dialogue systems design

As mentioned above, non-formal theories of those aspects of language use which resist formalisation can be used as a basis for design of aspects of dialogue systems that do not need to be modelled by the system itself. For example, it is likely that any speech synthesizer voice has certain emotional or other cognitive connotations; it might sound silly, angry, etc.. It is extremely difficult, if not impossible, to design a completely neutral voice. However, if we have some idea of how different voices are perceived (or perhaps even how different

[8]I am not claiming that there is a strict division between semantics and pragmatics.

aspects of speech synthesis correlate with the perceived "personality" of the voice), we can use this (informal) knowledge to provide a dialogue system application with an appropriate voice for that application.

6 Conclusions

This paper has distinguished two extreme views on formal dialogue research; the "engineering" view and the "simulation view". On the basis of Dreyfus' criticism of AI, I have argued that the simulation view, at least in its most extreme form, is probably untenable as an explicit or implicit research goal. I have also argued, on the basis of Winograd's ideas, that formal dialogue research may nevertheless be useful for improving dialogue systems in limited domains and with limited linguistic capabilities. I have also suggested that formal theories of language use are limited in scope and should be complemented by non-formal theories in the design of dialogue system interfaces.

Domains of language use that may be susceptible to formalisation (i.e. creation of systematic domains) can be roughly divided into pragmatic and semantic domains. Pragmatic domains include e.g. aspects of dialogue management such as turntaking, feedback and grounding, referent resolution, and topic management and sequencing. Issues related to semantic domains concern e.g. application-specific ontologies and the fine-grainedness and expressivity of the formal semantic representation required for a domain or group of domains (e.g. database search, device programming, collaborative planning). The general issue of how to determine whether a task domain is amenable to formal semantic description is one that deserves to be further investigated, as well as the closely related issue of how to extract a formal description from available data of the domain, e.g. transcripts of dialogues. A third related issue is how to decide, for a given task domain, what level of sophistication is required by a formal semantic framework in order for it to be useful in that domain. In some domains, simple feature-value frames may be sufficient while others may require something along the lines of situation semantics, providing treatments of intensional contexts etc.[9]

The question is still open exactly how far it is possible to go in the formal description of phenomena related to language use, and the only way to find out is to by trial-and-error (i.e., research). I'm thus by no means arguing that one should stop trying to extend the coverage of formal semantics and pragmatics, rather that one might be well-advised to keep in mind the following points:

1. Formal theories of language use should be regarded as the result of a creative process of finding regularities in language use as a basis for the construction of formal representations that can be used in dialogue systems to open up new possibilities for human activity.

2. Even though some aspects of language use may indeed be susceptible to formal description, this does not mean that human language use actually relies on such formal descriptions represented in the brain or elsewhere.

3. Repeated failures to formally capture some aspect of human language may be due to the limits of formal theory when it comes to human language use, rather than to some aspect of the theory that just needs a little more tweaking.

Even though the arguments against AI cited above constitute, in my view, good reasons

[9] As a special case, when adapting a dialogue system to function as an interface to an existing program, there is already a formalised domain in the form of the actions, entities etc. of the existing interface. In such cases, it is usually sufficient with a very basic semantic formalism. In addition, existing interfaces such as menu-based GUIs can provide a readily available formalisation of useful conversational structures, e.g. by converting menu systems into dialogue plans.

to be very sceptical about the possibility of simulating human language use, it would certainly be premature to completely abandon this research[10]. However, as this (as has been argued above) constitutes a rather different project than that of building good interfaces and tools for systematic domains, it would be good practice to explicitly state what the goals of a certain piece of research are in case this is not obvious.

Being clear about this can also serve to motivate different research strategies and to estimate the validity of different types of argument used when presenting the research. For research taking the "engineering" view, methodologies should be concerned not primarily with how human cognition and language use works, but rather with designing and engineering of useful and flexible human-computer dialogue interfaces. If interface engineering is liberated from concerns related to simulation, it can instead be focused on the creation of new forms of human-computer (and computer-mediated) communication, adapted to and exploring the respective limitations and strengths of humans and computers. Of course, knowledge about human language use is relevant here as well (as a source of inspiration, if nothing else) but is not regarded as an end in itself.

References

Johan Bos. 2005. Towards wide-coverage semantic interpretation. In *Proceedings of Sixth International Workshop on Computational Semantics (IWCS-6)*.

Rudi Cilibrasi and Paul Vitanyi. 2005. Automatic meaning discovery using google. Technical report, University of Amsterdam, National ICT of Australia.

Hubert Dreyfus. 1992. *What computers still can't do*. The MIT Press.

John Haugeland. 1985. *Artificial intelligence: the very idea*. Massachusetts Institute of Technology, Cambridge, MA, USA.

John Haugeland. 1987. An overview of the frame problem. In Z. W. Pylyshyn, editor, *The Robot's Dilemma: The Frame Problem in Artificial Intelligence*, Theoretical Issues in Cognitive Science, pages 77–93. Ablex, Norwood, New Jersey.

J. Laird, A. Newell, and P. Rosenbloom. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64.

D. Lenat and R.V. Guha. 1990. *Building Large Knowledge-Based Systems - Representation and Inference in the Cyc Project.* Addison-Wesley, Reading, Massachusetts.

Alan Newell and Herbert Simon. 1963. GPS: A program that simulates human thought. In Edward Feigenbaum and Jerome Feldman, editors, *Computers and Thought*, pages 279–293. McGraw-Hill.

Norvig P Russell S. 1995. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Englewood Cliffs, New Jersey.

Luc Steels and Paul Vogt. 1997. Grounding adaptive language games in robotic agents. In C. Husbands and I. Harvey, editors, *Proceedings of the Fourth European Conference on Artificial Life*. The MIT Press.

David Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In Ronnie Smith and Jan Kuppevelt, editors, *Current and New Directions in Discourse & Dialogue*. Kluwer Academic Publishers.

T Winograd and F. Flores. 1987. *Understanding Computers and Cognition*. Addison-Wesley Professional.

Terry Winograd. 1985. Moving the semantic fulcrum. *Linguistics and Philosophy*, 8(1):91–104.

[10] Some recent developments in non-symbolic or semi-symbolic AI that can be seen as addressing the problems discussed above (e.g., overcoming the need for formal descriptions of context, produced by humans) include simulated evolution of linguistic communication between robots (Steels and Vogt, 1997), and mining for semantic relations on the web (Cilibrasi and Vitanyi, 2005). The former approach does not, however, aim to evolve *human* language, and there are good reasons for this (Winograd, 1985). The latter approach is certainly interesting, but there are reasons to be sceptical as to how far it can get, as it does not address many of the issues raised by Dreyfus, e.g. the importance of embodiment and social interaction.

Multimodal Dialogue System Grammars*

Björn Bringert, **Peter Ljunglöf**, **Aarne Ranta**
Department of Computing Science
Chalmers University of Technology
and Göteborg University
{bringert,peb,aarne}@cs.chalmers.se

Robin Cooper
Department of Linguistics
Göteborg University
cooper@ling.gu.se

Abstract

We describe how multimodal grammars for dialogue systems can be written using the Grammatical Framework (GF) formalism. A proof-of-concept dialogue system constructed using these techniques is also presented. The software engineering problem of keeping grammars for different languages, modalities and systems (such as speech recognizers and parsers) in sync is reduced by the formal relationship between the abstract and concrete syntaxes, and by generating equivalent grammars from GF grammars.

1 Introduction

We are interested in building multilingual multimodal grammar-based dialogue systems which are clearly recognisable to users as the same system even if they use the system in different languages or in different domains using different mixes of modalities (e.g. in-house vs in-car, and within the in-house domain with vs without a screen for visual interaction and touch/click input). We wish to be able to guarantee that the functionality of the system is the same under the different conditions.

Our previous experience with building such multilingual dialogue systems is that there is a software engineering problem keeping the linguistic coverage in sync for different languages. This problem is compounded by the fact that for each language it is normally the case that a dialogue system requires more than one grammar, e.g. one grammar for speech recognition and another for interaction with the dialogue manager. Thus multilingual systems become very difficult to develop and maintain.

In this paper we will explain the nature of the Grammatical Framework (GF) and how it may provide us with a solution to this problem. The system is oriented towards the writing of multilingual and multimodal grammars and forces the grammar writer to keep a collection of grammars in sync. It does this by using computer science notions of abstract and concrete syntax. Essentially abstract syntax corresponds to the domain knowledge representation of the system and several concrete syntaxes characterising both natural language representations of the domain and representations in other modalities are related to a single abstract syntax.

GF has a type checker that forces concrete syntaxes to give complete coverage of

*This project is supported by the EU project TALK (Talk and Look, Tools for Ambient Linguistic Knowledge), IST-507802

the abstract syntax and thus will immediately tell the grammar writer if the grammars are not in sync. In addition the framework provides possibilities for converting from one grammar format to another and for combining grammars and extracting sub-grammars from larger grammars.

2 The Grammatical Framework and multilingual grammars

The main idea of Grammatical Framework (GF) is the separation of abstract and concrete syntax. The abstract part of a grammar defines a set of abstract syntactic structures, called abstract terms or trees; and the concrete part defines a relation between abstract structures and concrete structures.

As an example of a GF representation, the following abstract syntax tree represents a possible user input in our example dialogue system.

```
GoFromTo
  (PStop Chalmers)
  (PStop Valand)
```

The English concrete syntax relates the query to the string

```
I want to go from Chalmers
to Valand
```

The Swedish concrete syntax relates it to the string

```
Jag vill åka från Chalmers
till Valand
```

The strings are generated from the tree in a compositional rule-to-rule fashion. The generation rules are automatically inverted to parsing rules.

The abstract theory of Grammatical Framework (Ranta, 2004) is a version of dependent type theory, similar to LF (Harper et al., 1993), ALF (Magnusson and Nordström, 1994) and COQ (Coq, 1999). What GF adds to the logical framework is the possibility of defining concrete syntax. The expressiveness of the concrete syntax has developed into a functional programming language, similar to a restricted version of programming languages like Haskell (Peyton Jones, 2003) and ML (Milner et al., 1997).

The separation between abstract and concrete syntax was suggested for linguistics in (Curry, 1963), using the terms "tectogrammatical" and "phenogrammatical" structure. Since the distinction has not been systematically exploited in many well-known grammar formalisms, let us summarize its main advantages.

Higher-level language descriptions The grammar writer has a greater freedom in describing the syntax for a language. As illustrated in figure 1, when describing the abstract syntax he/she can choose not to take certain language specific details into account, such as inflection and word order. Abstracting away smaller details can make the grammars simpler, both to read and understand, and to create and maintain.

Multilingual grammar writing It is possible to define several different concrete syntax mappings for one particular abstract syntax. The abstract syntax could e.g. give a high-level description of a family of similar languages, and each concrete mapping gives a specific language instance, as shown in figure 2.

This kind of multilingual grammar can be used as a model for interlingual translation between languages. But we do not have to restrict ourselves to only multilingual grammars; different concrete syntaxes can be given for different modalities. As an example, consider a grammar for displaying time table information. We can have one concrete syntax for writing the information as plain text, but we could also present the information in the form of a table output as a LaTeX file or in Excel format, and a third possibility is to output the information in a format suitable for speech synthesis.

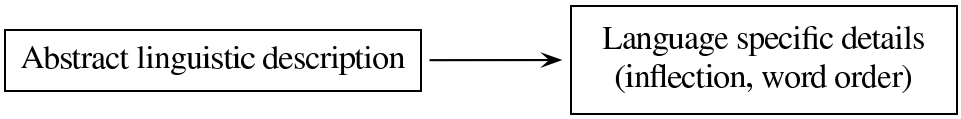


Figure 1: Higher-level language descriptions

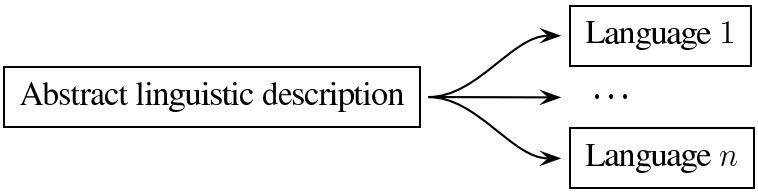


Figure 2: Multilingual grammars

Several descriptional levels Having only two descriptional levels is not a restriction; this can be generalized to as many levels as is wanted, by equating the concrete syntax of one grammar level with the abstract syntax of another level. As an example we could have a spoken dialogue system with a semantical, a syntactical, a morphological and a phonological level. As illustrated in figure 3, this system has to define three mappings; *i*) a mapping from semantical descriptions to syntax trees; *ii*) a mapping from syntax trees to sequences of lexical tokens; and *iii*) a mapping from lexical tokens to lists of phonemes.
This formulation makes grammars similar to transducers (Karttunen et al., 1996; Mohri, 1997) which are mostly used in morphological analysis, but have been generalized to dialogue systems by (Lager and Kronlid, 2004).

Grammar composition A multi-level grammar as described above can be viewed as a "black box", where the intermediate levels are unknown to the user. Then we are back in our first view as a grammar specifying an abstract and a concrete level together with a mapping. In this way we can talk about *grammar composition*, where the composition $G_2 \circ G_1$ of two grammars is possible if the abstract syntax of G_2 is equal to the concrete syntax of G_1.

If the grammar formalism supports this, a composition of several grammars can be precompiled into a compact and efficient grammar which doesn't have to mention the intermediate domains and structures. This is the case for e.g. finite state transducers, but also for GF as has been shown by (Ranta, 2005).

Resource grammars The possibility of separate compilation of grammar compositions opens up for writing *resource grammars* (Ranta, 2005). A resource grammar is a fairly complete linguistic description of a specific language. Many applications do not need the full power of a language, but instead want to use a more well-behaved subset, which is often called a *controlled language*. Now, if we already have a resource grammar, we do not even have to write a concrete syntax for the desired controlled language, but instead we can specify the language by mapping structures in the controlled language into structures in the resource grammar, as shown in figure 4.

3 Extending multilinguality to multimodality

Parallel multimodality *Parallel multimodality* is a straightforward instance of multilinguality. It means that the concrete syntaxes associated with an abstract syntax are not just different natural languages, but different representation modalities, encoded by language-like notations such as graphic

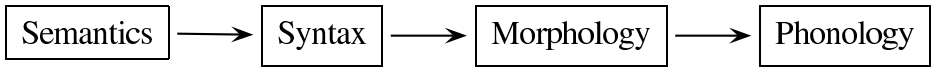


Figure 3: Several descriptional levels

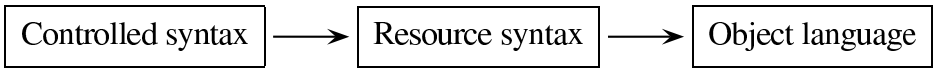


Figure 4: Using resource grammars

representation formalisms. An example of parallel multimodality is given below when a route is described, in parallel, by speech and by a line drawn on a map. Both descriptions convey the full information alone, without support from the other.

This raises the dialogue management issue of whether all information should be presented in all modalities. For example, in the implementation described below all stops are indicated on the graphical presentation of a route whereas in the natural language presentation only stops where the user must change are presented. Because GF permits the suppression of information in concrete syntax, this issue can be treated on the level of grammar instead of dialogue management.

Integrated multimodality *Integrated multimodality* means that one concrete syntax representation is a combination of modalities. For instance, the spoken utterance *"I want to go from here to here"* can be combined with two pointing gestures corresponding to the two *"here"*s. It is the two modalities in combination that convey the full information: the utterance alone or the clicks alone are not enough.

How to define integrated multimodality with a grammar is less obvious than parallel multimodality. In brief, different modality "channels" are stored in different fields of a record, and it is the combination of the different fields that is sent to the dialogue system parser.

4 Proof-of-concept implementation

We have implemented a multimodal route planning system for public transport networks. The example system uses the Göteborg tram/bus network, but it can easily be adapted to other networks. User input is handled by a grammar with integrated speech and map click modalities. The system uses a grammar with parallel speech and map drawing modalities. The user and system grammars are split up into a number of modules in order to simplify reuse and modification.

The system is also multilingual, and can be used in both English and Swedish. For every English concrete module shown below, there is a corresponding Swedish module. The system answers in the same language as the user made the query in.

In addition to the grammars shown below, the application consists of a number of agents which communicate using OAA (Martin et al., 1999). The grammars are used by the Embedded GF Interpreter (Bringert, 2005) to parse user input and generate system output.

4.1 Transport network

The transport network is represented by a set of modules which are used in both the query and answer grammars. Since the transport network is described in a separate set of modules, the Göteborg transport network may be replaced easily. We use **cat** judgements to declare categories in the abstract syntax.

```
abstract Transport = {
 cat
  Stop;
```

```
}
```

The Göteborg transport network grammar extends the generic grammar with constructors for the stops. Constructors for abstract syntax terms are declared using **fun** judgements.

```
abstract Gbg = Transport ** {
 fun
  Angered : Stop;
  AxelDahlstromsTorg : Stop;
  Bergsjon : Stop;
  ...
}
```

4.2 Multimodal input

User input is done with integrated speech and click modalities. The user may use speech only, or speech combined with clicks on the map. Clicks are expected when the user makes a query containing *"here"*.

Common declarations The `QueryBase` module contains declarations common to all input modalities. The `Query` category is used to represent the sequentialization of the multimodal input into a single value. The `Input` category contains the actual user queries, which will have multimodal representations. The `Click` category is also declared here, since it is used by both the click modality and the speech modality, as shown below.

```
abstract QueryBase = {
 cat
  Query;
  Input;
  Click;
 fun
  QInput : Input -> Query;
}
```

Since `QueryBase` is language neutral and common to the different modalities, it has a single concrete syntax. In a concrete module, **lincat** judgements are used to declare the linearization type of a category, i.e. the type of the concrete representations of values in the category. Note that different categories may have different linearization types. The concrete representation of abstract syntax terms is declared by **lin** judgements for each constructor in the abstract syntax.

Values in the `Input` category, which are intended to be multimodal, have records with one field per modality as their concrete representation. The `s1` field contains the speech input, and the `s2` field contains the click input. Terms constructed using the `QInput` constructor, that is sequentialized multimodal queries, are represented as the concatenation of the representations of the individual modalities, separated by a semicolon.

```
concrete QueryBaseCnc of QueryBase = {
 lincat
  Query = { s : Str };
  Input = { s1 : Str; s2 : Str };
  Click = { s : Str };
 lin
  QInput i = { s = i.s1 ++ ";" ++ i.s2 };
}
```

Click modality Click terms contain a list of stops that the click might refer to:

```
abstract Click = QueryBase ** {
 cat
  StopList;
 fun
  CStops : StopList -> Click;
  NoStop : StopList;
  OneStop : String -> StopList;
  ManyStops : String -> StopList -> StopList;
}
```

The same concrete syntax is used for clicks in all languages:

```
concrete ClickCnc of Click = QueryBaseCnc ** {
 lincat
  StopList = { s : Str };
 lin
  CStops xs = { s = "[" ++ xs.s ++ "]" };
  NoStop = { s = "" };
  OneStop x = { s = x.s };
  ManyStops x xs = { s = x.s ++ "," ++ xs.s };
}
```

Speech modality The `Query` module adds basic user queries and a way to use a click to indicate a place.

```
abstract Query = QueryBase ** {
 cat
  Place;
 fun
  GoFromTo : Place -> Place -> Input;
  GoToFrom : Place -> Place -> Input;
  PClick   : Click -> Place;
}
```

This module has a concrete syntax using English speech. Like terms in the `Query` category, `Place` terms are linearized to records with two fields, one for each modality.

```
concrete QueryEng of Query = QueryBaseCnc ** {
 lincat
  Place = {s1 : Str; s2 : Str};
 lin
  GoFromTo x y = {
   s1 = ["i want to go from"] ++ x.s1
        ++ "to" ++ y.s1;
   s2 = x.s2 ++ y.s2
  };
  GoToFrom x y = {
   s1 = ["i want to go to"] ++ x.s1
        ++ "from" ++ y.s1;
   s2 = x.s2 ++ y.s2
  };
  PClick c = { s1 = "here"; s2 = c.s };
}
```

Indexicality To refer to her current location, the user can use *"here"* without a click, or omit either origin or destination. The system is assumed to know where the user is located. Since *"here"* may be used with or without a click, inputs with two occurrences of *"here"* and only one click are ambiguous. A query might also be ambiguous even if it can be parsed unambiguously, since one click can correspond to multiple stops when the stops are close to each other on the map.

These are the abstract syntax declarations for this feature (in the `Query` module):

```
fun
 PHere    : Place;
 ComeFrom : Place -> Input;
 GoTo     : Place -> Input;
```

The English concrete syntax for this is added to the `QueryEng` module. Note that the click (`s2`) field of the linearization of an indexical *"here"* is empty, since there is no click.

```
lin
 PHere = { s1 = "here" ; s2 = [] };
 ComeFrom x = {
  s1 = ["i want to come from"] ++ x.s1;
  s2 = x.s2
 };
 GoTo x = {
  s1 = ["i want to go to"] ++ x.s1;
  s2 = x.s2
 };
```

Tying it all together The `TransportQuery` module ties together the transport network, speech modality and click modality modules.

```
abstract TransportQuery
       = Transport, Query, Click ** {
 fun
  PStop : Stop -> Place;
}
```

4.3 Multimodal output

The system's answers to the user's queries are presented with speech and drawings on the map. This is an example of parallel multimodality as the speech and the map drawings are independent. The information presented in the two modalities is however not identical, as the spoken output only contains information about where to change trams/buses. The map output shows the entire path, including intermediate stops.

Abstract syntax for routes The abstract syntax for answers (routes) contains the information needed by all the concrete syntaxes. All concrete syntaxes might not use all of the information. A route is a non-empty list of legs, and a leg consists of a line and a list of at least two stops.

```
abstract Route = Transport ** {
 cat
  Route;
  Leg;
  Line;
  Stops;
 fun
  Then : Leg -> Route -> Route;
  OneLeg : Leg -> Route;
  LineLeg : Line -> Stops -> Leg;
  NamedLine : String -> Line;
  ConsStop : Stop -> Stops -> Stops;
  TwoStops : Stop -> Stop -> Stops;
}
```

Concrete syntax for drawing routes The map drawing language contains sequences of labeled edges to be drawn on the map. The string *"drawEdge (6, [Chalmers, Vasaplatsen]); drawEdge (2, [Vasaplatsen, Gronsakstorget, Brunnsparken]);"* is an example of a string in the map drawing language described by the `RouteMap` concrete syntax.

The TransportLabels module extended by this module is a simple concrete syntax for stops.

```
concrete RouteMap of Route
       = TransportLabels ** {
 lincat
  Route, Leg, Line, Stops = { s : Str } ;
 lin
  Then l r = { s = l.s ++ ";" ++ r.s };
  OneLeg l = { s = l.s ++ ";" };
  LineLeg l ss =
   { s = "drawEdge" ++ "(" ++ l.s ++ ","
         ++ "[" ++ ss.s ++ "]" ++ ")" };
  NamedLine n = { s = n.s };
  ConsStop s ss = { s = s.s ++ "," ++ ss.s };
  TwoStops x y = { s = x.s ++ "," ++ y.s };
}
```

English concrete syntax for routes In the English concrete syntax we wish to list only the first and last stops of each leg of the route. The TransportNames module gives English representations of the stop names by replacing all non-English letters with the corresponding English ones in order to give the speech recognizer a fair chance.

```
concrete RouteEng of Route
       = TransportNames ** {
 lincat
  Route, Leg, Line = { s : Str } ;
  Stops = { start : Str; end : Str };
 lin
  Then l r = { s = l.s ++ "." ++ r.s } ;
  OneLeg l = { s = l.s ++ "." };
  LineLeg l ss =
   { s = "Take" ++ l.s ++ "from" ++ ss.start
         ++ "to" ++ ss.end };
  NamedLine n = { s = n.s };
  ConsStop s ss = { start = s.s;
                    end = ss.end };
  TwoStops s1 s2 = { start = s1.s;
                     end = s2.s };
}
```

5 Related Work

Johnston (1998) describes an approach to multimodal parsing where chart parsing is extended to multiple dimensions and unification is used to integrate information from different modalities. The approach described in this paper achieves a similar result by using records along with the existing unification mechanism for resolving discontinuous constituents. The main advantages of our approach are that it supports both parsing and generation, and that it does not require extending the existing formalism.

6 Conclusion

GF provides a solution to the problems named in the introduction to this paper. Abstract syntax can be used to characterise the linguistic functionality of a system in an abstract language and modality independent way. The system forces the programmer to define concrete syntaxes which completely cover the abstract syntax. In this way, the system forces the programmer to keep all the concrete syntaxes in sync. In addition, since GF is oriented towards creating grammars from other grammars, our philosophy is that it should not be necessary for a grammar writer to have to create by hand any equivalent grammars in different formats. For example, if the grammar for the speech recogniser is to be the same as that used for interaction with dialogue management but the grammars are needed in different formats, then there should be a compiler which takes the grammar from one format to the other. Thus, for example, we have a compiler which converts a GF grammar to Nuance's format for speech recognition grammars. The idea of generating context-free speech recognition grammars from grammars in a higher-level formalism has been described by Dowding et al. (2001), and implemented in the Regulus system (Rayner et al., 2003).

Another reason for using GF grammars has to do with the use of resource grammars and cascades of levels of representation as described in section 2. This allows for the hiding of grammatical detail from language and the precise implementation of modal interaction for other modalities. This enables the dialogue system developer to reuse previous grammar or modal interaction implementations without herself having to reprogram the details for each new dialogue system. Thus

the dialogue engineer need not be a grammar engineer or an expert in multimodal interfaces.

References

Björn Bringert. 2005. Embedded grammars. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden, February.

The Coq Development Team, 1999. *The Coq Proof Assistant Reference Manual*. Available at `http://pauillac.inria.fr/coq/`

Haskell B. Curry. 1963. Some logical aspects of grammatical structure. In Roman Jacobson, editor, *Structure of Language and its Mathematical Aspects: Proceedings of the 12th Symposium in Applied Mathematics*, pages 56–68. American Mathematical Society.

John Dowding, Beth Ann Hockey, Jean Mark Gawron, and Christopher Culy. 2001. Practical issues in compiling typed unification grammars for speech recognition. In *Meeting of the Association for Computational Linguistics*, pages 164–171.

R. Harper, F. Honsell, and G. Plotkin. 1993. A framework for defining logics. *Journal of the ACM*, 40(1):143–184.

Michael Johnston. 1998. Unification-based multimodal parsing. In *Proceedings of the 36th conference on Association for Computational Linguistics*, pages 624–630. Association for Computational Linguistics.

Lauri Karttunen, Jean-Pierre Chanod, Gregory Grefenstette, and Anne Schiller. 1996. Regular expressions for language engineering. *Natural Language Engineering*, 2(4):305–328.

Torbjörn Lager and Fredrik Kronlid. 2004. The Current platform: Building conversational agents in Oz. In *2nd International Mozart/Oz Conference*, October.

Lena Magnusson and Bengt Nordström. 1994. The ALF proof editor and its proof engine. In *Types for Proofs and Program*, volume 806 of *LNCS*, pages 213–237. Springer.

David L. Martin, Adam J. Cheyer, and Douglas B. Moran. 1999. The Open Agent Architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, 13(1–2):91–128, January–March.

Robin Milner, Mads Tofte, Robert Harper, and David MacQueen. 1997. *The Definition of Standard ML – Revised*. MIT Press, Cambridge, MA.

Mehryar Mohri. 1997. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–312.

Simon Peyton Jones. 2003. *Haskell 98 Language and Libraries*. Cambridge University Press, New York.

Aarne Ranta. 2004. Grammatical Framework, a type-theoretical grammar formalism. *Journal of Functional Programming*, 14(2):145–189.

A. Ranta. 2005. Modular Grammar Engineering in GF. *Research in Language and Computation*. To appear.

Manny Rayner, Beth Ann Hockey, and John Dowding. 2003. An open-source environment for compiling typed unification grammars into speech recognisers. In *EACL*, pages 223–226.

Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations

Kallirroi Georgila, Oliver Lemon, and James Henderson
Human Communication Research Centre
School of Informatics, University of Edinburgh
{kgeorgil,olemon,jhender6}@inf.ed.ac.uk

Abstract

We present and evaluate an automatic annotation system which builds "Information State Update" (ISU) representations of dialogue context for the COMMUNICATOR (2000 and 2001) corpora of human-machine dialogues (approx 2300 dialogues). The purposes of this annotation are to generate training data for reinforcement learning (RL) of dialogue policies, to generate data for building user simulations, and to evaluate different dialogue strategies against a baseline. The automatic annotation system uses the DIPPER dialogue manager. This produces annotations of user inputs and dialogue context representations. We present a detailed example, and then evaluate our annotations, with respect to the task completion metrics of the original corpus. The resulting data has been used to train user simulations and to learn successful dialogue strategies.

1 Introduction

We present and evaluate an automatic annotation system which builds "Information State Update" (ISU) representations of dialogue context (Larsson and Traum, 2000; Bos et al., 2003; Lemon and Gruenstein, 2004) for the COMMUNICATOR (2000 and 2001) corpora of human-machine dialogues (2331 dialogues) (Walker et al., 2001). The purpose of this annotation is to generate enough training data for a reinforcement learning (RL) approach to dialogue management, and also to be able to build user simulations, and to evaluate different dialogue strategies against a baseline. In general, for such an approach we require data that has either been generated and logged by ISU systems or that has been subsequently annotated (or a mixture of both).

A particular problem is that although the COMMUNICATOR corpus (recently released by the LDC) is the largest corpus of speech-act-annotated dialogues that we know of, it does not meet our requirements on corpus annotation for dialogue strategy learning and user simulation. For example, the user dialogue inputs were not annotated with speech act classifications, and no representation of dialogue context was annotated. We explain how we addressed such problems by building an automated annotation system which extends the COMMUNICATOR corpus, and we evaluate the resulting annotations. Note that prior work on ISU annotations (Poesio et al., 1999) was not automated, and was not suitable for large-scale annotations. We first

survey basic principles for annotating dialogue data with feature values for learning approaches. Section 2 describes the annotation system and section 3 presents our evaluation of the automatic annotations.

1.1 The DATE annotation scheme

The DATE (Dialogue Act Tagging for Evaluation) scheme (Walker et al., 2001) was developed for providing quantitative metrics for comparing and evaluating the 9 different DARPA COMMUNICATOR spoken dialogue systems. The scheme employs three orthogonal dimensions of utterance classification:

- *conversational domain*: about_task, about_communication, situation_frame
- *task-subtask*: top_level_trip (origin, destination, date, time, airline, trip_type, retrieval, itinerary), ground (hotel, car)
- *speech act*: request_info, present_info, offer, acknowledgement, status_report, explicit_confirm, implicit_confirm, instruction, apology, opening/closing.

The conversational domain dimension categorises each utterance as belonging to a particular "arena of conversational action". About_task refers to the domain task (in COMMUNICATOR this is air travel, hotel, and car-rental booking), and about_communication refers to conversational actions managing the communication channel (e.g. "are you still there?"). Situation_frame utterances manage the "culturally relevant framing expectations" in the dialogue (e.g. that the conversation will be in English, or that the system cannot issue airline tickets).

The task-subtask dimension relates to a model of the domain tasks that the dialogue system is designed to support. In COMMUNICATOR there were 2 main tasks: booking a flight ("top_level_trip"), and "ground" which was to determine whether the user also wanted to book a car rental and/or a hotel. The subtasks were elements such as finding the dates and times of the flights.

The speech_act dimension relates to the utterance's communicative goal. The speech acts used are relatively standard, and are described in detail in (Walker et al., 2001). Note that in the COMMUNICATOR data only the system's side of the dialogue is already annotated using the DATE scheme.

1.2 Annotation principles for ISU systems

The question arises of what types of information should ideally be logged or annotated for the purposes of building simulated users and optimising ISU dialogue systems via RL (Young, 2000). We can divide the types of information required into 5 main levels: *dialogue-level, task-level, low-level, history-level*, and *reward-level*. We also divide the logging and annotations required into information about *utterances*, and information about *states*. Utterances (by humans or systems) will have dialogue-level, task-level, and low-level features, while dialogue states will additionally contain some history-level information (see figure 2). Entire dialogues will be assigned reward features, e.g. taken from questionnaires filled by users.

A notable constraint on the information to be useful for machine learning is that all captured features should in principle be available to a dialogue system at runtime – so that a dialogue system using a learned policy can compute a next action in any state. This excludes, for example, word error rate from the state information usable for RL, since it can only be computed after transcription of user speech. In this case, for example, automatic speech recognition (ASR) confidence scores should be used instead. It also means that we need to annotate the ASR hypotheses of the systems, rather than the transcribed user utterances.

We now present an extended version of the

DATE scheme, producing sequences of dialogue information states that feed into learning algorithms and user simulations.

2 The automated annotation system

The annotation of the COMMUNICATOR data with information states was implemented using DIPPER (Bos et al., 2003) and OAA (Cheyer and Martin, 2001). Several OAA agents have been developed:

The first OAA agent *(readXMLfile)* is used for reading the original COMMUNICATOR corpus XML files, which contain information about dialogues, turns, utterances, transcriptions, and so on. When the agent reads information from an XML file, a corresponding DIPPER update rule fires and the dialogue information state is updated accordingly. Each information state corresponds to an utterance in the COMMUNICATOR data and a turn may contain several utterances.

A second OAA agent *(saveISsequence)* appends the current information state values to the file that will finally contain the whole sequence of information states (a DTD defining the format of IS-sequence files is available).

2.1 Confidence scoring

Ideally we would have dialogue data that contains ASR confidence scores. Unfortunately the COMMUNICATOR data does not have this information. However, the COMMUNICATOR data contains both the output of the speech recognition engine for a user utterance and a manual transcription of the same utterance carried out by a human annotator. We consider the word error rate (WER) to be strongly related to confidence scores and thus each time a user utterance is read from the XML file a third agent is called to estimate error rates (the *ComputeErrorRates* agent). Four different error rates are estimated: classical WER, WER-noins, SER, and KER.

WER-noins is WER without taking into account insertions. The distinction between WER and WER-noins is made because WER shows the overall recognition accuracy whereas WER-noins shows the percentage of words correctly recognised. The sentence error rate (SER) is computed on the whole sentence. All the above error estimations have been performed using the HResults tool of HTK (Young et al., 2002), which is called by *ComputeErrorRates*. Finally the keyword error rate (KER) is also computed by *ComputeErrorRates* (after the utterance has been parsed) and shows the percentage of the correctly recognised keywords (cities, dates, times, etc.). This is also a very important metric regarding the efficiency of the dialogues.

2.2 Interpreting user utterances, extending DATE

Even though all the above agents play a crucial rule in the annotation task, the most important subtask is to interpret the user's input and find its effect on the dialogue, or in other words to associate the user utterances with the correct speech acts and tasks. Multiple levels of parsing are thus required and are performed using Prolog clauses (part of the DIPPER *.resources* file).

Unfortunately, in the original COMMUNICATOR data XML files there is no distinction between the origin and destination cities in multiple-leg trips. That is, the tag "dest_city" could be used for any type of destination, regardless of whether the trip is single or multiple-leg. However, we believe that it is important to annotate these distinctions so that there is no overwriting of the values in filled slots such as "dest_city", "depart_date", etc. Moreover, the COMMUNICATOR data does not distinguish between departure and arrival dates or times, and sometimes it has times labelled as dates.

We use the following extended tasks and speech acts for annotating user utterances. These are in addition to the DATE scheme (Walker et al., 2001) used for the system

prompts annotation:

- Tasks which take values: continue_dest_city, depart_date, continue_depart_date, return_depart_date, arrive_date, continue_arrive_date, return_arrive_date, depart_time, continue_depart_time, return_depart_time, arrive_time, continue_arrive_time, return_arrive_time.

- Tasks which are either present or absent: no_continue_trip, return_trip, no_return_trip, accept_hotel_offer, reject_hotel_offer, accept_flight_summary, reject_flight_summary, accept_car_offer, reject_car_offer, accept_ground_offer, reject_ground_offer, accept_flight_offer, reject_flight_offer, hotel_city, car_interest, car_rental, rental_company, no_airline_preference, change_airline, flight_interest, send_itinerary, price_itinerary, id_number, number, continue, request_help, request_repetition, request_stop, bye, nonstop_flight, start_over.

- User speech acts: provide_info, reprovide_info, correct_info, reject_info, yes_answer, no_answer, question, command.

2.3 Computing grounding information

Part of our task is also to compute dialogue context information from the existing COMMUNICATOR annotations – for example, which utterances are grounded.

During processing the user's utterance the automatic annotation system will take into account the history of the dialogue and the labels on previous system utterances and then decide (via processing based on "patterns", see below) whether one or more slots should be filled, grounded, or even emptied if the slot is not confirmed. We define a piece of information as "grounded" (according to the system's perspective) only if it has been positively confirmed. Thus grounding processing can only take place after system utterances labelled as explicit_ or implicit_ confirmation. Here one could worry that our computation of grounding information is based on the assumption that the COMMUNICATOR systems had some notion of grounding in their algorithms, but that this kind of information is not included in the original COMMUNICATOR corpus. Nevertheless, the fact that we attempt to ground a slot only when the system attempts confirmation makes our assumption "safe". Moreover, only the speech recognition output is used for processing and deciding on the slots that will be filled or grounded. The human transcription of the user's input is only considered for computing error rates as explained above. This also ensures that we do not base our annotation on information that the systems did not have in runtime.

2.3.1 Example ISU annotation

We now examine an extract from the original COMMUNICATOR 2001 data (figure 1) and its new ISU annotation (figure 2). System utterances are marked with "S(n)" and user utterances as "U(n)" where n is the number of the utterance. For the system utterances the speech act and task pairs are given, for the user utterances only the speech recognition output is provided.

In utterance (U3) the user gives the departure date and time. However, the speech recognition output "october three first" was not considered by the system to be a valid date so the system understands only the time "late morning" and tries to confirm it in (S6). As we see in (S6) the speech act is "implicit_confirm" and the task is tagged as "depart_arrive_date" instead of "depart_arrive_time". Similar phenomena cause problems for correctly annotating the dialogues. In the above example, in (U3) our automatic annotation system will fill slot "de-

```
(S1) what city are you leaving from?
              (request_info, orig_city)
(U1) <CITY>hartford connecticut</CITY>
(S2) a flight from <CITY>hartford</CITY>
              (implicit_confirm, orig_city)
(S3) where would you like to go?
              (request_info, dest_city)
(U2) <CITY>orlando florida</CITY>
(S4) traveling to <CITY>orlando</CITY>
              (implicit_confirm, dest_city)
(S5) on what date would you like to travel?
              (request_info, depart_arrive_date)
(U3) <DATE_TIME>october three first
                          late morning</DATE_TIME>
(S6) traveling <DATE_TIME>late morning</DATE_TIME>
              (implicit_confirm, depart_arrive_date)
(S7) on what date would you like to travel?
              (request_info, depart_arrive_date)
(U4) <DATE_TIME>october thirty one</DATE_TIME>
```

Figure 1: Original COMMUNICATOR data.

part_time" with the value "late morning" and it will ground the "dest_city" slot. Then it reads the next system utterance (S6). Note that if it considers only the task label (depart_arrive_date) it will attempt to ground the wrong slot "depart_arrive_date", or in other words it will try to ground a slot that has not been filled yet. Therefore routines have been implemented so that the system can distinguish between valid dates or times.

In figure 2 we can see the automatically annotated Information State[1] corresponding to the dialogue context after U3 (the actual system output is in XML, but we do not show that here). Note especially the grounding of "dest_city" information in this move, and the history level of the annotation, which contains the sequences of speech acts and filled and grounded slots for the entire dialogue.

Note that full dialogues are also annotated with reward level features (e.g. actual task completion) from the PARADISE evaluations (Walker et al., 2000). These are used in reinforcement learning with the data.

In order to further explain the "patterns" we use to compute grounding, consider a variation on the above example. Imagine that in U3 the user does not give the departure date

```
DIALOGUE LEVEL
Turn: user
TurnStartTime: 988306674.170
TurnEndTime: 988306677.510
TurnNumber: 5
Speaker: user
UtteranceStartTime: 988306674.170
UtteranceEndTime: 988306677.510
UtteranceNumber: 5
ConvDomain: [about_task]
SpeechAct: [provide_info]
AsrInput: <date_time>october three first late
          morning</date_time>
TransInput: <date_time>october thirty first late
          morning</date_time>
System Output:

TASK LEVEL
Task: [depart_time]
FilledSlotValue: [late morning]
FilledSlot: [depart_time]
GroundedSlot: [dest_city]

LOW LEVEL
WordErrorRatenoins: 20.00
WordErrorRate: 20.00
SentenceErrorRate: 100.00
KeyWordErrorRate: 50.00

HISTORY LEVEL
SpeechActsHist: [yes_answer],opening_closing,[],
  opening_closing,instruction,request_info,
  [provide_info],implicit_confirm,request_info,
  [provide_info],implicit_confirm,request_info,
  [provide_info]
TasksHist: [null],meta_greeting_goodbye,[],
  meta_greeting_goodbye,meta_instruct,orig_city,
  [orig_city],orig_city,dest_city,[dest_city],
  dest_city,depart_arrive_date,[depart_time]
FilledSlotsHist: [null],[],[orig_city],[dest_city],
                 [depart_time]
FilledSlotsValuesHist: [yes],[],[hartford connecticut],
                  [orlando florida], [late morning]
GroundedSlotsHist: [],[],[],[orig_city],[dest_city]
```

Figure 2: Information State after U3.

but instead only replies to the confirmation prompt about the destination city (S4). There are 6 general ways the user could reply [2] : yes-class, e.g. "yes"; no-class, e.g. "no"; yes-class, city, e.g. "yes, orlando"; no-class, city, e.g. "no, boston"; no-class, city, city, e.g. "not orlando, boston"; city, e.g. "orlando".

In the first 5 cases it is easy for the annotation system to infer that there is positive or negative confirmation and thus ground the slot or not accordingly because of the appearance of "yes-class" or "no-class". However, in the last case the annotation system should compare the user's utterance with

[1] Items appearing between [brackets] are user inputs (sometimes not annotated) and other items are system actions.

[2] The "yes-class" corresponds to words or expressions like "yes", "okay", "right", "correct", etc. In the same way "no-class" stands for "no", "wrong", and so on.

the previous system's prompt for confirmation in order to decide whether the slot should be grounded or not. If the user says "orlando" he *re-provides* information and the slot "dest_city" is grounded whereas if he/she utters "boston" he/she corrects the system (correct_info), which means that the slot "dest_city" is not grounded and therefore its current value will be removed. In the "no-class, city, city" case the user rejects the value of the slot and corrects it at the same time. These are examples of the patterns used to compute grounding.

2.3.2 Confirmation strategies

When computing grounding it is important to take into account the different ways in which COMMUNICATOR systems ground information through various types of confirmation. In general all the COMMUNICATOR systems follow one of 3 general confirmation strategies. In the first strategy the system asks the user to fill a slot, then asks for confirmation (explicit or implicit), and moves to the next slot if the user confirms, or may keep asking for confirmation if the user does not cooperate. In the second strategy the system asks the user to fill several slots and then attempts to confirm them in one single turn. That means that the system's turn could consist of several utterances labelled as "explicit_confirm" or "implicit_confirm". A third strategy, which is a variation of the second strategy is when the system tries to confirm several slots in a single action, e.g. "explicit_confirm, trip", "implicit_confirm, orig_dest_city". Before confirmation the slots could be filled either in a single turn or in multiple turns.

For the first and third confirmation strategies it proves adequate to look only 1 or 2 steps backwards in the history of system utterances, whereas for the second strategy looking further back is required. We consider only the following speech acts: *request_info, explicit_confirm, implicit_confirm*, and *offer*. Other utterances (e.g. instructions) are not taken into account because they do not affect whether a slot will be filled or grounded.

Note that first the annotation system extracts the speech acts and possible tasks related to the current user utterance and then attempts to ground based on this information. Any kind of disambiguation required, e.g. to decide whether the speech act should be tagged as "provide_info" or "reprovide_info", is done before grounding. We deal with possible task ambiguity simultaneously with grounding e.g. if the user uttered a "city" name we cannot be sure whether it refers to an origin or destination city until we consider the context. The reason for this sequential procedure is that we want grounding to be computed exactly in the same way for both our annotation and simulation systems, so that we are able to straightforwardly compare our simulated dialogues with COMMUNICATOR data (Henderson et al., 2005). In simulation the only information we have is the list of tasks and speech acts for the user's input and not the ASR or real utterance transcription. For the first two confirmation strategies the annotation system should check whether the tasks extracted by parsing the user's utterance are included in the task labels of the previous explicit- or implicit- confirmation system prompts. The "explicit_confirm, trip" case adds further difficulty to grounding calculation because of the general task "trip". Thus the annotation system has to parse the system prompt to detect the slots that the system attempts to confirm. Then according to the type of speech act (reprovide_info, provide_info, correct_info, etc.) the system grounds one or more previously filled slots or fills one or more new ones.

3 Evaluating the automatic annotations

We evaluated our automatic annotation system by automatically comparing its output

with the actual (ATC) and perceived (PTC) task completion metrics as they are given in the COMMUNICATOR corpus. Our evaluation is restricted in the 2001 corpus because no such metrics are available for the 2000 data collection. If the final state of a dialogue – that is, the information about the filled and grounded slots – agrees with the ATC and PTC for the same dialogue, this indicates that the annotation is consistent with the task completion metrics. We consider only dialogues where the tasks have been completed successfully – in these dialogues we know that all slots have been correctly filled and grounded[3] and thus the evaluation process is simple to automate. This automatic method cannot give us exact results – it only indicates whether the dialogue is annotated more or less correctly.

We have applied our automatic evaluation method on the flight-booking portions of the automatically annotated COMMUNICATOR corpora. The results are that, for dialogues where ATC or PTC is marked as "1" or "2" (i.e. where the flight booking portion of the dialogue was successful or was considered to be successful), the current automatic annotations for the whole corpus showed 88.47% of the required slots to be filled ("filled slots accuracy") and 71.56% of the slots to be grounded ("grounded slots accuracy"). Detailed results are depicted in table 1.

The IBM system avoided confirmation and therefore we could not obtain results for the "grounded slots accuracy". In cases where the system attempts to confirm more than one slots in a single turn (second and third confirmation strategies), if the user gives a simple "no_answer" there is no way for the annotation system to detect the slot that the "no_answer" refers to. This can lead to fewer slots being grounded. One of the rules that the annotation system uses in grounding calculation is that only filled slots can be grounded, mostly to ensure that the system policies trained with the COMMUNICATOR annotated corpus (e.g. using RL) will be reasonable. This rule can cause problems in cases where for example the system knows the user's residence and therefore does not ask for the "orig_city" but in the sequel tries to confirm it, or when the user gives a negative confirmation to a filled slot value (thus the filled slot is emptied) but the system performs a second confirmation request with an alternative slot value. Now even if the user gives a "yes_answer" the slot will not be grounded because it is not filled anymore. The above observations explain the low scores of the Colorado and ATT systems (and to a lesser extent CMU) for "grounded slots accuracy".

System	Number of dialogues	Filled slots	Grounded slots
ATT	122	91.15	65.31
BBN	126	86.17	84.96
CMU	114	80.09	69.08
COL	152	84.83	55.40
IBM	165	94.51	-
LUC	127	93.44	78.90
MIT	159	89.19	74.42
SRI	85	87.08	78.28
ALL	1050	88.47	71.56

Table 1: ISU annotation accuracy for COMMUNICATOR 2001 data.

Our future work will focus on dealing with the above problems (e.g. by being more selective as to where some rules are applied). Moreover, we plan to perform manual evaluation of a portion of randomly selected annotated dialogues. Preliminary manual annotation has shown that not only the flight-booking portions of the data have been annotated with a high accuracy but also the hotel and car rental bookings.

As described in (Henderson et al., 2005), the first results of our supervised and rein-

[3]Error analysis showed that this assumption that the successful dialogues had all slots grounded (not just fi lled) is too strong.

forcement learning techniques trained with this data are promising, which also indicates that a significant number of dialogues have been annotated accurately.

4 Conclusion

We explained that the original COMMUNICATOR data (2000 & 2001) is not sufficient for our purposes (of learning dialogue strategies and user simulations from a corpus) since it does not contain speech-act annotations of user utterances or representations of dialogue contexts. We briefly reviewed the DATE annotation scheme, and our extensions to it. We then described an automatic annotation system which uses DIPPER. This annotates user inputs and dialogue "information state" context representations. We presented an example, discussed grounding and confirmation strategies, and evaluated our annotations with respect to the task completion metrics of the original corpus. This resulting data has been used to learn successful dialogue strategies (Henderson et al., 2005), and to train user simulations (Georgila et al., 2005).

Finally, we think that this automatic annotation system could be extended and altered for use in producing ISU annotations for other dialogue corpora – in particular for human-machine dialogue corpora where the semantics of the system output is already logged by the dialogue system itself.

Acknowledgements

This work is funded by the European Commission's 6th framework project "TALK: Talk and Look, Tools for Ambient Linguistic Knowledge" (IST 507802). We thank Johanna Moore for proposing work on this data.

References

Johan Bos, Ewan Klein, Oliver Lemon, and Tetsushi Oka. 2003. DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture. In *4th SIGdial Workshop on Discourse and Dialogue*, pages 115–124, Sapporo.

Adam Cheyer and David Martin. 2001. The Open Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1/2):143–148.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2005. Learning User Simulations for Information State Update Dialogue Systems. In *Eurospeech, (submitted)*.

James Henderson, Oliver Lemon, and Kallirroi Georgila. 2005. Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR data. In *IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems, (to appear)*.

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.

Oliver Lemon and Alexander Gruenstein. 2004. Multithreaded context for robust conversational interfaces: context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction (ACM TOCHI)*, 11(3):241– 267.

M. Poesio, R. Cooper, S. Larsson, C. Matheson, and D. Traum. 1999. Annotating conversations for information state update. In *Proceedings of Amstelogue'99 workshop on the semantics and pragmatics of dialogue*.

Marilyn A. Walker, Candace A. Kamm, and Diane J. Litman. 2000. Towards Developing General Models of Usability with PARADISE. *Natural Language Engineering*, 6(3).

Marilyn A. Walker, Rebecca J. Passonneau, and Julie E. Boland. 2001. Quantitative and Qualitative Evaluation of Darpa Communicator Spoken Dialogue Systems. In *Meeting of the Association for Computational Linguistics*, pages 515–522.

Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. 2002. *The HTK Book*. Cambridge University Engineering Department. (for HTK version 3.2).

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

Integration of Live Video in a System for Natural Language Dialog with a Robot

Erik Sandewall, Hannes Lindblom and Björn Husberg
Department of Computer Science
Linköping University
Linköping, Sweden
erisa@ida.liu.se, hanli513@student.liu.se, bjorn.husberg@home.se

Abstract

For the communication with mobile robots during their missions to locations that are inaccessible or dangerous for people, it is desirable to make use of a combination of natural language, preferably in spoken form, and a graphical presentation of what is in the robot's field of sight. The present article addresses architectural and methodological issues for such multimodal systems against the background of a system of this kind that we have developed, the WITAS Robot Dialog Environment (RDE).

1 Goals and Issues

A major reason for having mobile robots is to let them go to places where it is impossible, inconvenient, or dangerous for people to go. In this article we consider robots that are equipped with a video camera as a part of their perception system. For the communication with such robots during their missions, one would like to use a combination of spoken natural language and visual presentation of what is in the robot's field of sight. In order for the interaction to be as natural as possible, it should be possible to show the operator the actual video that is 'seen' by the robot. It should also be possible for the operator as well as the robot to refer to the moving video, both by phrases in the vocal communication and by gestures that indicate objects or areas in the passing video image. We have built a software system, the WITAS Robotic Dialog Environment (RDE) that provides major parts of these services for the English-language dialog with an unmanned helicopter (UAV). More specifically, the system provides two-way spoken dialog using entire phrases in restricted English, combined with display of live or previously recorded video and the possibility for the operator to point into the video.

In the course of designing this system we have identified a number of specific issues that will be important in any system of a similar kind. The most important issues are:

- Linguistic expressions and gestures that are used for referring to points, areas, trajectories and moving objects in the video.
- Synchronization between actual time, time referred to in the spoken interaction, and time of recording of presently displayed video during playback.
- Markup of video frames, allowing the dialog system to relate positions on a video frame to positions in the physical or simulated world of the robot's environment.
- Linguistic expressions that refer to the passing time in which the actual dialog takes place, including the impact of the time that is defined by the playing video, on the conduct of dialog.

Besides these specific issues, the overriding issue of system architecture is important and non-trivial. A system of this kind is by definition very

heterogeneous and requires the combined operation of different subsystems of very different character; in terms of design it is much more than the sum of its parts.

Therefore, although robot dialog is indeed an interesting topic for research on dialog systems, it can not be treated merely as an extension of other kinds of dialog. The purpose of the present article is to identify and discuss additional, sometimes extralinguistic aspects that must also be taken into account. We address general architectural issues for such multimodal systems, as well as the first three of the four specific problems mentioned above. The article is written from the background of our actual RDE system and the experience from developing it. The fourth issue above is also very important but will not be addressed in the present article.

2 The Robotic Dialog Environment

2.1 Outline of Architecture

The WITAS RDE software system (Robotic Dialog Environment) consists of three subsystems that in turn have several parts:

- An *Autonomous Operator's Assistant*, AOA, consisting of two parts: a *Speech and Graphics User Interface*, SGUI, and a *Dialog Manager* in a broad sense of the word.
- A *Robotic Agent* consisting of a *Robotic World*[1] that can be either the actual UAV system and the world it is flying in[2], or a simulator for this, and a *Video Server* that provides the channel from the UAV's on-board video camera to the AOA.
- A *Development Infrastructure* that provides the services that are needed for the development, demonstration, and validation of the dialog system.

An earlier version of the Dialog Manager was described in (Sandewall et al., 2003). The subsystems and parts of RDE communicate by message-passing and video-flow. In particular, the Dialog Manager and the SGUI communicate using KQML-like messages that in most cases contain a phrase in natural English, together with some protocol information. Messages from SGUI to dialog manager may also contain several alternative interpretations of a given phrase from spoken input.

The SGUI manages both an interface on the screen of the laptop or tablet that is used by the operator, and the audio input and output using a headset. At present we are using the Nuance[3] system for input and a choice of several alternatives, such as Festival[4], for the spoken output. In addition, the SGUI passively displays the video that is passed to it from the video server, while the video server in turn is actively controlled by the dialog manager. The SGUI also interprets the gestures that the operator makes on still images and on the moving video. Its interpretations of these gestures are passed to the dialog manager.

The dialog manager receives English phrases in textual form, and produces appropriate responses that are sent back to the SGUI to be pronounced. Apart from standard command and query behavior it contains managing multiple threads. Several versions of the dialog manager exist; please refer to (Eliasson, 2005) for recent work on this topic in our project. The dialog manager also receives messages representing the SGUI's interpretations of the user's gestures on the screen. Its interpretation of these gestures in combination with the language input results in two kinds of requests: helicopter operation requests that are sent to the Robotic World, and display requests that are sent to the video server, which in turn directs the requested video stream to the SGUI.

Both the robotic dialog situation per se and the integration with the video flow have a significant influence on the design of the dialog manager. Dialog with a robot results in multiple threads in the dialog, since both events in the robot's environment and policies requested by the operator may lead to initiatives from the robot's side in that dialog. The fact that the robot moves and acts in real time imposes real-time constraints on the dialog.

[1] We reserve the term 'environment' for the software system, and use the terms 'robot world' and its 'surroundings' for the place that the robot is in.

[2] See subsection 6.1 for additional details about the WITAS UAV system.

[3] http://www.nuance.com/

[4] http://www.cstr.ed.ac.uk/projects/festival/

Finally, the existence of a video stream concurrently with the dialog and the possibility of referring from language to video means that the dialog manager must be consistently time-aware.

Additional information about the WITAS RDE can be found via the WITAS website at
`http://www.ida.liu.se/ext/witas/`

2.2 The Real-Time of a UAV

Any robotic dialog system must take time into account, but this does not necessarily mean that everything must happen very fast. In fact, one of the observations when we started experimenting with UAV dialog scenarios was that often there is plenty of time. If it takes 20 seconds for the UAV to fly from point A to point B, then it may not matter so much whether a particular vocal response is made in two seconds or in three. In the end, there are some situations where very fast response by the dialog system is a high priority, and there are others where the system must instead 'pass the time' and indicate to its operator that it is still there while operator and dialog system are jointly waiting for the UAV to finish a particular task. The dialog system must be able to adapt to different real-time requirements and to switch gracefully to higher-priority communication tasks when needed.

2.3 Varieties of Video Input

In principle, the scene that the robot is facing can be presented either directly, using video obtained from a video camera that is mounted on the robot, or using virtual reality based on the combination of a world model and sensors mounted on the robot or in the environment. Our present system uses a composite video signal which is obtained, during actual flights, from the UAV's video camera. During simulations we use archived videos from earlier flights with our project's UAV. (Animated 'virtual reality' in closed-loop simulation is being implemented at present to serve as a development tool).

The real or synthesized video is recorded from a video camera that may sometimes be directed straight down during flights, but which may often be directed at an angle against the vertical. The coordinate transformation between a video frame and the map is therefore non-trivial and varies with time.

One particular facility in our system has turned out to be very important, namely the use of *playback*. Video that is received from the UAV is directed to the video server that is able to both forward it to the dialog system, and to accumulate it to its archive. Correspondingly, the dialog system is able to request both current video and playback from a particular time from the video server. This facility is important for several applications, but it has also been very helpful in the development work since it provides a natural way of integrating previously recorded video into simulation sessions.

3 Requirements and Methodology

3.1 System Aspects

The ultimate test of a system for dialog with a UAV is of course to carry out those dialogs during actual flights. However, this does not mean that its development can and should be performed through a large number of tests during actual flights; doing so would be very costly and inconvenient, in particular because of the safety arrangements that must surround test flights. It does not even mean that the validation and evaluation of the dialog system design should be done only through test flights. Many aspects of the system design are better verified in laboratory settings. In other words, the ability to conduct dialog during actual UAV flights is a necessary but not a sufficient requirement on the entire dialog system.

For both the development and the validation of the system it is useful to identify a few distinct and, as it turns out, fairly independent subtasks:

1. Solving the equipment problems that arise when computer-based dialog is to be performed at the airfield, working outdoors: taking into account the audio disturbances from the helicopter noise and the wind, as well as the difficulties of using a laptop or tablet in full daylight; handling wireless transmissions between the UAV itself, the UAV base station, and a nomadic operator; arranging for the operator to carry the necessary computer equipment in backpack style for easy walking, arranging for spectators to be able

to hear and see the multimedia dialog, and others more.

2. Implementing the software for interfacing the dialog system to the UAV control system, so that the dialog system can receive sensor information from the UAV on the appropriate level and also send commands to the UAV.

3. Implementing a simulator that interfaces to the dialog system in the same way and across the same interface as the actual UAV does.

4. Implementing the interactive video subsystem in such a way that it can be run both with video that arrives in the course of the session (closed loop) and with previously recorded video that has been archived on the video server.

5. Implementing a dialog system that is able to operate in a laboratory setting, using simulators, video servers, etc.

6. Integrating the above into a system that convincingly demonstrates well functioning dialog during actual flights.

Implementations of all these tasks were completed and integrated in time so that the dialog system could be demonstrated as a part of the main WITAS demo in October, 2003 in the presence of an international evaluation committee. However, this was done using an early version of the dialog system and a very early version of the video system that did not provide for gesture input. Considerable additional development has been done since the main demo but exclusively in a laboratory setting.

3.2 How Often Should we Fly?

The degree of interdependence or independence between the tasks mentioned above is an important question with respect to the development methodology. Concretely speaking: given that we have verified in late 2003 that our early dialog system worked together with the flying helicopter, and given that we have continued to develop and extend the system using a simulated robotic agent, how often do we need to test the dialog system in actual test flights in order to convince ourselves, and our colleagues, that the entire system is viable and that the proposed design is to be considered as valid?

The artificial intelligence community is traditionally skeptical towards simulations, and many ground-robot projects work with a tight testing loop. It is frequently argued that simulations do not (or can not) capture the full variability of what may happen in real-world situations, which suggests that test runs are needed continuously during the development work.

This argument does not automatically carry over to the case of a UAV robot, however. To begin with, every flight experiment is fairly complex and requires considerable preparation due to the complexity of the equipment and the obligatory safety measures, so that the overhead of working with very frequent tests would be forbidding. Secondly, the world of flying objects is very structured anyway. The possibility of "a lot of unexpected things happening" is just not permitted; civil aviation authorities would certainly not allow these devices for general use if that were the state of the current technology. The UAV per se must be strictly designed and strictly modelled and its conformity to the model must be validated stringently. Under these conditions it is natural that the development of a dialog system can largely proceed in the laboratory setting, using a simulation system that correctly models the possible and anticipated behaviors of a correct UAV, including its possible fault conditions. The verification that the dialog equipment is functional in the outdoor setting at the airfield must also be done, or course, but to a large extent it can be factored out as a separate issue.

3.3 Obtaining Information from Sensors for the Dialog

We have now argued that the ability of the dialog system to give commands to the UAV during actual flights does not need to be tested so often, and that most of the time it can be safely replaced by simulations, provided that consistency of interfaces and other elementary software discipline is applied. Unfortunately, the same does not apply for the information from sensors, and in particular for the interpretation of video input. In principle,

the dialog system should rely on the capability of the on-board UAV system to interpret the video signal in combination with other sensor data, providing it with the information about the Robotic World that it needs for the dialog.

In practice, however, the ability of the on-board WITAS system to provide this information is fairly restricted. If the dialog is restricted to those topics that are possible with the actual sensor-provided information, then it will be quite limited. Conversely, it has been easy to extend the dialog system so that the dialog can also cover many topics for which the required sensor information is just not available.

This situation can be met in a number of different ways. One possible reaction may be to postpone the work on the dialog system for a number of years, until the sensor information has become available. This is the only possibility if one insists that only those results that have been demonstrated in actual flights are of interest. However, applying the principle of concurrent engineering, it is arguably better to proceed with the development of the dialog technology while using prerecorded video (realistic, but sacrificing the closed control loop) with manual interpretation, as well as a simulator for the Robotic World (closed control loop, but virtual reality instead of real video) as substitutes for testing during actual flights. It is not too difficult after all to define a plausible interface for the connection between the dialog system and the forthcoming video interpretation system, and both sides may of course participate in specifying the interface between them.

4 World and Video

4.1 Ontology of the UAV Domain

The surroundings of our UAV is defined to be road traffic phenomena on the ground. Other aerial vehicles besides the UAV itself are not considered, and the ontology for ground phenomena is based on roads, vehicles that move along those roads, road crossings, buildings, persons, and a few other major types. The vehicle and building types are subdivided into subtypes, and they may have named parts such as the roof of a building. Objects of these types as well as their parts may be characterized with elementary properties, such as color and building material. There are the obvious actions for the UAV: take off, land, fly to point X (described as e.g. a building, a street intersection, or a person), follow vehicle A, fly along road R, and so on. Similarly for the observed ground vehicles there are actions such as ”arrive at point X”, ”overtake vehicle B”, and so forth.

This ontology and repertoire of phrases was first developed for vocal-only communication. As graphic interaction was added, we decided to begin with the following four types of gestures:

- Indicate a particular point in the image, for example for a fly-to command
- Indicate a particular area in the image, for example for a command to survey the area or to not fly over it
- Indicate a particular trajectory in the image, for example a segment of a road that the UAV is to fly along, or patrol back and forth
- Indicate a particular vehicle or other moving object that is part of a query or command to the UAV, for example that the UAV should catch up with it.

There are more usages of these gestures than one may notice at first. For example, the trajectory gesture is also useful for specifying the likely current or past position of a particular ground vehicle that one wishes to designate.

Gesture input is made using the touch screen of a tablet, or using a mouse on a conventional screen[5]. The gesture part of the SGUI interprets the movements of the pen or the mouse, and attempts to classify the input according to these four cases. The gesture type and the position and size parameters characterizing it are sent to the Dialog Manager. The gesture input is sometimes ambiguous, however, and it is then necessary to combine it with the spoken input in order to make the correct analysis. In such cases the SGUI sends the list of the alternatives to the Dialog Manager and allows it to make the choice.

[5]We acquired touch-screen tablets for this purpose but found that for development purposes it was more convenient to work with a mouse and an ordinary screen.

4.2 Synchronization issues

Consider a particular time when the user indicates an item in the live video and utters an accompanying phrase. The time of speech and the time of the gesture are used to connect those two speech acts, and they are therefore recorded independently. Their contents and timestamps must be stored, since there are situations where the later dialog makes reference back to one or the other of them. Furthermore, if the video is in playback mode so that the interaction refers to an earlier time then additional timepoints are involved.

The gesture only specifies points and figures in the coordinate system of frames in the video; it must be translated into the corresponding coordinates in the physical world, from which one can also derive what object is being referenced, for example a building, or a vehicle. It is therefore important that the time of display of a particular frame can be related to the exact *time of recording* of that frame. For this purpose, our video server puts a timestamp into every frame that arrives to it from the video source[6]. This timestamp is in the video frame itself, so that timestamped video can be archived and forwarded using standard video formats, and it is insensitive to the video encoding methods. When the SGUI interprets the input gestures of the user, it identifies both the gesture itself and the timestamps of the successive frames where the gesture was made.

4.3 Markup of Video Frames

Besides accounting for time, the system must also account for the coordinate transformation between the surfaces of the successive video frames on one hand and the Robotic World on the other. During actual UAV flights this information must come from the video interpretation and other sensor data interpretation that is done in the UAV itself. For archived videos from previous UAV flights it is possible to add it more or less manually, and for simulations with adjoining visualization these parameters can be generated as a by-product of the visualizer.

[6]Other approaches have been studied, in particular using the Microsoft "media time", but they were found not to give enough accuracy.

To be precise, there are two tasks that our system expects the data analysis in the UAV to perform: relating each frame to the proper coordinates in the physical world, and identifying the positions of moving objects, such as road vehicles, in the successive video frames. In fact, all objects in the world that one may wish to refer to in the user-system interactions except stationary objects that are known to the system's geographical model, must be recognized and reported by the data analysis system.

The information about each video frame constitutes a *markup* for that frame. In on-line mode the markup will be generated continuously by the data analysis system or the simulator; in playback mode it is possible to compute and archive the markup beforehand so that it is available when needed. In our case we worked almost exclusively with playback and archived markup, except for one occasion where a demonstration of on-line use was made. During that demo we used persons as "image analyzers" that produced the markup in real time by looking at the video and tracking reference points on their screens.

The markup sequence is parallel with the video frame sequence in the sense that each frame has its own markup. However, it is not necessary to send a continuous flow of markup information from the video server to the SGUI, since most of the time it would not be used. Instead, when the SGUI receives a gesture into the video being displayed, it identifies the timestamps of the frames being pointed into, and requests the accompanying markup of those frames from the video server.

4.4 Preliminary Gesture Interpretation

The structure of the message flow and the responsibilities of the respective software modules should now be clear. One specific practical point deserves to be mentioned, concerning the disambiguation of the input gestures. Each gesture is assumed to be in one of the four types mentioned above, and the exact choice of gesture type is sometimes dependent on the accompanying phrase. For example, a gesture showing three-quarters of a circle may either designate an area or a trajectory. In principle, it should therefore be necessary to first send incoming phrases through

language processing in the dialog manager before the gesture can be interpreted.

On the other hand, some interactive situations may require very rapid response. We have therefore adopted the following shortcut. The main part of the SGUI anyway receives input sentences in written form from the speech analyzer. The parsing of these sentences takes place in the dialog manager. However, in many cases the accompanying sentences are very simple, such as ”fly here”. Therefore, the SGUI is equipped with a list of standard phrases that it recognizes immediately by itself, and if a gesture is accompanied by such a standard phrase then its type can be decided at once.

Furthermore, the SGUI is defined to make such speech-gesture combination even in cases where the interpretation of the speech remains uncertain. It then sends the available information about the speech input, its assumptions about that speech input, and its resulting interpretation of the gesture to the dialog manager. If the latter should decide that the SGUI's interpretation of the speech was incorrect then it sends a message back to the SGUI asking it for a new interpretation of the gesture based on the alternative classification.

It might be argued that this solution is an artifact of a too strong separation between the SGUI part and the Dialog Manager part of the system, and that a more ”agent-oriented” architecture with many processes that send messages back and forth would have been a better way of handling the problem. We do not share that opinion: the present implementation does not have any particular disadvantage, and the separation of SGUI and Dialog Management as two distinct blocks has a performance advantage since it allows the SGUI to be implemented with strict consideration of real-time constraints while the Dialog Manager can give priority to the symbolic side of the computation.

5 Actual-Time Considerations

One of the interesting issues for this system is that several aspects of time must be taken into account in an effective way. We have already mentioned the connection between time of display and time of recording of the video, which is administrated by the SGUI using the timestamps. In addition, there are a number of computational and transmission delays that must be properly accounted for. The time where a spoken input phrase is concluded is not necessarily the same as the time when the pointing gesture is concluded. The times when those two speech acts have been interpreted in their respective computations need not coincide either. These aspects have been taken into consideration throughout our system, for example by keeping track of exact time of speech.

At present our system does not combine speech and gestures for output to the user, but only for input from the user. Output is speech only, or text only if the speech facility is disabled. When we proceed to two-way speech and gesture combinations it will be even more important to have full control of actual time, and for the system to choose its speech acts within the limitations of available time.

6 Related Work

6.1 The WITAS Projects

WITAS, the Wallenberg Laboratory for Information Technology and Autonomous Systems, is engaged in goal-directed basic research in the area of intelligent autonomous vehicles and other autonomous systems. Its main project focuses on the development of an airborne computer system that is able to make rational decisions about the continued operation of the aircraft, based on various sources of knowledge including pre-stored geographical knowledge, knowledge obtained from vision sensors, and knowledge communicated to it by data link.

The major part of the project addresses the UAV Technologies and is described e.g. in (Doherty et al., 2000; Doherty, 2004). The other part of the project concerns robotic dialog, in particular between a human operator and a UAV. Dialog activities in WITAS were organized as a project at Stanford during 2000-2002 and as a new project in Linköping since 2002. The work reported here is from the WITAS-Linköping Dialog Project.

6.2 Other Dialog Systems

The present article has addressed multimodal dialog with a robot using spoken language and

live video. Many earlier projects have addressed robotic dialog without the graphic modality or with still-image graphics without the live video aspect.

The KANTRA system by Lueth, Laengle, et al (Lueth et al., 1994) was a relatively early system providing natural-language communication for commanding a mobile ground robot. The report does not mention any use of graphics in this system.

Multimodal dialog systems that combine spoken language with still images include in particular the SmartKom (Reithinger et al., 2003; Herzog et al.,) and MATCH (Johnston et al., 2002) systems. These systems do not address robotic dialog since their task is to provide information for a mobile operator. The WITAS-Stanford dialog system of Lemon, Peters, et al (Lemon et al., 2002) is still one of the few published examples of a multimodal robotic dialog system, but its graphic capability is limited to specifying a point in a fixed, maplike aerial photograph. It therefore does not consider the problems of dealing with moving video and the resulting real-time and other issues.

Acknowledgements

The Video Server part of the RDE was designed and implemented by Björn Husberg, replacing an earlier video-archive system with fewer facilities. The extended SGUI that provides support for the Video Server, allows the user to point into the live video, and performs the necessary coordinate transformations etc. was designed and implemented by Hannes Lindblom as a M.Sc. thesis project. Erik Sandewall directs the WITAS Dialog Project including the work described here.

The entire WITAS RDE is the result of joint work with major contributions by Malin Alsén, Peter Andersson, Karolina Eliasson, Susanna Monemar and Tobias Nurmiranta as well as additional M.Sc. students, besides the present authors.

The support of the Wallenberg Foundation for the research reported here is gratefully acknowledged.

References

Patrick Doherty, Gosta Granlund, Kris Kuchcinski, Erik Sandewall, Klas Nordberg, Erik Skarman, and Johan Wiklund. 2000. The witas unmanned aerial vehicle project. In *Proc. 14th European Conference on Artificial Intelligence*, pages 747–755.

Patrick Doherty. 2004. Advanced research with autonomous unmanned aerial vehicles. In *Proc. 9th International Conference on Knowledge Representation and Reasoning*.

Karolina Eliasson. 2005. Integrating a discourse model with a learning case-based reasoning system. In *Proceedings of DIALOR-05*.

G. Herzog, H. Kirchmann, S. Merten, A. Ndiaye, and P. Poller. MULTIPLATFORM testbed: An integration platform for multimodal dialog systems. In H. Cunningham and J. Patrick, editors, *Proceedings of the HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS)*.

Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marily Walker, Steve Whittaker, and Preetam Maloor. 2002. MATCH: An architecture for multimodal dialog systems. In *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, pages 376–383.

Oliver Lemon, Alexander Gruenstein, and Stanley Peters. 2002. Collaborative activities and multitasking in dialogue systems. *Traitement Automatique des Langues (TAL)*, 43(2):131 – 154. Special Issue on Dialogue.

T.C. Lueth, Th. Laengle, G. Herzog, E. Stopp, and U. Rembold. 1994. KANTRA human-machine interaction for intelligent robots using natural language. In *Proceedings of the 3rd IEEE International Workshop on Robot and Human Communication, RO-MAN'94*, pages 106–111.

Norbert Reithinger, Gerd Herzog, and Alassane Ndiaye. 2003. Situated multimodal interaction in SmartKom. *Computers and Graphics*, 27(6):899–903.

Erik Sandewall, Patrick Doherty, Oliver Lemon, and Stanley Peters. 2003. Real-time dialogues with the WITAS unmanned aerial vehicle. In Andreas Günter, editor, *Annual German Conference on AI*, pages 52–63.

The Discourse Function of Final Rises in French dialogues

Marie Šafářová
ILLC
University of Amsterdam
m.safarova@uva.nl

Philippe Muller
IRIT
Université Paul Sabatier
muller@irit.fr

Laurent Prévot
Laboratory for Applied Ontology
ISTC-CNR, Trento
prevot@loa-cnr.it

Abstract

We report the results of an empirical study which aims to describe the discourse function of rises at right edge intonation boundaries in French. A Map-Task corpus containing two dialogues was annotated for IP boundaries and pitch transition points with the INTSINT intonational alphabet. The transcripts of the dialogues were labeled for dialogue structure and dialogue acts, using form and function tags. The relation between rises at IP boundaries with types of dialogue acts and topic shifts was statistically evaluated. As expected, the results show a positive correlation between rises and polar questions and between rises and discourse topic openings. Interestingly, the second correlation was stronger than the first, suggesting that the association of rises with topic openings is not simply due to the effect of questions as introducing new topics.

1 Introduction

In this paper, we report the results of an empirical study which aims to describe the discourse meaning of rises at right edge intonation boundaries in French dialogues. According to most French speakers, it is possible to turn an assertion into a question in French solely by pronouncing it with a rising intonation. While existing empirical studies (Grundstrom, 1973; Fónagy and Bérard, 1973) confirm that there is some correlation between rising and falling contours, and questions and assertions, respectively, they also show that rising intonation does not always go hand in hand with question intonation. Leaving aside the problem of question identification, one can thus legitimately raise the issue: What is the meaning of final rises in French? Clearly, an answer to this question cannot be given without a proper empirical study of the use of rises in natural speech but a corpus study of this sort is currently lacking for French (for other languages, see (Kowtko, 1996), for Glasgow English, or (Fletcher et al., 2002) for Australian English). An initial study has been performed on Post's Map Task corpus (Post, 2000) with two speakers and two dialogues (for a total of 301 speech turns); its goal was to resolve a number of annotation issues, such as the definition of intonation phrase boundary, its automatic assignment, the reliability of the employed algorithm and alphabet for intonation transcription, as well as the contribution of different kinds of dialogue acts and discourse structure taxonomies. The tested methodology is used

for the study of the Caelen corpus (Bessac and Caelen-Haumont, 1995), which is currently in progress.

In the following sections of this paper, we first describe in detail the basic theoretical issues pertaining to the methodology of analyzing the discourse function of intonation. In section 2., we focus on the definition of rises and on assignment of intonational boundaries. In section 3., we discuss the annotation of dialogue acts and dialogue structure. In the final section, we present the results of Post's Map Task corpus study.

2 Annotating Intonation: rises and intonation boundaries

2.1 Definition of a rise

As noted by (Post, 2000), there exists no consensus in French intonation studies about which changes in contours are categorical and whether one should take contours as holistic units or as a composition of individual tones, anchored on stressed syllables and intonation unit boundaries. Also with respect to rising intonation, a number of proposals can be found in the literature (Grundstrom, 1973; Post, 2000; Gunlogson, 2001). Leaving aside the option of direct perceptual distinctions, which may be highly unreliable, one can either opt for a phonetic description – with direct reference to the F0 contour, or a phonological one, which presupposes the adoption of an intonational grammar (necessarily a theoretical construct). A phonetic description (e.g., using points of maximum and minimum pitch) has the advantage that its result is annotator-independent; it can be done automatically and its quality depends solely on the algorithm used to calculate F0. The disadvantage is that there may be no linguistic reality corresponding to the phonetic information and generalizing over phonetic parameters may be difficult in a larger corpus study (with (semi-)free speech and many speakers). A phonological description (like ToBI) is/should be by definition linguistically relevant but it is time-consuming, costly (several professional annotators have to be employed), not quite reliable and with respect to intonation meaning probably both too fine-grained and not powerful enough.

In our present study, we have made use of the INTSINT annotation system which, in our view, circumvents some of the difficulties mentioned above. INTSINT (INternational Transcription System for INTonation) is a language-independent intonation transcription system developed in Aix-en-Provence (Hirst and Christo, 1998). It is phonetic to the extent that it makes use of automatically calculated macro-prosodic component of the F0 done by the accompanying MOMEL (MOdélisation de MELodie) algorithm; at the same time it is phonological in that it only labels certain target points on the MOMEL curve which are assumed to carry linguistic information. Basically, the MOMEL algorithm (Hirst and Espesser, 1993) provides an automatic stylization of the F0 contour, detected from the acoustic signal with the comb algorithm (Espesser, 1982) (see also (Louw and Barnard, 2004)). INTSINT covers both absolute prosodic events (**T** – Top; **M** – Mid; **B** – Bottom) and relative ones (**H** – Higher; **S** – Same; **L** – Lower; **U** – Upstep; **D**). The results still have to be checked manually but in general, the process is less time consuming than ToBI labeling.

2.2 Intonation Boundaries

When studying the role of rising intonation, it is not enough to focus on ends of utterances. All boundary tones associated with right edges of intonational phrases are assumed to be meaningful (Beyssade et al., 2004); moreover, important intonational events (e.g., encoding the difference between questions and non-questions) may not be aligned with right utterance edges. Intonational phrases in

French are optionally associated with acoustically and perceptually identifiable events of both rhytmical and tonal nature, such as pauses, drop in amplitude, final syllable lengthening, pitch resetting (on the first syllable of the subsequent phrase), and lack of some segmental assimilation processes (viz (Jun and Fougeron, 2002), (Post, 2000), (Féry, to appear), among others). Nevertheless, they also appear to be related to information structure articulation and (Beyssade et al., 2004) define them only as a reflection of the information structure of an utterance. It is also normally assumed that there is some correlation between prosodic phrasing and syntactic boundaries. Taking these observations into account, prior to the annotation process, the following rules were proposed to serve as a guidance to the annotators, together with their perceptual impression of the speech signal:

Intonation Boundary [Def.]

• Every completed turn boundary is a right edge IP boundary.

• Phonologically, an IP boundary is often (i) indicated by a pause, (ii) accompanied by syllable lengthening of the preceding syllable, (iii) followed by pitch resetting and (iv) accompanied by a drop in amplitude.

• An IP boundary often coincides with a major syntactic boundary (e.g., a finite clause boundary).

• An information structure constituent (topic, focus) can be followed by an IP boundary.

The three authors of the study (two native speakers of French, one non-native) served as annotators of the corpus. The results were evaluated for inter-annotator agreement using the *kappa*-statistics with the average κ for the three annotators being .718 ('good'). Evaluation of problematic examples showed that short phrases like *'oui'* (yes) were often a source of disagreement. Note that a general rule is impossible, since in some cases, *oui* is clearly parenthetical, identifiable by lower intensity than the rest of the unit, and should be treated as a separate IP, while other examples are more arguable. Short phrases such as *'bon'*, as in *'Bon, d'accord'*, particles and adverbial phrases like *'alors'*, *'donc'* or *'par contre'* and the utterance final *'quoi'* raised a similar problem. The annotators also disagreed at hesitation points (often filled with *'euh'*) and interruptions and self-corrections,[1] and at events which normally imply an intonational phrase boundary, such as pauses and vowel lengthening. Given that in case of disagreement, it was usually difficult to decide for or against a label, all the intonational phrase boundaries proposed by the three annotators were merged together in the final annotation.

Because the manual annotation of IP boundaries was judged to be rather time-consuming and thus unsuitable for the subsequent large corpus study, its results were compared to a semi-automatic method of boundary assignment, based on the automatic determination of pauses (with minimal length of 15 ms and maximal intensity of 40 dB) and a manual assignment of boundaries to all points of speaker switch. The semi-automatic method gave $\frac{2}{3}$ and $\frac{3}{4}$ of the manually assigned intonational phrase boundaries for the two dialogues, respectively;[2] only in a small number of cases did the pause not coincide to the original IP label, mostly due to long pre-plosive silences (some of them longer than 350ms). Also in view of the fact that some of the original manually assigned boundaries were quite likely just boundaries of smaller phrases (i.e., the accentual phrases), the result of the semi-automatic method was judged suitable to re-

[1] These have been found to be problematic also in MAE-ToBI (the American English standard for prosody labelling, viz (Beckman and Ayers, 1997)) for the same reason.

[2] The fact that the results were better for the second dialogue than for the first is probably due to the fact that the manual annotations were better in the second dialogue due to improved annotating skills of the labelers.

place the manual method in the Caelen corpus study.

3 Annotating dialogue acts and dialogue structure

3.1 Dialogue acts

As noted above, rising intonation is often assumed to be a marker of questions. One problem with testing this intuition empirically is that many utterances are ambiguous between questioning and asserting. (Grundstrom, 1973):26 lists the following cases as typically posing a problem to a clear question/non-question classification: (i) the speaker wants a simple confirmation from the addressee; (ii) the speaker is making a supposition which is only partially interrogative; (iii) the speaker is suggesting some word to the addressee to complete his utterance; (iv) the speaker pronounces only a part of his utterance which would have been a question if completed. The ambiguity between questions and assertions is one of the reasons why finding an objective procedure for identifying questions in a corpus is problematic. (Fónagy and Bérard, 1973) propose a simple solution by considering as questions all utterances that received a *oui/non* reply. This definition is too strong, however, given that many assertions receive an acknowledgment synonymous with the *oui*-reply. It is also too weak because some utterances which function as replies only contextually entail a *yes/no* response or express speaker's ignorance with respect to an issue. Questions are also often defined with reference to their intonation but for the purposes of the current study, it was necessary to identify them independently of their prosodic properties.

Assuming that the annotation of wh-questions is unproblematic, we made use of the following definition to identify polar questions (PQ), originally developed for English.[3] The definition takes into consideration segments larger than single utterances and/or turns.

Polar Questions [Def.] A polar question is an utterance that satisfies the following properties:

- it is turn-final
- it is followed by a reply from the addressee that contextually entails *yes/no/I don't know*
- if the utterance is of a declarative form, it can in the context be turned into a corresponding interrogative by inverting the subject and the finite verb, without resulting in an infelicitous discourse.

While the first two conditions are more or less straightforward, the third one actually relies on the intuitions of the annotator. Since in French, syntactic inversion is a rather obsolete way of forming questions, the inversion test can be replaced by a similar one using the *est-ce que* phrase. For instance, in example 1 (taken from Post's Map Task corpus), the declarative (G_{106}) satisfies the first two conditions of the definition above and can also be felicitously turned into an *est-ce que* question in its context. While the definition of questions proposed above may not identify *all* utterances intended as questions, it was designed to avoid cases of overgeneralization (though it is sometimes difficult to distinguish between the *assertion – acknowledgment* and *question – answer* sequences).

(1) (G_{103}) est-ce que tu as IP tu as le profond étang H IP[4]
[do you have the deep pond]
(F_{104}) oui, H sur la gauche IP
[yes, on the left]
(G_{105}) oui, tout à gauche. IP
[yes, all the way on the left]

[3] The procedure was tested for inter-rater agreement and against native speaker judgments, for results and discussion, see (Šafářová, in prep).

[4] Our examples will be presented with numbered utterances annotated with the speaker information (**G**iver and **F**ollower). **H** here stands for 'High' in the INTSINT intonational alphabet, **IP** stands for 'intonation phrase'.

(G_{106}) et tu as la grande plaine H IP
[and you have the big plain]
(F_{107}) non IP *[no]*

Apart from questions, the corpus was also annotated for other types of dialogue acts, partly based on an existing annotation scheme (Prévot, 2004) for route description dialogues. For the dialogue act annotations, the annotators had no access to the recordings and to the original punctuation signs in the transcript to avoid bias. Because of the difficulty of the dialogue act segmenting and labeling task, the final annotation was mainly based on a post-hoc discussion (rather than a majority decision). As in the case of intonational phrase boundary assignment, a number of problematic cases was identified, e.g.,
(i) it was sometimes difficult to determine if an utterance was an alternative bipolar question with an ellipsis of the second constituent, or if the utterance-final *ou* connective merely served to indicate speaker's uncertainty (as in "*et à beaucoup de centimètre du pic ou... ?*" - "and at many centimeters from the peak or...?"), especially if the question was responded to with a *'yes/no'* answer;
(ii) the *'est-ce que'*-test for questionhood sometimes gave unnatural renderings of the original declaratives, given that this form of questioning is rarely used in spoken French;
(iii) it was also difficult to decide whether a sequence of *'d'accord'* – *'d'accord'* represented a feedback elicitation (ALIGN in the MAPTASK schema (Kowtko, 1996)) and its answers or only two acknowledgments;
(iv) with respect to wh-questions, there is a potential difficulty with interrogative utterances with an ellipsis, as in some cases, the *wh* constituent may be missing, as in *"et alors l'hôtel par rapport aux torrents et l'océan?"* - "and so the hotel with respect to the torrents and the ocean?";
(v) it appeared desirable to classify also utterances like *je ne sais pas si tu le vois* - "I don't know if you can see it" as questions. The presence/absence of a rise against the main act categories (acknowledge, instruct, inform, question, answer, with or without new landmark introduction) was statistically evaluated. We considered as instances of IP-final rises those intonational events that were aligned with the manually assigned right-edge IP boundaries and labeled as **T**, **H** or **U** in INTSINT.[5] There was no convincing correlation between the act labels and the presence of a rise, except for instruction (using a landmark) associated with a rise (χ^2, p=0.006), answers to a question with absence of a rise (p=0.001), and polar question (with new referent) with a rise (p=0.03).

3.2 Annotating Discourse Structure

The aim of the discourse structure annotation task is to test for a possible correlation between discourse opening/closing and rises. At this level of discourse organization, however, two organizational principles are competing: game and topic structures. Game initiations and topic openings are often realized through the same move. These moves are utterances whose discourse functions are primarily "forward-looking" rather than "backward-looking" in the DAMSL terminology (Core and Allen, 1997). Despite this vicinity new games do not necessarily bring new topics into discussion (e.g., simple checks or verification questions), nor do topic shifts always initiate a new game (e.g a long speech-turn introducing a complex discourse structure made of several discourse topics). One possible explanation for these discrepancies is the very purpose of the dialogue game account which is purely to describe dialogues. Topics, on the other hand, concern any kind of discourse and in particular monologue stretches which are not interesting for dialogue games. The clues for recognizing these structures are also very

[5] In theory, the upstepped (U) rise should be a rather small pitch movement upwards but in practice, the U-rises were often as big as the H- or T-rises, which is why we included them in the evaluation.

different: While topic structures may require a deep semantic understanding of the conversation, game structures might be determined more directly from move types and move type sequences.

The notions of dialogue games and discourse topics have been discussed at length in the past and many proposals already exist in the literature. With respect to dialogue game definition, we opted for the MAPTASK schema detailed in (Kowtko, 1996). In this framework, dialogue games are sequences of potential moves initiated by a particular move (instruct, check, queries, explain, align)[6]. Regarding the notion of discourse topic, we rely on the account of (Asher, 2004), who recalls discourse topics can be either explicit and introduced by a specific utterance, or implicit and inferred from discourse content. In practice, it is thus difficult to identify topic openings in a systematic way. Finally, both game and topic structures admit sub-structures like embedded games and sub-topics.

The discourse structure was partly determined on the basis of the dialogue act annotation. The targets of each dialogue act were systematically identified (including "backward-looking" acts, such as acknowledgment or answer), and discourse relations (such as Elaboration, Background, Narration) were annotated. The resulting discourse structure provided a hierarchy of sub-dialogues, including cases of discourse popping (attachment of a new constituent higher in the hierarchy than the previous utterance).

Game and topic openings and closings should be derived without much difficulties from discourse structure. However, for succeeding in this task we need a rich discourse structure of our dialogues. The discourse relations involved in direction-giving dialogues have been studied in details in (Prévot, 2004) and we present a rough sum-up below.

[6]In DAMSL (Core and Allen, 1997) these functions are classified under the forward-looking function.

Successive *instructions* (e.g 2:$G_{125-126}$) are related by the coherence relation of *Narration* and constitute a topic.[7] Therefore, a sequence of instructions without landmark explanation constitutes only one discourse topic. Similarly in the dialogue game framework, an INSTRUCTING-game is possibly made of sequences of acknowledged instructions.

Landmark introductions (e.g 1:G_{103}) are related to *background* and are explicit new topics that can be elaborated with landmark descriptions. Similarly, in the MAPTASK they corresponds to the EXPLAINING-game which often appears embedded in the INSTRUCTING game.

We treat landmark descriptions and localizations (e.g 2:G_{130}) as *elaborations* of the constituent in which the landmark has been introduced. This could have been tackled in MAPTASK dialogue definitions by allowing the EXPLAINING-game to be recursive just like the INSTRUCTING ones.

Openings were identified with the following clues: **(i) discourse pop-ups, (ii) clarification and feedback requests**. Additional clues were provided by some discourse markers such as *donc* and *alors*.

The clues for *closings* relied more directly on the dialogue act annotation and included: **(i) double acknowledgments, (ii) acknowledgment following answers, (iii) answers to feedback request, (iv) specific discourse markers** such as *voilà* and *bon* (see (Prévot, 2004) for more details).

The examples 2 and 3 illustrate the opening/closing annotation. In the bracketed text are given some of the tags we used: SURFACE-FORM, FUNCTION, DISCOURSE-STRUCTURE, DISCOURSE-TOPICS. Surface forms included assertions (ASS), yes-no questions (QYN), wh-questions (QWH), alternative questions (QAL) and indeterminate

[7]We do not develop this point here but see (Asher, 2004) for more details on the nature of discourse relation and their consequences for discourse topic.

forms (IND). Functions included instructions (PAR, PSR), landmark introduction (IR), question-answer pair (QAP) and acknowledgment (ACK) . Discourse structure tags give information about discourse relations and targets. Finally discourse topic is added in case of an opening or a closing and discourse topics are numbered.

(2) (F_{124}) euh tu fais une boucle autour du deuxième petit pin *[err you do a loop around the second small pine tree]*
[ASS PAR NARR-119 OPEN-28]
(G_{125}) c'est à dire que tu passes par derrière *[so that means that you pass behind]*
[ASS PSR ELA-124]
(G_{126}) et tu reviens devant. *[and you come back in front]*
[ASS PSR NARR-125,ELA-124]
(F_{127}) mm *[mm]* [ASS ACK ACK-126]
(G_{128}) est-ce que tu as une colline *[do you have a hill]*
[QYN IR PELAQ-0 OPEN-29]
(F_{129}) non, j'ai pas de colline *[no I don't have a hill]*
[ASS QAP QAP-128]
(G_{130}) à côté du petit pin *[near the small pine tree]*
[IND DR ELAQ-128]
(F_{131}) j'ai rien à côté du petit pin *[I have nothing near the small pine tree]*
[ASS QAP QAP-130]

(3) (G_{103}) est-ce que tu as tu as le profond etang *[do you have the deep pond]*
[QYN IR PELAQ-0 OPEN-25]
(F_{104}) oui, sur la gauche *[yes on the left]*
[ASS QAP QAP-103]
(G_{105}) oui, tout à gauche *[yes completely on the left]*
[ASS ACK ACK-104 CLOSE-25]

In the resulting annotation, the number of openings was significantly higher than the number of closings (75 vs 52). It was sometimes difficult to identify closings by using the rules summarized above because some of them are implicit.

Rise was found to be correlated with the open/close distinction ($p<0.001$), rises being associated with openings and rise absences with closings. The corpus size was not sufficient to analyze the link between intonation and speaker roles, but there was no apparent bias due to specificities of the speakers. More work is needed to investigate the 'local roles' of speakers (associated with competence with respect to the current topic), which seems to be closely related to Kowtko dialogue game definitions (Kowtko, 1996).

It became clear that once the discourse structure is established, openings and closings are easier to determine and this can be done in a general way. However, building discourse structure was possible only with the input of a careful analysis of direction giving dialogues. Though we would like to abstract as much as possible from dialogue genre specificities, it did not appear to be feasible in practice. The reason is that dialogue game rules are usually defined for a particular dialogue game and discourse relation inference rules are established for a given discourse genre (e.g narrative, argumentative).

4 Conclusion and Future work

The results of the study of Post's Map Task corpus showed that with respect to dialogue acts, a positive correlation can be found between rises and (polar) questions, thus confirming earlier observations in the literature, and between rises and prescriptions using landmarks. On the other hand, answers to questions were more likely to appear without a rise. Mirroring similar results for English, we found that rises were significantly correlated to topic openings and rise absences with closings. The rise/openings correlation was stronger than the correlation rise/questions,

suggesting that the first association was not simply due to the question effect of introducing new discourse topics. Finally, speaker variation was observed, especially in the use of rises on acknowledgments which could, however, be due to their distinct dialogue roles (instruction giver vs. instruction follower), given that one of the dialogues was substantially shorter than the other.

Although the results of the Map Task corpus study are promising, they need to be tested on a corpus of a larger size and containing free conversations. A study of the Caelen corpus of tourist office dialogues is currently in progress. In order to describe the role of intonation in discourse in more detail, it may also turn out to be necessary to use a more fine-grained intonational transcription; alternatives to the MOMEL-based INTSINT alphabet are being investigated.

Acknowledgments

The authors would like to thank the Elisabeth Delais-Roussarie, Jean-Marie Marandin and Claire Beyssade for their comments on earlier versions of this manuscript, Brechtje Post for allowing us to use her data, and the DIALOR reviewers for their comments and suggestions.

References

N. Asher. 2004. Discourse topic. *Theoretical Linguistics*, (30):161–201.

M. Beckman and G. Ayers. 1997. Guidelines for ToBI labeling, version 3.0. Technical report, The Ohio State University.

M. Bessac and G. Caelen-Haumont. 1995. Analyses pragmatiques, prosodiques et lexicales d'un corpus de dialogue oral, homme-homme. In *Proceedings of the 3rd International Conference on Statistical Analysis of Textual Data*, pages 363–370, Rome.

C. Beyssade, E. Delais-Roussarie, J. Doetjes, J.-M. Marandin, and A. Rialland. 2004. Prosodic, syntactic and pragmatic aspects of information structure. an introduction. In F. Corblin and H. de Swart, editors, *Handbook of French Semantics*, pages 455–475. CSLI Publications.

M. Core and J. Allen. 1997. Coding dialogs with the DAMSL annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative actions in Humans and Machines*, pages 28–35, Cambridge, MA.

R. Espesser. 1982. Un système de détection du voisement et de f0. In *TIPA8*, pages 241–261.

C. Féry. to appear. Gradient prosodic correlates of phrasing in French. *Nouveaux départs en phonologie*.

J. Fletcher, R. Wales, L. Stirling, and I. Mushin. 2002. A dialogue act analysis of rises in Australian English Map Task dialogues. In *Proceedings of Speech and Prosody '02*, Aix-en-Provence.

I. Fónagy and E. Bérard. 1973. Questions totales simples et implicatives en Français Parisien. In A. Grundstrom and P. Léon, editors, *Interrogation et Intonation*, number 8, pages 53–98. Didier, Paris.

A. Grundstrom. 1973. L'intonation des questions en Français Standard. In A. Grundstrom and P. Léon, editors, *Interrogation et Intonation*, number 8, pages 19–51. Didier, Paris.

C. Gunlogson. 2001. *True to Form: Rising and Falling Declaratives as Questions in English*. Ph.D. thesis, UCSC.

D. Hirst and A. Di Christo, editors. 1998. *Intonation systems: a survey of twenty languages*. Cambridge University Press.

D. Hirst and R. Espesser. 1993. Automatic modelling of fundamental frequency using a quadratic spline function. *Travaux de l'Institut de Phonétique d'Aix*, 15:71–85.

S-A. Jun and C. Fougeron. 2002. The realizations of the accentual phrase in French intonation. *Probus*, 14:147–172.

J. Kowtko. 1996. *The function of intonation in task-oriented dialogues*. Ph.D. thesis, University of Edinburgh.

J.A. Louw and E. Barnard. 2004. Automatic intonation modeling with INTSINT. In *Proceedings of the Pattern Recognition Association of South Africa*, pages 107–111.

B. Post. 2000. *Tonal and Phrasal Structures in French Intonation*. Ph.D. thesis, University of Nijmegen.

L. Prévot. 2004. *Structure sémantique et pragmatique pour la modélisation de la cohérence dans des dialogues finalisés*. Ph.D. thesis, Université Paul Sabatier.

M. Šafářová. in prep. *Rises and Falls. Studies in the semantics and pragmatics of intonation*. Ph.D. thesis, University of Amsterdam.

Action at a distance: the difference between dialogue and multilogue

Jonathan Ginzburg and Raquel Fernández
Dept of Computer Science
King's College, London
The Strand, London WC2R 2LS
UK
{ginzburg,raquel}@dcs.kcl.ac.uk

Abstract

The paper considers how to scale up dialogue protocols to multilogue, settings with multiple conversationalists. We extract two benchmarks to evaluate scaled up protocols based on the long distance resolution possibilities of nonsentential utterances in dialogue and multilogue in the British National Corpus. In light of these benchmarks, we then consider three possible transformations to dialogue protocols, inspired by Goffman's audience taxonomy and formulated within an issue-based approach to dialogue management. We show that one such transformation yields protocols for querying and assertion that fulfill these benchmarks. We indicate how these protocols can be implemented in terms of conversational update rules.

1 Introduction

Dialogue—two person conversation—is by now a topic with an ever increasing theoretical, corpus-based, and implementational literature. In contrast, the study of *multilogue*—conversation with 3 or more participants—is still in its early stages. *The* fundamental issue in tackling multilogue is: to what extent do mechanisms motivated for dialogue (e.g. information states, protocols, conversational rules etc) scale up directly to multilogue?

There are of course various plausible views of the relation between dialogue and multilogue. One possible approach to take is to view multilogue as a sequence of dialogues. Something like this approach seems to be adopted in the literature on communication between autonomous software agents. However, even though many situations considered in multiagent systems do involve more than two agents, most interaction protocols are designed only for two participants at a time, perhaps in parallel. See e.g. the protocol specifications provided by FIPA (FIPA, 2003). Modelling of obligations and grounding becomes more complex when considering multilogue situations. The model of grounding implemented in the Mission Rehearsal Exercise (MRE) Project (Traum and Rickel, 2002), one of the largest multilogue systems developed hitherto, derives from the one designed by (Matheson, Poesio, and Traum, 2000) for dialogue and can only be used in cases where there is a single initiator and responder. It is not clear what the model should be for multiple addressees: should the contents be considered grounded when any of the addressees has acknowledged them? Should evidence

of understanding be required from every addressee?

Since their resolution is almost wholly reliant on context, non sentential utterances (NSUs) provide a large testbed concerning the structure of both dialogue and multilogue. In section 2 we present data from the British National Corpus (BNC) concerning the resolution of NSUs in dialogue and multilogue. The main focus of this data is with the distance between antecedent and fragment. We use this to extract certain benchmarks concerning multilogue interaction. In section 3 we sketch the basic principles of issue based dialogue management which we use as a basis for our subsequent investigations of multilogue interaction. This will include information states and formulation of protocols for querying and assertion in dialogue. In section 4 we consider three possible transformations on dialogue protocols into multilogue protocols. These transformations are entirely general in nature and could be applied to protocols stated in whatever specification language. We evaluate the protocols that are generated by these transformations with reference to the benchmarks extracted in section 2. Finally, in section 5 we discuss how these protocols can be implemented in terms of conversational update rules.

2 Long Distance Resolution of NSUs in Dialogue and Multilogue: some benchmarks

The work we present in this paper is based on data extracted from the British National Corpus (BNC). Our current corpus is a subportion of the BNC conversational transcripts consisting of 14,315 sentences. The corpus was created by randomly excerpting a 200-speaker-turn section from 54 BNC files. Of these files, 29 are transcripts of conversations between two dialogue participants, and 25 files are multilogue transcripts. A total of 1285 NSUs were found in our sub-corpus, 709 in dialogue and 576 in multilogue. All NSUs encountered within the corpus were classified according to the NSU typology presented in (Fernández and Ginzburg, 2002). Additionally, the distance from their antecedent was measured. Although the proportion of NSUs found in dialogue and multilogue is roughly the same, when taking into account the distance of NSUs from their antecedent, the proportion of long distance NSUs in multilogue increases radically: the longer the distance, the higher the proportion of NSUs that were found in multilogue. These differences are significant ($\chi^2 = 62.24$, $p \leq 0.001$). In fact, as Table 1 shows, NSUs that have a distance of 7 sentences or more appear exclusively in multilogue transcripts:

	1	2	3	4	5	≥6
Dia.	658	37	11	1	1	1
	(93%)	(5%)	(1.5%)	(1.5%)	(.1%)	(.1%)
Multi.	467	45	15	8	6	35
	(81%)	(8%)	(3%)	(1.5%)	(1%)	(6%)

Table 1: NSUs in dialogue and multilogue sorted by distance

Table 2 shows the distribution of NSU categories and their antecedent separation distance.[1] The classes of NSU which feature in our discussion below are boldfaced.

The last row in Table 2 shows the distribution of NSU-antecedent separation distances as percentages of the total of NSUs found. This allows us to see that about 87% of NSUs have a distance of 1 sentence (i.e. the antecedent was the immediately preceding sentence), and that the vast majority (about 96%) have a distance of 3 sentences or less.

[1]The distance we report is measured in terms of sentence numbers. It should however be noted that taking into account synchronous speech would not change the data reported in Table 2 in any significant way, as manual examination of all NSUs at more than distance 3 reveals that the transcription portion between antecedent and NSU does not contain any completely synchronous sentences in such cases.

NSU Class	Total	Distance 1	2	3	4	5	6	>6
Acknowledgment *Mm mm.*	595	578	15	2				
Short Answer *Ballet shoes.*	188	104	21	17	5	5	8	28
Affirmative Answer *Yes.*	109	104	4			1		
Clarification Ellipsis *John?*	92	76	13	2	1			
Repeated Ack. *His boss, right.*	86	81	2	3				
Rejection *No.*	50	49	1					
Factual Modifier *Brilliant!*	27	23	2	1	1			
Repeated Aff. Ans. *Very far, yes.*	26	25	1					
Helpful Rejection *No, my aunt.*	24	18	5		1			
Check Question *Okay?*	22	15	7					
Filler *... a cough.*	18	16	1		1			
Bare Mod. Phrase *On the desk.*	16	11	4			1		
Sluice *When?*	11	10	1					
Prop. Modifier *Probably.*	11	10	1					
Conjunction Phrase *Or a mirror.*	10	5	4	1				
Total	1285	1125	82	26	9	7	8	28
Percentage	100	87.6	6.3	2	0.6	0.5	0.6	2.1

Table 2: NSUs sorted by Class and Distance

The data in table 2 highlights two significant generalizations about multilogue: the first concerns short answers. With a few exceptions, NSUs that have a distance of 3 sentences or more are exclusively short answers. Not only is the long distance phenomenon almost exclusively restricted to short answers, but the frequency of long distance short answers stands in strong contrast to the other NSUs classes; indeed, over 44% of short answers have more than distance 1, and over 24% have distance 4 or more, like the last answer in the following example:

(1) Allan(1): How much do you think? Cynthia(2): Three hundred pounds. Sue(3): More. Cynthia(4): A thousand pounds. Allan(5): More. Unknown(6): <unclear> Allan(7): Eleven hundred quid apparently. [BNC, G4X]

It should be emphasized that long distance short answers is primarily a multilogue effect. Table 3 shows the total number of short answers found in dialogue and multilogue respectively, and the proportions sorted by distance over those totals. Note that only 18% of short answers found in dialogue have a distance of more than 1 sentence, with all of them having a distance of at most 3. This dialogue/multilogue asymmetry argues against reductive views of multilogue as sequential dialogue.

Short Answers	Total #	1	2	3	>3
Dialogue	54	82	9	9	0
Multilogue	134	44	11	8	37

Table 3: % over the totals found in dialogue and multilogue

The other striking generalization is the adjacency to their antecedent utterance of the remaining majoritarian classes of NSUs, Ack(nowledgements), Affirmative Answer, CE (clarification ellipsis), Repeated Ack(nowledgements), and Rejection. These are used either in grounding interaction, or to affirm/reject propositions.[2] The overwhelming adjacency to their antecedent underlines the locality of these interactions.

These data suggest two benchmarks protocols for multilogue need to satisfy:

(2) a. **Multilogue Long Distance short answers (MLDSA)**: querying protocols for multilogue must license short answers an unbounded number of turns from the original query.

b. **Multilogue adjacency of grounding/acceptance (MAG)**: assertion and grounding protocols for multilogue should license grounding/clarification/acceptance moves only adjacently to their antecedent utterance.

MLDSA and MAG have a somewhat different status: whereas MLDSA is a direct generalization from the data, MAG is a negative

[2] Acknowledgements and acceptances are, in principle, distinct acts: the former involves indication that an utterance has been understood, whereas the latter that an assertion is accepted. In practice, though, acknowledgements in the form of NSUs commonly simultaneously signal acceptance. Given this, corpus studies of NSUs (e.g. (Fernández and Ginzburg, 2002) often conflate the two.

constraint, posited given the paucity of positive instances. As such MAG is more open to doubt and we shall develop alternatives to it in the sequel.[3]

3 Dialogue Protocols and Conversational Rules

In this section we outline some of the basic principles of Issue-based Dialogue Management (Ginzburg (1996, forthcoming), Larsson, 2002) which we use as a basis for our subsequent investigations of multilogue interaction. Following (Larsson, 2002; Cooper, 2004), our dialogue theory is formulated in Type Theory with Records (TTR). This allows simple interfacing with the grammar, which is a Constraint-based Grammar closely modelled on HPSG but formulated in TTR, rather than using typed feature structures. See (Ginzburg, forthcoming) for details.

Within this approach, each dialogue participant's view of the common ground, the dialogue gameboard (DGB), are records of the type given in (3). We will frequently find it useful to talk directly of the first element of the Moves and QUD lists, referring to them respectively as **LatestMove** and **MaxQUD**.

(3)
$$\begin{bmatrix} \text{facts} : \text{Prop} \\ \text{Moves} : \text{list(IllocProp)} \\ \text{QUD} : \text{list(Question)} \end{bmatrix}$$

The querying/assertion protocols (in their most basic form) we assume for dialogue are summarized in Table 4.[4]

These protocols arise from the composition of *conversational (update) rules* akin to those introduced by (Larsson, 2002). A conversational rule is a mapping that specifies how one DGB configuration (the *preconditions*) can be modified into another (the *effects*). Two conversational rules *part1, part2* can be composed if they satisfy *preconds(part2)* $\sqsubseteq$ *effects(part1)*.

querying	assertion
LatestMove = Ask(A,q)	LatestMove = Assert(A,p)
A: push q onto QUD; release turn;	A: push p? onto QUD; release turn
B: push q onto QUD; take turn; make max-qud–specific; utterance[5] take turn.	B: push p? onto QUD; take turn; Option 1: Discuss p? Option 2: Accept p
	LatestMove = Accept(B,p)
	B: increment FACTS with p; pop p? from QUD;
	A: increment FACTS with p; pop p? from QUD;

Table 4: Protocols for querying and assertion

Specifically, the conversational rules that give rise to the protocols in Table 4 are the following TTR formulated rules from (Ginzburg, forthcoming), which for reasons of space are stated here informally in English:

(4) a. **QUD-Specificity (QSPEC)**: given MaxQUD = q, one can make an utterance which is q-specific.

b. **Ask QUD Update**: given LatestMove = Ask(A,B,q), q becomes QUD maximal

c. **Assert QUD Update**: given LatestMove = Assert(A,B,p), p? becomes QUD maximal

d. **Accept**: given LatestMove = Assert(A,B,p), B can make utterance such that LatestMove = Accept(B,A,p).

e. **UpdateFacts + DowndateQUD**: Given LatestMove = Accept(B,p),

[3] We would like to thank an anonymous reviewer for Dialor for strengthening our open mindedness regarding MAG.

[4] For reasons of space we do not formulate an explicit protocol for grounding here—the structure of such a protocol resembles the assertion protocol. Our subsequent discussion of assertion can be modified *mutatis mutandis* to grounding.

[5] An utterance whose content is either a proposition p About max-qud or a question q_1 on which max-qud Depends. For the latter see footnote 8. If one assumes QUD to be a stack, then 'max-qud–specific' will in this case reduce to 'q–specific'. But the more general formulation will be important below.

conjoin p with FACTS, downdate p? and all other qs from QUD resolved by FACTS

NSU Resolution We assume the account of NSU resolution developed in (Ginzburg and Sag, 2000). The essential idea they develop is that NSUs get their main predicates from context, specifically via unification with the question that is currently *under discussion*, an entity dubbed the *maximal question under discussion* (MAX-QUD). NSU resolution is, consequently, tied to conversational topic, viz. the MAX-QUD.[6]

Dialogue short answers The QUD–based resolution strategy affords the potential for non adjacent short answers in *dialogue*, given the assumption that QUD is a stack. These, as discussed in section 2, are relatively infrequent. Two commonly observed *dialogue* conditions will jointly enforce adjacency between short answers and their interrogative antecedents: (a) Questions have a simple, one phrase answer. (b) Questions can be answered immediately, without preparatory or subsequent discussion. For multilogue (or at least certain genres thereof), both these conditions are less likely to be maintained: different CPs can supply different answers, even assuming that relative to each CP there is a simple, one phrase answer. The more CPs there are in a conversation, the smaller their common ground and the more likely the need for clarificatory interaction. A pragmatic account of this type of the frequency of adjacency in dialogue short answers seems clearly preferable to any actual mechanism that would *rule out* long distance short answers. These can be perfectly felicitous—see e.g. example (1)above which would work fine if the turn uttered by Sue had been uttered by Allan instead.

[6]The resolution of NSUs, on the approach of (Ginzburg and Sag, 2000), involves one other parameter, an antecedent sub-utterance they dub the *salient-utterance* (SAL-UTT). This plays a role similar to the role played by the *parallel element* in higher order unification–based approaches to ellipsis resolution (see e.g. (Pulman, 1997)). For current purposes, we limit attention to the MAX-QUD as the nucleus of NSU resolution.

4 Scaling up Protocols

(Goffman, 1981) introduced the distinction between *ratified participants* and *overhearers* in a conversation. Within the former are located the speaker and participants whom she takes into account in her utterance design—the intended addressee(s) of a given utterance, as well as *side participants*. In this section we consider three possible principles of protocol extension, each of which can be viewed as adding roles for participants from one of Goffman's categories. The final principle we consider, Add Side Participants (ASP), seems to yield the best results, relative to the benchmarks we introduced in section 2. We state the principles informally as transformations on operational construals of the protocols and then in section 5 we indicate how such protocols could be implemented in terms of conversational update rules.

Add Overhearers (AOV) This involves adding participants who merely observe the interaction. They keep track of facts concerning a particular interaction, but their context is not facilitated for them to participate:

(5) Given a dialogue protocol π, add roles C_1,...,C_n where each C_i is a silent participant: given an utterance u_0 classified as being of type T_0, C_i updates C_i.DGB.FACTS with the proposition $u_0 : T_0$.

Applying AOV yields essentially multilogues which are sequences of dialogues. A special case of this are moderated multilogues, where all dialogues involve a designated individual (who is also responsible for turn assignment.). AOV will not allow for long distance short answers across more than two participants, as in e.g. (1), so will fail the MLDSA benchmark.

Duplicate Responders(DR)

(6) Given a dialogue protocol π, add roles C_1,...,C_n which duplicate the responder role

Applying DR to the querying protocol in Table 4 yields a protocol in which each responder to A's query q gets to provide their input concerning q (i.e. a q-specific utterance). This yields interactions such as (7) from the BNC:

(7) Anon (1) How about finance then? <pause> Unknown1 (2): Corruption. Unknown2(3): Risk <pause dur=30> Unknown3(4): Wage claims <pause dur=18>

Such a querying protocol licenses long distance short answers, so satisfies the MLDSA benchmark. On the other hand, the contextual updates it enforces will not enable it to deal with the following (constructed) variant on (7), in other words does not afford responders to comment on previous responders, as opposed to the original querier:

(8) A(1): Who should we invite for the conference? B(2): Svetlanov. C(3): No (=Not Svetlanov), Zhdanov. D(4): No (= Not Zhdanov, $\neq$ Not Svetlanov), Gergev

Applying DR to the assertion protocol will yield a protocol in which multiple responders get to sequentially accept an assertion. This will licence long distance acceptance and thus is inconsistent with the MAG benchmark.On the other hand, it is potentially useful for interactions where there is explicitly more than one direct addressee.

Add Side Participants (ASP) This is a principle intermediate between AOV and DR:

(9) Given a dialogue protocol π, add roles C_1,...,C_n, which affect the same contextual update as the interaction initiator.

In terms of the protocols introduced in section 3, ASP involves the same protocols modified such that (a) the audience is a non-singleton, (b) one member of this audience instantiates the addressee role and responds, the others update their DGBs in similar fashion to the original speaker.

This will yield a protocol for assertion that satisfies the MAG benchmark in that acceptance is strictly local. This is because it enforces *communal acceptance*—acceptance by one CP can count as acceptance by all other addressees of an assertion. There is an obvious rational motivation for this, given the difficulty of a CP constantly monitoring an entire audience (when this consists of more than one addressee) for acceptance signals—it is well known that the effect of visual access on turn taking is highly significant (Dabbs and Ruback, 1987). It also enforces quick reaction to an assertion—anyone wishing to accept p or dissent from p must get their reaction in early i.e. immediately following the assertion since further discussion of p? is not countenanced if acceptance takes place. The latter can happen of course as a consequence of a dissenter not being quick on their feet; on this protocol to accommodate such cases would require some type of backtracking.[7]

Applying ASP to the dialogue querying protocol yields a protocol that improves on the DR generated protocol because it does allow responders to comment on previous responders—the context is modified as in the dialogue protocol. Nonetheless, as it stands,

[7] In this respect an example pointed out by an anonymous Dialor reviewer is relevant; the reviewer suggests that 'that a disagreement by one respondent need not precede acknowledgement by another. E.g. I don't think there's anything wrong with this dialogue:

(i) A: We're inviting Svetlanov. B: Right. C: No we're not.

We agree that the dialogue is fine. However, intuitively, it seems to us, (and indeed on any protocol in which acceptance does not itself require acceptance,) that C's move will potentially give rise to some sort of backtracking, at least from B. See below though for a version of acceptance, *distributed acceptance* that can accommodate such cases.

this protocol won't fully deal with examples such as (7)—the issue introduced by each successive participant takes precedence given that QUD is assumed to be a stack. This can be remedied by slightly modifying this latter assumption: we will assume that when a question q gets added to QUD it doesn't subsume *all* existing questions in QUD, but rather only those on which q does not depend:[8]

(10) q is QUD$^{mod(dependence)}$ maximal iff for any q_0 in QUD such that $\neg$Depend(q, q_1): $q \succ q_0$.

This is conceptually attractive because it reinforces that the order in QUD has an intuitive semantic basis. The effect of this will be to ensure that any polar question $p?$ introduced into QUD, whether by an assertion or by a query, subsequent to a wh-question q on which $p?$ depends does not subsume q. Hence, q will remain accessible as an antecedent for NSUs, as long as no new unrelated topic has been introduced. Assuming this modification to QUD is implemented in the above ASP–generated protocols, both MLDSA and MAG benchmarks are fulfilled.

5 Conversational Rules for Multilogue

In this section we consider how the protocols scaled up according to the principles ASP and DR discussed in section 4 can be compositionally decomposed from conversational rules akin to those in (4).[9] QSPEC does not require any modification—once a question q is pushed on QUD, licensing a q–specific utterance is characteristic of both querying and assertion protocols.

[8] The notion of dependence we assume here is one common in work on questions, e.g. (Ginzburg and Sag, 2000), intuitively corresponding to the notion of 'is a subquestion of'. q_1 depends on q_2 iff any proposition p such that p resolves q_2 also satisfies p is about q_1.

[9] Adding overhearers (AOV) involves no substantive change to the previously discussed protocols: AOV is already in the form of an update rule, which concerns solely the overhearers.

Adding Side Participants (ASP) involves one rather minor modification to the rules in (4). The illocutionary propositions that constitute the values of LatestMove in the various rules now need to have a plural set of individuals as their type. For instance:

(11) **Ask QUD Update (plural audience)**: given LatestMove = Ask(A,$\{C_1, \ldots, C_n\}$,q), q becomes QUD maximal for $\{A, C_1, \ldots, C_n\}$

(12) **UpdateFacts + DowndateQUD (plural audience)**: Given LatestMove = Accept(B,$\{A, C_1, \ldots, C_n\}$,p), $\{B, A, C_1, \ldots, C_n\}$ conjoin p with FACTS, downdate p? and all other qs from QUD resolved by FACTS

Pluralized QUD update rules are also components of DRed querying and assertion rules. Given the modification to QUD proposed in the previous section, a reasonably direct treatment of DRed querying follows: following a query q by A, Ask QUD update enables the next speaker to provide a q-specific answer. By the ordering in QUD, q will remain maximal for any subsequent speaker who has not downdated it.

The main additional modification seems to concern acceptance. Consider first the preconditions for an acceptance move—the difference from the dialogue case is that they no longer involve adjacency of the assertion in question. They now involve the combination of the existence of a prior assertion of p and the maximality of $p?$ in QUD:

(13) **Distributed Accept**: given Moves = $\langle \ldots$Assert(A,$\{C_1, \ldots, C_n\}$,p) $\ldots\rangle$, and Max-QUD = p?, C_i can make utterance such that LatestMove = Accept(C$_i$,$\{A, , C_1, \ldots, C_n\}$,p).

It seems like fact-incrementation/QUD downdate needs to be divided into two subcases: one that concerns the addressees, the

other that concerns the original asserter. To take these in order: for the addressees, given the distributed nature of acceptance here, the precondition for fact-incrementation/QUD-downdate has to be an acceptance by that particular individual. For the original asserter the precondition for fact-incrementation/QUD-downdate is the existence of acceptance acts of p by all addressees:

(14) **Distributed UpdateFacts + DowndateQUD (audience version)**: Given LatestMove = Accept(C_i, $\{A, C_1, \ldots, C_n\}$,p), C_i conjoin p with FACTS, downdate p? and all other qs from QUD resolved by FACTS

(15) **Distributed UpdateFacts + DowndateQUD (asserter version)**: Given Moves = $\langle \ldots$ Assert(A,$\{C_1, \ldots, C_n\}$,p) ..., Accept(C_1,$\{A, \ldots, C_n\}$,p), ..., Accept(C_n,$\{A, , C_1, \ldots\}$,p) $\rangle$, A conjoin p with FACTS, downdate p? and all other qs from QUD resolved by FACTS

6 Conclusions and Future Work

In this paper we have considered how to scale up dialogue protocols to multilogue. We have extracted two benchmarks, MLDSA and MAG, to evaluate scaled up protocols based on the long distance resolution possibilities of NSUs in dialogue and multilogue. In light of these benchmarks, we consider three possible transformations to protocols, which can be intuited as adding roles that correspond to different categories of an audience originally suggested by Goffman. We then indicate how such protocols could be implemented in terms of conversational update rules.

In the future we intend to implement multilogue protocols in CLARIE so it can simulate multilogue. We will then evaluate its ability to process NSUs from the BNC.

Acknowledgements

We would like to thank three anonymous Dialor reviewers for some very useful comments and challenges, which reshaped our thinking on a number of significant issues. We would also like to thank Pat Healey, Shalom Lappin, Richard Power, and Matt Purver for discussion. Earlier versions of this work were presented at colloquia at ITRI, Brighton, and at the Université Paris, 7. The research described here is funded by grant number RES-000-23-0065 from the Economic and Social Research Council of the United Kingdom.

References

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

R. Cooper. 2004. A type theoretic approach to information state update in issue based dialogue management. Invited paper, *Catalog'04, the 8th Workshop on the Semantics and Pragmatics of Dialogue*, UPF, Barcelona.

J. Dabbs and R. B. Ruback. 1987 Dimensions of group process: amount and structure of vocal interaction. *Advances in Experimental Social Psychology* 20, pp. 123–169.

N. Fay, S. Garrod, and J. Carletta. 2000. Group discussion as interactive dialogue or serial monologue. *Psychological Science*, pp. 481–486.

R. Fernández and J. Ginzburg. 2002. Non-sentential utterances: A corpus study. *Traitement automatique des languages. Dialogue*, 43(2):13–42.

FIPA. 2003. The foundation for intelligent physical agents. Interaction protocol specifications. http://www.fipa.org.

J. Ginzburg and R. Cooper. 2001. Resolving ellipsis in clarification. In *Proc. of the 39th Meeting of the ACL*, Toulouse.

J. Ginzburg and I. Sag. 2000. *Interrogative Investigations: the form, meaning and use of English Interrogatives*. CSLI Lecture Notes, 123. CSLI Publications.

J. Ginzburg. (forthcoming). *Semantics and Interaction in Dialogue* CSLI Publications and U. of Chicago Press.

J. Ginzburg. 1996. Interrogatives: Questions, facts, and dialogue. In Shalom Lappin, editor, *Handbook of Contemporary Semantic Theory*. Blackwell, Oxford.

E. Goffman 1981 *Forms of Talk*. University of Pennsylvania Press, Philadelphia.

S. Larsson. 2002. *Issue based Dialogue Management*. Ph.D. thesis, Gothenburg University.

C. Matheson and M. Poesio and D. Traum. 2000. Modelling Grounding and Discourse Obligations Using Update Rules. *Proc. of NAACL 2000*, Seattle.

S. Pulman. 1997. Focus and higher order unification. *Linguistics and Philosophy*, 20.

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

D. Traum and J. Rickel. 2002. Embodied agents for multiparty dialogue in immersive virtual world. In *Proc.of AAMAS 2002*, pp. 766–773.

D. Traum. 2003. Semantics and pragmatics of questions and answers for dialogue agents. In *Proc. of the 5th IWCS*, pp. 380–394. University of Tilburg.

Empirical determination of thresholds for optimal dialogue act classification

Nick Webb, Mark Hepple and Yorick Wilks

Natural Language Processing Group
Department of Computer Science
University of Sheffield, UK
{n.webb,m.hepple,y.wilks}@dcs.shef.ac.uk

Abstract

We present recent experiments which build on our work in the area of Dialogue Act (DA) tagging. Identifying the dialogue acts of utterances is recognised as an important step towards understanding the content and nature of what speakers say. We describe a simple dialogue act classifier based on purely *intra-utterance* features — principally word n-gram cue phrases. Such a classifier performs surprisingly well, rivalling scores obtained using far more sophisticated language modelling techniques for the corpus we address. The approach requires the use of thresholds effecting the selection of n-gram cues, which have previously been manually supplied. We here describe a method of automatically determining these thresholds to optimise classifier performance.

1 Introduction

In the area of spoken language dialogue systems, the ability to assign user input with a functional tag which represents the communicative intentions behind each utterance — the utterance's *dialogue act* — is acknowledged to be a useful first step in dialogue processing. Such tagging can assist the semantic interpretation of user utterances, and can help an automated system in producing an appropriate response.Researchers, for example (Hirschberg and Litman, 1993; Grosz and Sidner, 1986), speak of cue phrases in utterances which can serve as useful indicators of dialogue acts.

In common with the work of (Samuel et al., 1999), we wanted to detect automatically word n-grams in a corpus that might serve as potentially useful cue phrases, potential indicators of dialogue acts. The method we chose for selecting such phrases is based on their *predictivity*. The predictivity of cue phrases can be exploited directly in a simple model of dialogue act classification that employs only intra-utterance features. The core of this paper investigates whether the crucial values for predicitivity of cue phrases can be determined empirically using a validation set of data, held out from evaluation. In a recent paper (Webb et al., 2005), we report early results of experiments evaluating our simple approach to

classification on the SWITCHBOARD corpus, using manually pre-set thresholds for our key variables. Surprisingly, the results we obtain rival the best results achieved on that corpus, in work by Stolcke *et al.* (Stolcke et al., 2000), who use a far more complex approach involving Hidden Markov modelling (HMM), that addresses both the sequencing of words *within* utterances and the sequencing of dialogue acts *over* utterances.

2 Related Work

There has been an increasing interest in using machine learning techniques on problems in spoken dialogue. One thread of this work has addressed dialogue act modelling, i.e. the task of assigning an appropriate dialogue act tag to each utterance in a dialogue. It is only recently, with the availability of annotated dialogue corpora, that research in this area has become possible.

One approach that has been tried for dialogue act tagging is the use of n-gram language modelling, exploiting principally ideas drawn from the area of speech recognition. For example, (Reithinger and Klesen, 1997) have applied such an approach to the VERBMOBIL corpus, which provides only a rather limited amount of training data, and report a tagging accuracy of 74.7%. (Stolcke et al., 2000) apply a somewhat more complicated HMM method to the SWITCHBOARD corpus, one which addresses both the sequencing of words *within* utterances and the sequencing of dialogue acts *over* utterances. They use a single split of the data for their experiments, with 198k utterances for training and 4k utterances for testing, achieving a DA tagging accuracy of 71% on word transcripts. These performance differences, with a higher tagging accuracy score for the VERBMOBIL corpus despite significantly less training data, can be seen to reflect the differential difficulty of tagging for the two corpora.

A second approach that has been applied to dialogue act modelling, by (Samuel et al., 1998), uses transformation-based learning over a number of utterance features, including utterance length, speaker turn and the dialogue act tags of adjacent utterances. They achieved an average score of 75.12% tagging accuracy over the VERBMOBIL corpus.

A significant aspect of this work, that is of particular relevance here, has addressed the automatic identification of word sequences that might serve as useful dialogue act cues. A number of statistical criteria are applied to identify potentially useful word n-grams which are then supplied to the transformation-based learning method to be treated as 'features'.

3 Simple DA Classification

In previous work, we describe our simple approach to DA classification, based on intra-utterance features, together with our experiments to evaluate it (Webb et al., 2005). A key aspect of the approach is the selection of word n-grams to use as cue phrases in tagging. (Samuel et al., 1999) investigate a series of different statistical criteria for use in automatically selecting cue phrases. We use a criterion of *predictivity*, described below, which is one that Samuel *et al.* do not consider.

Predictivity values are straightforward to compute, so the approach can feasibly be applied to very large corpora. As we shall see, predictivity scores are used not only in selecting cue phrases, but also directly as part of the classification method.

Dialogue Act	*% of corpus*	*Dialogue Act*	*% of corpus*
statement-non-opinion	36%	action-directive	0.4%
acknowledge	19%	collaborative completion	0.4%
statement-opinion	13%	repeat-phrase	0.3%
agreeaccept	5%	open-question	0.3%
abandoned	5%	rhetorical-questions	0.2%
appreciation	2%	hold before answer	0.2%
yes-no-question	2%	reject	0.2%
non-verbal	2%	negative non-no answers	0.1%
yes answers	1%	signal-non-understanding	0.1%
conventional-closing	1%	other answers	0.1%
uninterpretable	1%	conventional-opening	0.1%
wh-question	1%	or-clause	0.1%
no answers	1%	dispreferred answers	0.1%
response acknowledgement	1%	3rd-party-talk	0.1%
hedge	1%	offers, options commits	0.1%
declarative yes-no-question	1%	self-talk	0.1%
other	1%	downplayer	0.1%
backchannel in question form	1%	maybeaccept-par	$< 0.1\%$
quotation	0.5%	tag-question	$< 0.1\%$
summarisereformulate	0.5%	declarative wh-question	$< 0.1\%$
affirmative non-yes answers	0.4%	apology	$< 0.1\%$

Figure 1: SWITCHBOARD dialogue acts

3.1 Experimental corpus

For our experiments, we used the SWITCHBOARD data set of 1,155 annotated conversations. The dialogue act types for this set can be seen in (Jurafsky et al., 1997). Altogether these 1,155 conversations comprise in the region of 205k utterances.

The corpus is annotated using an elaboration of the DAMSL tag set (Core and Allen, 1997), involving 50 major classes, together with a number of diacritic marks, which combine to generate 220 distinct labels. (Jurafsky et al., 1998) propose a clustering of the 220 tags into 42 larger classes, listed in Figure 1, and it is this clustered set used both in the experiments of (Stolcke et al., 2000), and those reported here.

We used 198k utterances for training and 4k for testing, with pre-processing to remove all punctuation and case information, in common with (Stolcke et al., 2000) in order that we might compare figures.

Some of the corpus mark-up, such as filler information described in (Meteer, 1995), was also removed.

Our experiments use a cross-validation approach, with results being averaged over 10 runs. For our data, the test set is much less than a tenth of the overall data, so a standard ten-fold approach does not apply. Instead, we randomly select dialogues out of the overall data to create ten subsets of around 4k utterances for use as test sets.

In each case, the corresponding training set was the overall data minus that subset. In addition to cross-validated results, we also report the single highest score from the ten runs performed for each experimental case. We have done this to facilitate comparison with the results of (Stolcke et al., 2000).

3.2 Cue Phrase Selection

For our experiments, the word n-grams used as cue phrases during classification are computed from the training data. All word n-grams of length 1–4 within the data are

considered as candidates. The phrases chosen as cue phrases are selected principally using a criterion of *predictivity*, which is the extent to which the presence of a certain n-gram in an utterance is predictive of it having a certain dialogue act category. For an n-gram n and dialogue act d, this corresponds to the conditional probability: $P(d \mid n)$, a value which can be straightforwardly computed. Specifically, we compute all n-grams in the training data of length 1–4, counting their occurrences in the utterances of each DA category and in total, from which the above conditional probability for each n-gram and dialogue act can be computed. For each n-gram, we are interested in its *maximal* predictivity, i.e. the highest predictivity value found for it with any DA category. This set of n-grams is then reduced by applying thresholds of predictivity and occurrence, i.e. eliminating any n-gram whose maximal predictivity is below some minimum requirement, or whose maximal number of occurrences with any category falls below a threshold value. The n-grams that remain are used as cue phrases. It should be obvious that the levels of these two thresholds, frequency and predictivity, are crucial to the performance of the system.

3.3 Using Cue Phrases in Classification

The selected cue phrases are used directly in classifying previously unseen utterances in the following manner. To classify an utterance, we identify all the cue phrases it contains, and determine which has the highest predictivity of some dialogue act category, and then that category is assigned. If multiple cue phrases share the same maximal predictivity, but predict different categories, one category is assigned arbitrarily. If no cue phrases are present, then a default tag is assigned, corresponding to the most frequent tag within the training corpus.

3.4 Experimental cases

In previous work (Webb et al., 2005) we performed five different experiments using a variety of simple word processing techniques. The model which gained the best results used a corpus clustered into 42 dialogue act classes, had special tags marking the beginning and end of each utterance, had models trained for different lengths of user utterances and removed some of the effects of disfluencies from the corpus. Our best reported figures on the 202k utterance corpus are a cross-validated score of 69.09%, with a single high score of 71.29%, which compares well with the (non-cross-validated) 71% reported in (Stolcke et al., 2000).

In each experiment, there are two important variables used to select n-grams as potential cue phrases - the frequency of occurrence of each n-gram, and the notion of how predictive a particular n-gram is of some dialogue act.

The values of these variables were set in an arbitrary manner, selecting most likely candidates through prior knowledge of experiments. In the experiments reported in (Webb et al., 2005), these are a minimum frequency count of 2, and minimum predictivity score of 30%. N-gram cues with scores lower than these thresholds were discarded from the possible set used for classification.

This approach has a number of inherent problems. First, we do not know if there are some other values which will work better. The scores we used were chosen following extensive work with a 50k utterance training set - it is possible these pre-set threshold values would no longer be optimal when used with larger training sets.

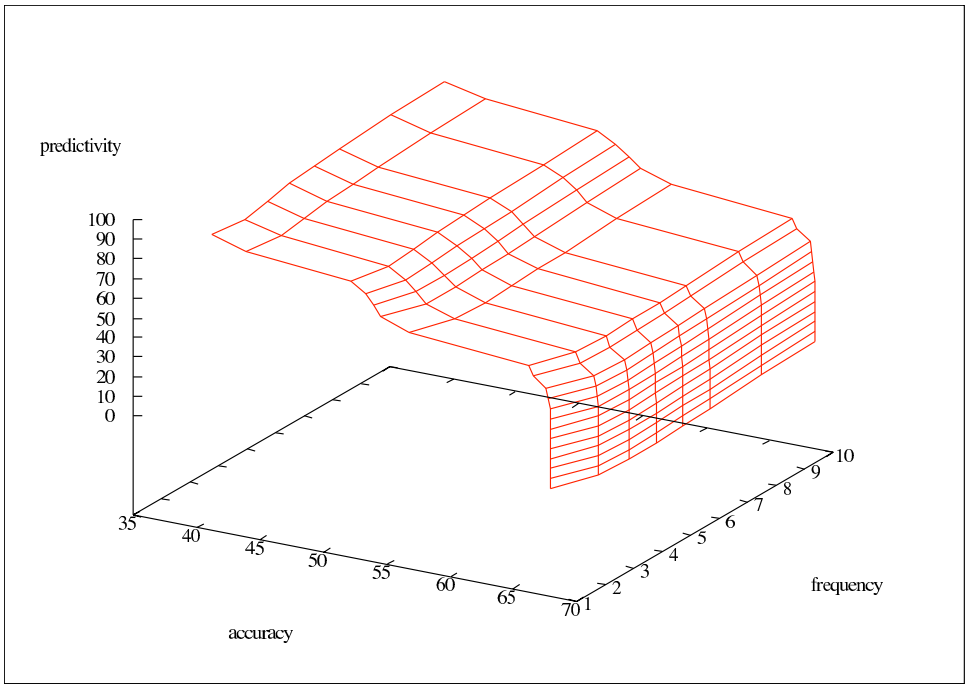


Figure 2: Effects of predictivity and frequency on tagging accuracy

Secondly, these values were chosen for their ability to perform well over the test data. Such an approach undermines our attempts to establish a baseline for classification performance. Our subseqeuent experiments aim to address these problems directly.

3.5 Exhaustive Thresholds

To address the two concerns we sought to develop a method that would determine thresholds automatically, as part of the training process, through the use of a validation set. As a prelude to this, we investigated how the performance of the classification approach interacts with the selection of thresholds, by computing performance results at an exhaustive range of threshold values.

For this search we computed scores for frequency count thresholds of 1, 2, 3, 4, 5, 6, 8 and 10. For predictivity, all scores from 0 to 100% were used in steps of 5%, for each of the possible frequency cut-offs.

Figure 2 shows the effect on tagging accuracy of varying thresholds. A quick interpretation of this graph shows that the classifier performs well with minimum predictivity thresholds of 40% and below, but falls rapidly for thresholds above that value. Although the classifier performs optimally with a frequency threshold around 2 or 3, the behaviour is tolerant of higher thresholds.

As can be seen in Figure 3, the best cross validated accuracy scores occur at a frequency count of 3, minimum predictivity of 35%. This score is higher than our manually selected thresholds of frequency 2, predictivity 30%, although the effective gain is 0.17%

Additionally, this single highest score oc-

Freq	*Pred*	*Cross Validated Score*	*Single Best Score*
2	30	69.48%	74.89%
3	30	69.65%	**74.95%**
3	35	**69.65%**	74.92%

Figure 3: Experiments with 202k data set

curs at 30% predictivity, although again the difference is extremely low, at 0.06%. It is worth noting that the figures quoted here for both cross-validation and single highest score are greater than our best published figures to date, and the highest score is 3.95% higher than that reported in (Stolcke et al., 2000).

3.6 Validation Model

We recognise that selecting thresholds manually by performance on the test set may not be a robust method for this task. To counter this, we split training data into two parts - training and validation. After training is complete, we will validate on the second part of the data, to automatically select the best values for minimum frequency and predictivity counts. This directly addresses the original problem of setting values based on the test data.

Experimentally, we now take the 198k utterance training set, and take 10% (around 20k utterances) to use for validation, a set distinct from the 4k utterances used for testing. We derive n-grams from the 178k training set, then do exhaustive testing over the validation set, using the range of variables described in the previous experiment. These experiments select the best performing combination of frequency and predictivity scores which are then used when applying the n-grams to the test set. We repeat this 10 times, using a random selection of dialogues for both the validation and testing data sets. In each case, we also tag the test data using our original, arbitrary values of frequency 2, predictivity 30%, to establish some kind of baseline.

The average frequency count selected by our automatic method is 2.9, average minimum predicitivity of 32.5%. The cross-validated tagging accuracy when classifying using these automatically selected thresholds is 67.44% (with a high score of 70.31%). This compares favourably to the cross-validated score of 67.49% (high score 70.72%) obtained using our static, manually prescribed thresholds on the same data splits. These results are perhaps not surprising given the previous experiment, which seems to demonstrate a broad range of values for these thresholds over which tagging accuracy is largely unaffected.

These overall cross-validated scores seem to be down on other reported scores - this could be due in part to the loss of training data caused by the creation of the validation set. However it is encouraging to see that we can use the validation data to select scores which perform well over the test data.

4 Discussion, Future Work

We have shown that a simple dialogue act tagger can be created that uses just intra-utterance cues for classification. This approach performs surprisingly well given its simplicity. The model appears to be robust, given that there is a range of possible values which combine to give good tagging accu-

racy scores. We are able to determine the settings for these variables independently from the test data.

Future work include a thorough investigation of the effects of the amount of data available for training, and the most effective size of validation set. Further, an error analysis of the data, to determine which dialogue act classes are most easily confused, would be interesting.

Clearly one next step is to pass these results to some machine learning algorithm, to exploit inter-utterance relationships. In the first instance, Transformation-Based Learning (TBL) will be investigated, but the attractiveness of this approach to previous researchers (Samuel et al., 1998; Lager and Zinovjeva, 1999) was in part the tolerance of TBL to a potentially large number of features. We will use our naive classification method to pass as a single feature our best-first guess.

References

Mark G. Core and James Allen. 1997. Coding dialogs with the DAMSL annotation scheme. In *AAAI Fall Symposium on Communicative Action in Humans and Machines*, MIT, Cambridge, MA.

Barbara Grosz and Candace Sidner. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 19(3).

Julia Hirschberg and Diane Litman. 1993. Empirical Studies on the Disambiguation of Cue Phrases. *Computational Linguistics*, 19(3):501–530.

D. Jurafsky, R. Bates, N. Coccaro, R. Martin, M. Meteer, K. Ries, E. Shriberg, A. Stolcke, P. Taylor, and C. Van Ess-Dykema. 1997. Automatic detection of discourse structure for speech recognition and understanding. In *Proceedings of the 1997 IEEE Workshop on Speech Recognition and Understanding, Santa Barbara.*

Daniel Jurafsky, Rebecca Bates, Noah Coccaro, Rachel Martin, Marie Meteer, Klaus Ries, Elizabeth Shriberg, Andreas Stolcke, Paul Taylor, and Carol Van Ess-Dykema. 1998. Switchboad discourse language modeling project final report. Research Note 30, Center for Language and Speech Processing, Johns Hopkins University, Baltimore.

Torbjörn Lager and Natalia Zinovjeva. 1999. Training a dialogue act tagger with the μ-TBL system. In *Proceedings of the Third Swedish Symposium on Mulitmodal Communication, Linköping University Natural Language Processing Laboratory.*

Marie Meteer. 1995. Dysfluency annotation stylebook for the switchboard corpus. Working paper, Linguistic Data Consortium.

Norbert Reithinger and Martin Klesen. 1997. Dialogue act classification using language models. In *Proceedings of EuroSpeech-97.*

Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. 1998. Dialogue act tagging with transformation-based learning. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, Montreal.

Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. 1999. Automatically selecting useful phrases for dialogue act tagging. In *Proceedings of the Fourth Conference of the Pacific Association for Computational Linguistics, Waterloo, Ontario, Canada.*

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. In *Computational Linguistics 26(3), 339–373.*

Nick Webb, Mark Hepple, and Yorick Wilks. 2005. Dialogue Act Classification Based on Intra-Utterance Features. Research Memorandum CS-05-01, Department of Computer Science, University of Sheffield.

On Plurals and Overlay*

Massimo Romanelli and **Tilman Becker** and **Jan Alexandersson**
DFKI GmbH
Stuhlsatzenhausweg 3, d-66123 Saarbrücken, Germany
{romanell,becker,janal}@dfki.de

Abstract

Using an ontology for the representation of user intentions in a multimodal dialogue system is a proven, flexible and powerful approach used in many implementations. Previous work has shown the advantages of representing instances of the ontology as typed feature structures and then using default unification to model the changes of the current user intentions. However, some scenarios require the use of sets to represent plurals and neither can typed feature structures naturally represent sets nor does standard default unification compute the intended results. To address this issue, we propose an extension of overlay by suggesting a representation of plurals, how to identify the intended set manipulations from the linguistic structure, extending the operational semantics for default unification and, finally, how to compute a score mirroring the success of the operation.

1 Introduction

The development of large ontologies makes it possible to use them for not only modeling the domain but also for the representation of the actions and intentions of a user and the system in real dialog systems. In previous work on using ontologies for this purpose, we show how, by simplifying the ontology to typed feature structures (TFS) it is possible to use unification-like operations for constructing new user intentions based on contextual—defeasible—information and new—strict—information (Alexandersson and Becker, 2003). This approach has been previously addressed by, e. g., (Grover et al., 1994) but also recently used in real multimodal systems, such as the MATCH system (Johnston et al., 2002).

However, the task of representing and manipulating plural-like objects is not addressed and we will focus on that in this paper. The treatment of plural-like objects is necessary in systems providing, for instance, the ability to book seats in a movie theater. Pluralities in general is a huge research topic, and we will, in this paper, restrict our focus on concrete, usually definite descriptions of sets.

The following example shows how sets of domain objects, here seats in a movie theater, need to be manipulated to represent the current state of the user intention.

(1) **U:** "I'd like to reserve these ↗[1] seats".

(2) **S:** Is this OK?

(3) **U:** "No, two more seats here ↗"

The interpretation of the utterance in (3) is obviously something like "I'd like to reserve the seats "the system" understood plus these two I'm pointing at".

In the following, we assume that meaning is analyzed on a pragmatic level, e.g., as instances of an appropriate ontology, that in turn is represented through typed feature structures (TFS) and that combining new (strict) information as in (3) with the previous context can be computed with default unification. Based on a lattice spanned by subsumption of TFS, (Carpenter, 1993) characterizes credulous default unification as the set of unifiers

*The research presented here is funded by the German BMBF under grants 01 IL 905 (SmartCom) and 01 IMD 01 (SmartWeb) by the EU under the grant FP6-506811 (AMI).

[1] Where "↗" stands for a pointing gesture, indicating the seats on a floor plan.

between the strict structure and the most special generalizations of the defeasible structure such that there exist a unifier between the strict structure and the generalized defeasible structure.

The representation of sets in TFS has been addressed by (Pollard and Sag, 1994) and recently by (Richter, 2004). The type hierarchy for sets proposed by (Richter, 2004) is shown in figure 1. Clearly, applying default unification on sets represented in this fashion will produce unpredictable results. There are two reasons for this. First, it is unclear how a lattice should be constructed such that the characterization of credulous default unification above produces the intended result. Second, applying our previous algorithm (see (Alexandersson and Becker, 2003)) the background will, in general, be overwritten, since in the subsumption hierarchy the sets are represented as nested sequences and default unification treats them blindly as such.

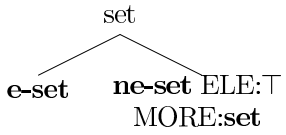


Figure 1: The type hierarchy for sets proposed in (Richter, 2004)

Within the semantic community, there is no generally accepted wide-coverage theory on how to represent and process plurals. The pioneer work on the formal representation and processing of plurals is due to (Link, 1983) where a formal model for the treatment of plurals is developed. His model has received extensions, e. g., (Landman, 1989) as well as critique, e. g., (Copestake, 1992). We will continue along the line of, e. g., (Kamp and Reyle, 1993; Schwarzschild, 1996) and use a union-based model for the representation and manipulation of plural entities.

The paper addresses the following points to include plurals into the previous approach: we identify surface forms relevant for the treatment of plurals and provide a mapping of their meaning to operations on the underlying representations; we extend the ontology and its representation as typed feature structures to represent sets and define the necessary operations on them. The paper provides an extension of the work in (Alexandersson and Becker, 2003), providing an in-context interpretation of plurals. This also requires an extension of scoring which is discussed in section 7.

Note, that this paper focuses on definite descriptions of sets that are changed in the discourse. Although constraints can be expressed as underspecified TFS, arbitrary constraints on a set, e.g., "I want ten or twelve seats somewhere in the front and not on the side." cannot in general be treated in the same way.

2 Phenomena

We extend the dialogue scenarios set out in (Alexandersson and Becker, 2001) and present a number of examples motivating this work. In a typical dialogue, the user intention is composed from the interpretation of user contributions over several turns. The user adds or changes information either because the system has requested the information, for instance, in order to access a database or because the user has refined or even modified his intention.

By using default unification, it is possible to add information inconsistent with the context in a very natural way:

(4) **U:** "What is running on TV tonight?"

(5) **S:** "Here you see a list of broadcasts for tonight"

(6) **U:** "and tomorrow?" (→ *what is running on TV tomorrow*)[2]

Plurals occurs naturally for, e. g., the reservation of seats in a movie theater:

(7) **U:** "I'd like to reserve these ↗ four seats".

but also for operating a video recorder:

(8) **U:** "I'd like to record these ↗ broadcasts".

As soon as we introduce sets into the TFS, default unification will not do the intended thing: using the representation suggested in (Pollard and Sag, 1994; Richter, 2004), the results become unpredictable. For instance, the intended behavior for the following continuation of (7) is not to overwrite the set in the defeasible structure:

(9) **S:** "Is this OK?"

(10) **U:** "No, two more seats here ↗"

Instead, the two sets should rather be combined with set *union*. There are other manipulations possible, such as *subtract*:

(11) **U:** "No, not these ↗two."

or *overwriting*:

[2] In this paper we skip the discourse processing required to resolve, for instance, ellipses. These tasks have been presented elsewhere, e. g., (Löckelt et al., 2002; Pfleger et al., 2003).

(12) **U:** "No, these ↗ instead."

or even a combination of subtraction and adjoining, which we will call *substitution*:

(13) **U:** "This ↗ one instead of these ↗ two"

It is thus necessary to extend the default unification operation by detecting (and storing) the type of manipulation from the surface structure to the position in the representation where the actual set manipulation takes place, see section 6.

3 Model

In his analysis of plural and mass terms Link (Link, 1983) argues that the relations between individuals and groups in the discourse domain can be captured with two basic operations defining individual sum (**isum**,$\oplus$) and individual part (**ipart**,Π). He observes that $\oplus$ corresponds to the join operation and Π corresponds to a partial ordering relation in a lattice. Moreover, the analysis of Link provides two lattices for the representation of individual and material parts. (Kamp and Reyle, 1993) explicitely adopt Link's approach of a lattice model of plurality and restrict their interest to Link's individual lattice for the representation of count nouns. We follow this approach and retain only the individual lattice. This way the domain of individuals can be represented in a lattice, ordered under $\oplus$ like in figure 2.

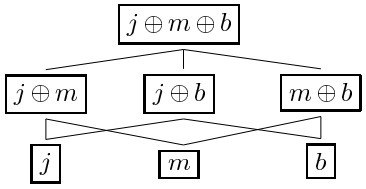


Figure 2: Link's complete join semi-lattice representing plural entities. j,b,m represent respectively the individuals John, Bill and Mari.

Landman (Landman, 1989) introduces an isometric set-theoretic model that captures the properties of the individual lattice in Link's approach. Landman argues that, given a set A of individuals, the power set of A without the empty set—$pow(A)\backslash\{0\}$—has the structure of a complete atomic join semi-lattice as in the Link's formalization of the individual lattice, where $\oplus$ corresponds to **union** and Π to **subset**.

(Schwarzschild, 1996) discusses whether there is a need for structured sets like in Landman's theory—requiring a higher order machinery—or, if it is sufficient, to use simple sets—called the union approach—and seems to advocate the latter. Schwarzschild discusses mostly problems of reference and intra-sentential cases, such as

(14) Tom and the boys reserved tickets.

Here, the question is how the resulting set is represented, and the union theory suggests—using Quine's innovation—that the semantics is $Tom \cup \{b_1, \ldots, b_n\} = \{Tom, b_1, \ldots, b_n\}$. Quite the same considerations on the opportunity of a structured set or simple set representation can be found in (Lasersohn, 1995), whereby the author tends to prefer the union theory solution. In our case, it is even simpler since every position in the ontology (represented as TFS) allows for either singular entities, such as one reservation, or sets of entities such as seats. Therefore, we always represent seats, no matter how many they are, as a set of one or more seats. Hence, we will never have the case that we have to combine a singular—represented as a single item—with a set, or structured sets with each other.

4 Approach

We make use of the representation of sets proposed by (Richter, 2004). Note, that in this representation, each position in the TFS is either of type set or not—it is not possible to represent either a singular or a plural. Also, the nodes of type set are not subject to (default) unification but rather to a special operation (e.g., union) as defined in figure 3. Thus, it is necessary in our approach to find the intended set-manipulation operation (see section 6) during analysis, which we then store to be able to perform the corresponding manipulation. We consider the following possible manipulations:

Union Utterances (7) and (10) are an example of union. The original set is enlarged with two more seats. Our implementation models this as *set union* of the original and the new set.

Difference Utterance (11) is an example of subtraction. This is implemented as *set difference*.

Overwrite See utterance (12). For the implementation, the set in background is discarded and replaced by the set in the covering.

Merge This is a variant of union. Union assumes that the new set is disjoint from that in the background, whereas merge allows for a nonempty intersection. In the implementation this is also realized as *set union*.

Substitute Utterance (13) is an example of substitution, assuming that the background set has more than two seats. Substitution requires that the user identifies *two* new sets: a subset of the set in background and its replacement. It is a combination of difference and union, where the set in context is subtracted by the first set using difference as described above. The result is then combined with the second set using union (set union).

5 Formalization

Our formalization presented here, differs slightly from that taken in (Alexandersson and Becker, 2003) in that we view atomic values as special cases of types, e. g., an integer, say, 42, is a feature-less sub-type of the feature-less type Integer. In this way, a type clash represents the unique source of failure for unification. Furthermore, we assume that there are no re-entrant structures.[3] Operations over set-TFS are specified over the member relation as in (Richter, 2004). Figure 3 gives the formal characterization for, e.g., the union operation. In our approach, testing for membership requires a definition of equality which we replace by unifyability.

Following (Alexandersson and Becker, 2001), the complete overlay consist of two steps:

Assimilation In order to guarantee that the two TFSs are in a direct subsumption relation (BG subsumes CO or vice versa) in a preprocessing step we perform an assimilation operation over the two TFS as shown in figure 4. The assimilation operation generalizes the BG to the least upper bound (**lub**) of the two TFS, unless one subsumes the other. Otherwise it returns the two TFS unchanged. The result of assimilation is passed to the overlay (proper), where the recursive call is performed as shown in definition 1.

Overlay At this point we distinguish two cases: If the assimilated TFS are sets, the *overlaySet* operation is invoked, otherwise we assign the greatest lower bound of CO and BG (**glb**), i. e., their unifier, as type for the result and recursively build the result TFS. In the *overlaySet* operation we assume a function—*assoc*—that computes the set operation that has to be performed. Thus, depending on the value of the *assoc* operation result, Union, Difference or Overwrite, we respectively perform *set union*, the *set difference*, or we assign CO as result for the *overlaySet*.

Definition 1 Overlay
Let $CO = \langle Q_{co}, \bar{q}_{co}, \theta_{co}, \delta_{co}\rangle$ and $BG = \langle Q_{bg}, \bar{q}_{bg}, \theta_{bg}, \delta_{bg}\rangle$[4] be two TFS (covering and background) such that the assimilated TFS are $CO' = \langle Q_{co'}, \bar{q}_{co'}, \theta_{co'}, \delta_{co'}\rangle$ and $BG' = \langle Q_{bg'}, \bar{q}_{bg'}, \theta_{bg'}, \delta_{bg'}\rangle$ f a feature, $Flag \in \{\texttt{Union}, \texttt{Difference}, \texttt{Overwrite}\}$ and $assoc(< TFS, Flag >)$ the operation denoting the set operation to be performed, then overlay(CO, BG) is defined as:

overlay(CO, BG) :=

| **if** ($\bar{q}_{co'} \wedge \bar{q}_{bg'} \sqsupseteq$ Set)
| **then** *overlaySet*(CO', BG');
|
| **otherwise** *overlay*$'(CO', BG')$;

overlay$'(CO', BG')$:= $\langle Q_o, \bar{q}_o, \theta_o, \delta_o\rangle$
where

$\bar{q}_o := \bar{q}_{co}$,
$\theta o(\bar{q}_o) := glb(\theta co'(\bar{q}_{co'}), \theta bg'(\bar{q}_{bg'}))$
$\delta_o(f, \bar{q}_o) :=$

| **if** (f in CO$'$ and BG$'$)
| **then** $overlay(\delta_{co'}(f, \bar{q}_{co'}), \delta_{bg'}(f, \bar{q}_{bg'}))$;
|
| **else if** (f exists only in CO$'$)
| **then** $\delta_{co'}(f, \bar{q}_{co'})$;
|
| **else if** (f exists only in BG$'$)
| **then** $\delta_{bg'}(f, \bar{q}_{bg'})$;

overlaySet(CO', BG') := $\langle Q_o, \bar{q}_o, \theta_o, \delta_o\rangle = Res$
where

| **if** ($assoc(< CO', Flag >) =$ `Union`)
| **then** $Res := union(CO', BG')$;
|
| **if** ($assoc(< CO', Flag >) =$ `Difference`)
| **then** $Res := difference(BG', CO')$;
|
| **if** ($assoc(< CO', Flag >) =$ `Overwrite`)
| **then** $Res := CO'$;

□

Overlaying a set with a non-set TFS will always return the TFS in the cover, so that set-manipulation operations in overlay only apply if the two TFS are typed as set. In an extended definition, Quine's approach, see section 3, can be used to combine a non-set with a set, but only if warranted by the input, i. e., through the *assoc* function. Otherwise any combination of two non-sets

[3] We are currently working on an extension of our operational semantics for the correct treatment of such structures together with multiple inheritance as suggested in (Alexandersson and Becker, 2003).

[4] For details of the definition of TFS as graphs, see (Carpenter, 1992) and (Romanelli, 2005)

could be interpreted as union, countermanding our operational semantics, i. e., assimilation.

$$
\begin{aligned}
&\mathit{union}(x,y,z) \overset{\forall}{\Longleftarrow} \\
&\quad \forall a(\mathit{member}(a,z) \leftrightarrow \\
&\qquad (\mathit{member}(a,x) \vee \mathit{member}(a,y))) \wedge \\
&\quad \mathit{set\text{-}properties}(^{x}[\mathit{set}]) \wedge \\
&\quad \mathit{set\text{-}properties}(^{y}[\mathit{set}]) \wedge \\
&\quad \mathit{set\text{-}properties}(^{z}[\mathit{set}])
\end{aligned}
$$

Figure 3: The union operation as in the work of Richter. The *set-properties* relation establishes non-cyclicity, finiteness and unicity.

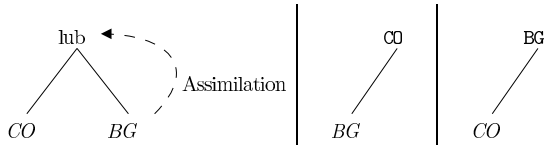


Figure 4: The three different cases for assimilation. lub is the least upper bound of the two TFS.

6 Linguistic Analysis

In this section we evidence the relation between a restricted number of uninflected words and modifications in the structure of plurals. With a lexical analysis that makes use of *lexical indicators* (see figure 5), we establish the relationship between the surface form and the semantic.

We ordered these lexical indicators depending on the operation they signal, union, difference, overwrite and substitute, as shown in table 1.

Some lexical indicators, e.g., "instead" are ambiguous wrt. their intended set operation. "Instead" can stand for a full replacement or only the substitution of a subset of the background. Further analysis is needed to disambiguate, which typically involves the discourse structure. In the case of "instead", this can be a test whether a subset of the background has been brought into focus before, e.g. "I want these ten seats. OK, but I don't like these two seats. I want these other two instead.", or not, e.g. "I want these ten seats. No, I want them in the first row instead." The former is a case of substitution, the latter of full replacement/overwrite.

We provide a sample grammar for German, see figure 6 based on the German word order analysis provided in (DUD, 1995). The grammar has only illustrative purpose and shows how the linguistic analysis could be performed. Note, that coordination is not covered by the grammar.

	Uni	**Diff**	**Mer**	**Ove**	**Sub**
and also	+	−	−	−	−
instead	−	+	−	+	−
rather	−	+	−	+	−
too	+	+	+	−	−
furthermore	+	−	−	−	−
additionally	+	−	−	−	−
further	+	−	−	−	−
further on	+	−	−	−	−
moreover	+	−	−	−	−
in addition	+	−	−	−	−
aside	+	−	−	−	−
in addition	+	−	−	−	−
instead	−	+	−	+	−
without	−	+	−	−	−
better	−	−	−	+	−
instead of	−	−	−	+	+
in place of	−	−	−	+	+
instead	−	−	−	+	+
alternatively	−	+	−	+	−
rather	−	+	−	+	−
another	+	−	+	−	−
only	−	−	−	+	−
not	−	−	−	+	−
too	+	+	+	−	−

Table 1: Lexical markers associated with set operations.

7 Scoring

The strength of overlay is the combination of default unification together with a scoring function (Pfleger et al., 2002; Alexandersson et al., 2004). The latter is necessary for actually using overlay in real dialogue systems where the analysis components produce multiple hypotheses. In the SmartKom system (Wahlster, 2003), their loci are indeed language and gesture recognition but also language interpretation produces multiple readings. Since default unification always succeeds, the scoring function makes it possible to choose the hypothesis that best fits the context. The score for overlay mirrors how similar two structures are by combining the amount of information stemming from cover or background together with type clashes and conflicting information.[5]

In the co-domain of $[1,-1]$, a positive score means roughly that the result is useful. However, when it comes to combining sets there is a new

[5]Note that, given the formalization presented in this paper, conflicting information will never occur. Instead this case is treated as a type clash.

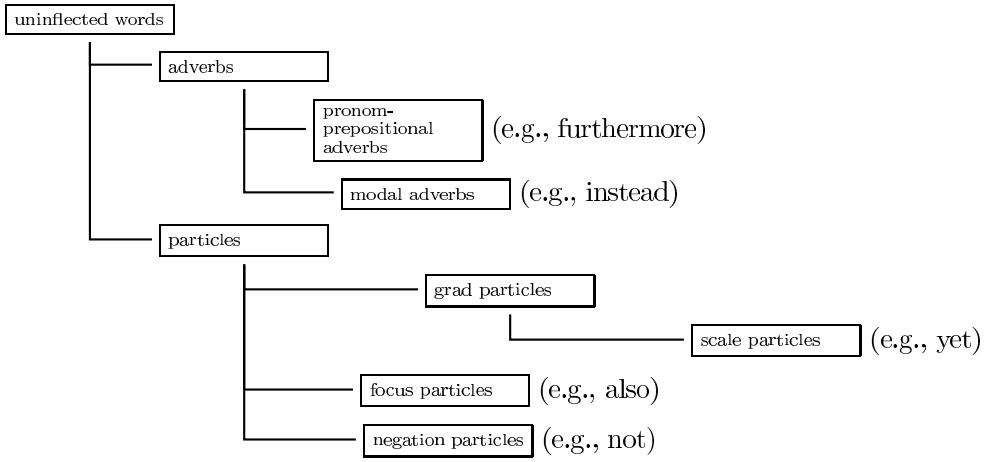


Figure 5: Word categories relevant to the modification of plural entities.

```
<S> ::= <frontfield>  <centerfield>  <endfield>

<frontfield> ::= [<np> <pf>]

<middlefield> ::= <pb> | <pa>

<endfield> ::= [<if>]

<pb> ::=   <particle> <deix> [<num>] [<obj>] [<gesture>]
         | <particle> <num> [<obj>] [<gesture>]
         | <particle> <gesture>

<pb> ::=   <deix> [<num>] [<obj>] [<gesture>] <particle>
         | <num> [<obj>] [<gesture>] <particle>
         | <gesture> <particle>

<np> ::= "ich"                              # I
<pf> ::= "moechte"                          # woud like
<particle> ::= "auch"                       # also
     | "ausserdem"                          # besides
     | "daneben"                            # aside
     | "dazu"                               # more
     | "desweiteren"                        # in addition
     | "hinzu"                              # to add
     | "ferner"                             # moreover
     | "noch"                               # another
     | "ueberdies"                          # too
     | "weiter"                             # further
     | "weiterhin"                          # further on
     | "zudem"                              # furthermore
     | "zusaetzlich"                        # additionally
<deix> ::= "diese"                          # these
<num> ::= "zwei"                            # two
<obj> ::= "plaetze"                         # seats
<gesture> ::= "hier"                        # here
<if> ::= "reservieren"                      # book
```

Figure 6: A grammar for the recognition of the Union case.

dimension. A set operation, e.g., difference, can be more or less successful. Suppose that an interpretation of (10), see section 2, contains seats not present in the cover. This indicates either that the interpretation was suboptimal, or it can be the case that the user said something inconsistent. The general schema for the possible relations between two sets is shown in figure 8.

The type of relation has different influence, depending on the type of set operation. For instance,

Op.	ScoreFn	Cons.	Incons.
Union:	$\lvert CO \cap BG \rvert$	2	1,4,5
Diff.:	$\lvert CO \backslash (CO \cap BG) \rvert$	4,5	1,2
Overw.:	$\lvert CO \cap BG \rvert$	2	1,4,5

Figure 7: The case analysis for the scoring operation for *overlaySet* depending on the relations between CO and BG, and on the set operation that has been performed. The **ScoreFn** column contains the operations used in the scoring functions. The numbers in the **Consistent** and **Inconsistent** columns refer to the cases depicted in figure 8, e. g., case 2 in figure 8 is consistent for **Union**.

the second case (2. in figure 8) is consistent for the union operation but not for difference. Consistent relations for the latter operation are 4. and 5., which are inconsistent for the former. Figure 7 shows the relation between the cases listed in figure 8 and the operations presented in section 1.

We supply two scoring functions

setScore for the inclusion into the overall score. setScore computes a value in the co-domain of $[-1, 1]$.

conScore (consistency score) for indicating that the operation has was inconsistent. conScore computes a value in the co-domain of $[-1, 1]$, where everything that is below or equal to 0 is more or less inconsistent. Note that it is up to the dialogue manager to utilize this information.

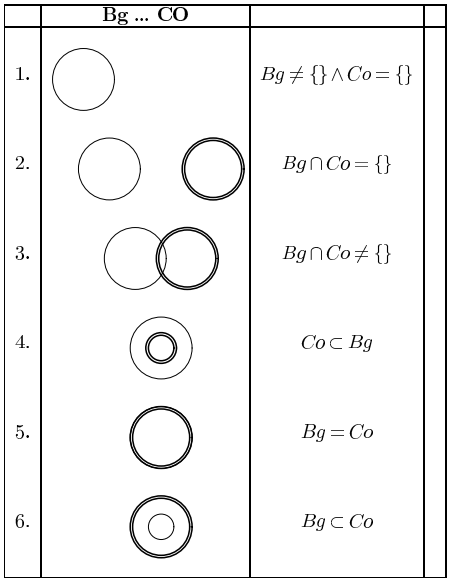


Figure 8: The different relations between the background and cover set respectively.

Definition 2 setScore

$$setScore(CO, BG, op) =$$

$$\begin{cases} \frac{|CO| - (|CO \cap BG| * 2)}{|CO|} & op = union \text{ or } overwrite \\ -1 & |CO| = 0 \\ \frac{|CO| - (|CO \setminus (CO \cap BG)| * 2)}{|CO|} & \text{otherwise} \end{cases}$$

□

Definition 3 conScore

$$conScore(CO, BG, op) =$$

$$\begin{cases} \frac{2}{|CO \cap BG| + 1} - 1 & op = union \text{ or } overwrite \\ -1 & |CO| = 0 \\ \frac{2}{|CO \setminus (CO \cap BG)| + 1} - 1 & \text{otherwise} \end{cases}$$

□

7.1 An example

To further highlight the behavior of our scoring functions, suppose the user has successfully selected five seats in a movie theater for reservation but suddenly remembers that the complete family should join and tries to add five more seats to the initial intention. The relation between the initial set and the new one will depend on the size of the intersection. The outcome of our scoring functions is depicted in figure 9.

For *conScore* we have 1 for the case that the intersection is empty and a value of 0 or less in case it is non-empty. *setScore* on the other hand is a linear function that uses the complete co-domain $[1, -1]$.

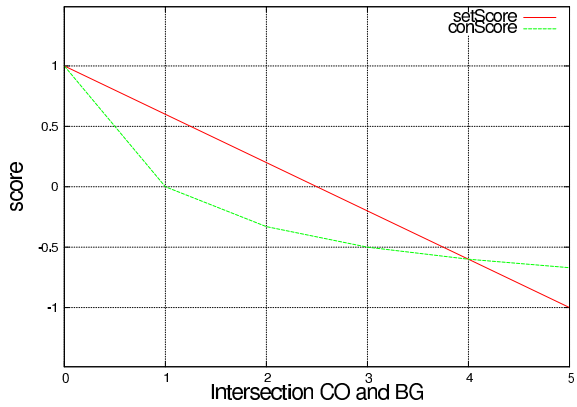


Figure 9: A plot representing the results of the different scoring operations depending on the size of the intersection.

Finally, it should be noted that the usage of the scoring functions, in particular, the conScore depends on the domain we are working with. If the task is to manipulate, say, grains on a plate, it might not be that inconsistent to make erroneously manipulations. For the selection of seats in a movie theater, however, every seat is important.

8 Conclusion

For the purpose of dialogue systems using a large ontology for the representation of user intentions, we have extended overlay (credulous default unification for TFS with scoring) to cope with sets. Sets are the natural modeling for plurals. We indicated how to extract information about the intended set manipulation from the surface structure. Additionally, we have sketched different possibilities to extend the scoring mechanism proposed in (Pfleger et al., 2002). Currently ongoing work is concerned with re-entrancy, the next step will be an extension of this work to general constraints on sets.

References

Jan Alexandersson and Tilman Becker. 2001. Overlay as the Basic Operation for Discourse Processing in a Multimodal Dialogue System. In *Workshop Notes of the IJCAI-01 Workshop on "Knowledge and Reasoning in Practical Dialogue Systems"*, Seattle, Washington, August.

Jan Alexandersson and Tilman Becker. 2003. The Formal Foundations Underlying Overlay. In *Proceedings of the Fifth International Workshop*

on Computational Semantics (IWCS-5), pages 22–36, Tilburg, The Netherlands, February.

Jan Alexandersson, Norbert Pfleger, and Tilman Becker. 2004. Scoring for overlay based on informational distance. In *KONVENS-04*, pages 1–4, Vienna,Austria, September 14 - 17.

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press, Cambridge, England.

B. Carpenter. 1993. Skeptical and Credulous Default Unification with Applications to Templates and Inheritance. In T. Briscoe, V. de Paiva, and A. Copestake, editors, *Inheritance, Defaults, and the Lexicon*, pages 13–37. Cambridge University Press, Cambridge, CA.

A. Copestake. 1992. *The Representation of Lexical Semantic Information*. Doctoral dissertation, University of Sussex.

1995. *DUDEN - Die Grammatik*, volume 4 of *Der Duden in zwölf Bänden*. Dudenverlag, Mannheim.

Claire Grover, Chris Brew, Suresh Manandhar, and Marc Moens. 1994. Priority Union and Generalization in Discourse Grammars. In *Proceedings of the* 32^{nd} *Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Las Cruces, New Mexico.

Michael Johnston, Srinivas Bangalore, Gunaranjan Vasireddy, Amanda Stent, Patrick Ehlen, Marilyn Walker, Steve Whittaker, and Preetam Maloor. 2002. MATCH: An Architecture for Multimodal Dialogue Systems. In *Proceedings of the* 42^{th} *Annual Meeting of the Association for Computational Linguistics, ACL'02*, pages 376–383, Philadelphia.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Kluwer Academic Publishers, London, Boston, Dordrecht.

Fred Landman. 1989. Groups I. *Linguistics and Philosophy*, 12:558–605.

Peter Lasersohn. 1995. *Plurality, Conjunctions and Events*, volume 55 of *Studies in Linguistics and Philosophy*. Kluwer.

Godehard Link. 1983. The Logical Analysis of Plurals and Mass Terms: A Lattice-theoretical Approach. In Rainer Bäuerle, Christoph Schwarze, and Arnim vov Stechow, editors, *Meaning, Use and Interpretation of Language*, pages 303–323. de Gruyter.

Markus Löckelt, Tilman Becker, Norbert Pfleger, and Jan Alexandersson. 2002. Making Sense of Partial. In *Proceedings of the sixth Workshop on the Semantics and Pragmatics of Dialogue (EDILOG 2002)*, pages 101–107, Edinburgh, UK, September.

Norbert Pfleger, Jan Alexandersson, and Tilman Becker. 2002. Scoring Functions for Overlay and their Application in Discourse Processing. In *KONVENS-02*, pages 139–146, Saarbrücken, September, 30–October, 2.

Norbert Pfleger, Ralf Engel, and Jan Alexandersson. 2003. Robust Multimodal Discourse Processing. In Kruijff-Korbayova and Kosny, editors, *Proceedings of Diabruck: 7th Workshop on the Semantics and Pragmatics of Dialogue*, Wallerfangen, Germany, September.

Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. Chicago Uni Press, Chicago - London.

Frank Richter. 2004. *A Mathematical Formalism for Linguistic Theories with an Application in Head-Driven Phrase Structure Grammar*. Phil. dissertation, Eberhard-Karls-Universität Tübingen. Version of April 28th, 2000. Superseded by Richter 2004.

Massimo Romanelli. 2005. Ontology-based representation and processing of plurals for human-machine dialogue systems with unification-based operations. Master's thesis, University of Saarland.

Roger Schwarzschild. 1996. *Pluralities*, volume 61 of *Studies in Linguistics and Philosophy*. Kluwer.

Wolfgang Wahlster. 2003. Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. In R. Krahl and D. Günther, editors, *Proceedings of the Human Computer Interaction Status Conference 2003*, pages 47–62, Berlin: DLR, June.

Conditional Anaphora

Henk Zeevat
ILLC
University of Amsterdam
Nieuwe Doelenstraat 15
1012 VB Amsterdam
henk.zeevat@uva.nl

Abstract

An alternative route for pronoun resolution is explored in which not the accessibility relation of discourse representation theory is taken as a starting point, but the idea of a single information state that supplies antecedents for pronouns. The single information state is obtained by a non-monotonic relation of overlaying. The approach is applied to a number of outstanding puzzles with pronouns where it gives a simple and uniform treatment.

The restrictions on accessibility of antecedents for anaphora formed one of the attractive sides of early DRT but —as was quite clear already at the time— is too restrictive. (**?**) gives many difficult examples, especially with plurals, but there is a host of other types of counterexample, like the modal subordination cases, the anaphora discussed in Asher, the Geach sentence (**?**), the examples in (**?**), the paycheck sentences and others.

In the basic DRT fragment, accessibility can be described as existence in the local context of interpretation of the pronoun. This local context can be described by recursion: it is the local DRS merged together with any higher DRS. One can quibble here about explicit existence as a discourse referent, or inferable existence. But that may well be conflating two kinds of restrictions on anaphoric relations: good antecedents do not merely exist in the local context, they also need to be maximally salient. And existents that are only inferable are almost per definition not salient enough. A realistic notion of pronominal antecedents combines the two: the antecedent is a salient local existent. So I take it that it is not formal presence of the discourse referent that explains why the tenth marble is not an antecedent, but a lack of salience[1].

If one considers extensions of the basic fragment to modals, corrections, or attitudes the local contexts need no longer be consistent. And this is a problem for the account I advocate. In inconsistent local contexts anything exists, in particular also the things that are not accessible for pronoun resolution but that are salient. The solution to this problem in DRT is technical: one defines accessibility not by local contexts but by geometrical configuration, as is done in certain kinds of syntax. The problem is however to explain the restrictions, not the descriptive definition.

The explanation above of accessibility in the basic fragment is not arbitrary. It follows

[1] Salience is way too crude a concept as a cursory glance at the study of pronoun resolution tells us, but it is good enough in the context of this paper.

a view of Ewan Klein (p.c.) on the development of context dependent semantics. First, in the work of Montague shortciteMontague and (**?**), one had lists of contextual parameters functioning much like Tarski's variable assignments in his definition of satisfaction. The revolution which can be attributed to (**?**; **?**; **?**) and citeHeim is that this set of parameters can be replaced by an information state and that the utterance itself partly determines what the information state for the next sentence or the subordinated material is. There are some adaptations necessary, like developing a theory of deixis (e.g. one can have a pointed information state that indicates the current utterance as one of its discourse referents), but this is an interesting view of what happened in the dynamic revolution. In this view, the possible discourse antecedents must be discourse referents of the local context of the pronoun.

Another problem arises when one considers an updated version of Montague's scheme:

$M \models \varphi[c, c']$

In the new theory c is the incoming context and $c' = c[\varphi]$, the update of c with φ. But what about M? Contexts that evolve in conversation are the works of humans and it cannot be excluded that imperfect knowledge will lead to error with respect to the state of affairs in which φ was uttered. The answer should be that it does not matter as long as there is a model on which φ, c and c' are true, i.e. they are all consistent and consistent with each other. Particular utterances may be false from an external point of view, but the conversational partners may not have noticed or have chosen to ignore the divergence. Only if the conversation gets trapped into inconsistencies, it cannot be a *bona fide* context of interpretation anymore and guide processes such as pronouns and presupposition resolution or disambiguisation.

So the revolution seems to have led to an improvement in our understanding of what a context is, of how the context influences interpretation and of how the communication itself changes the context. But apart from that it still follows the scheme of Montague. The context must be one simple object, e.g. a (pointed) information state and not some technical concoction out of information states, accessible discourse referents and whatever else.

So how about the extensions to modals, attitudes and corrections? The proposal of this paper is very simple. One should not merge the higher DRSs with the local one but only add so much of the higher DRSs as one can without becoming inconsistent. An operation doing this was invented by (**?**) (his satisfiable incrementation) which is a very cautious one: anything that could be inconsistent with anything is omitted. One can perhaps do better, but let's not worry about the refinements. This is Gazdar's definition applied to DRSs.

$K \cup ! \; K_1 = K \cup \{A \in K_1 : \neg\exists K_2 \subseteq K \cup K_1 : K_2$ is consistent and $K_2 \cup \{A\}$ is inconsistent$\}$

It is not necessary that $\cup!$ constructs the information state. One can keep the information states separate and merely use $\cup!$ to compute the information state that is coded by two information states or a sequence of information states. One then has one information state (the context of interpretation) and a sequence of information states that can be used to determine where the incoming new information goes (foregrounded material to the first element, accommodations towards the tail) and the result of the complete updates. So I assume that $\cup!$ merely gives the extension of a syntactic operation **over** that connects information states.

Local contexts can now be defined for corrections, modals and belief contexts in the following way.

Corrections

A: Bill ate the cake.

B: No, it was John who ate the cake.

1 **over** K where K is the old DRS and 1 the

empty DRS.
1 is updated by "It was John who ate the cake" using 1 **over** K or intermediate stages $K1$ **over** K as the context of interpretation (e.g. for the resolution of the presupposition "x ate the cake" that may well be resolved to part of the other speaker's utterance: Bill ate the cake). The updating process leads to a state $K2$ **over** $K3$ and its denotation $K2! \cup K3$ is the result of the correction.
Beliefs: John believes that S.
$bel(j, K)$ **over** K where K is the old DRS, $bel(x, K)$ the DRS representing the beliefs that x has according to K. S updates $bel(x, K)$ with the indicated context of interpretation that may be changed by accommodations. The result of the update $K1$ may be entered into the (possibly changed) K as a condition $belief(j, K1 \setminus bel(x, K))$.
Modals: It might have been that S.
1 is updated by S using 1 **over** K as he context of interpretation and the result can be stored under an appropriate modal operator.
counterfactuals: if A, would B
A uses 1 **over** K as context of interpretation, and B the result of updating 1 **over** K with A. If it is to be stored (rather than just checked?), this could be done by some suitable new syntax.

In corrections, initially the whole background is visible but correcting material will soon hide the corrected material. In beliefs, other beliefs that are in conflict with the common ground may hide parts of the common ground.

The proposal here leads to contexts of interpretation that make antecedents available that are not themselves identified by the local context: locally they may not be the object that is called "Bill", or the speaker's brother. The local contexts must identify these objects in their own ways, and the only claim seems to be that if the embedded context were true, the referents of the pronoun would be the same as the referents of the antecedents in the embedding context. In that sense, the modals, corrections, beliefs and suggestions can be said to be about the referents of their antecedents, if any. This is an approximation of the sense in which these modals and attitudes can be said to be about the objects that the embedding context is about. A lot can be said about this view of "quantifying in", but this is not the place. Suffice it to say that it conforms with a view in which the belief subject has her own mode of presentation and in which there is no criterion for when a mode of presentation is good enough for supporting "de re belief".

1a. John will come tonight.
1b. No, he is ill.
1c. Bill thinks that he will not
1d. He may be ill.
1e.If he forgot, he will not.

But the proposal can be extended. It is not problematic to take salient non-entailed material in the context (material that was denied, material that was merely reported or suggested) and add it in exactly the same way to the embedding context, i.e. the local material is added independently of its truth and can obliterate material from the context in which it is embedded. This allows pronominal reference to objects that the context itself is not committed to.

It gives an analysis of intentional anaphora. These are the famous cases like the Geach sentence or the examples provided by Edelberg. In all cases we, the conversationalists, know that there are no witches, that nobody had an accident and that Smith and Jones just had an accident. In (2b.), Harry arranged a fake accident by pushing a car against a tree and spraying tomato ketchup in the grass next to the driver's seat. John is the first who notices the car and reaches the conclusion that somebody had an accident. Mary arrives when John has left the scene, notices the ketchup and reaches her conclusion that the driver is wounded.

In (2c.) and (2d.) the two detectives are investigating the putative murders of Smith and Jones but have reached different conclusions. Arsky thinks that two different murderers were involved but Barsky thinks that only one murderer was responsible.

2a. Hob believes that a witch killed his pig and Nob believes that she poisoned his well.

2b. John thinks that someone had an accident and Mary thinks he was wounded.

2c. Arsky thinks that someone killed Smith and Barsky thinks that he killed Jones too.

2d. Barsky thinks that someone killed Jones and Arsky thinks that he killed Smith (too).

On the current proposal, the second clauses get interpreted in a context of interpretation that overlays the content of the belief that provides the antecedent over the embedding context. This provides an antecedent for the pronoun in the second clause, i.e. on the one hand, it licenses the speaker to use the pronoun because the intended referent is a highly activated member of the context for this part of the generation process and, on the other hand, it allows the interpreter to interpret the pronoun as standing for this highly activated member of the local context of interpretation.

The claim the speaker has to defend is that the amended context (the embedding context plus what Hob, John, Arsky or Barsky believes) fixes the referent of the pronoun in the second conjuncts: if the amended context were true, it would identify the object Nob, Mary, Barsky or Arsky has their belief about. This is true for (3b.) and (3c.) and false for (3d.) under the background that Edelberg provides. (in d. the first conjunct can be true because somebody killed Jones and nobody else and Arsky does not believe of that guy that he killed Smith). In (3a.) the counterfactual is true under any of the possible explanations that philosophers have provided: Hob telling Nob about his belief, a story in the newspaper, a rumour in the village etc. All these involve communication of some kind, but as Edelberg showed this is not essential. The real explanation is that a counterfactual is true: if the suggested material were true, the pronoun would refer to the same object as the antecedent.

Apparently the context that was computed for the antecedent belief is still available when the second belief report comes along and provides all that is needed for pronominal reference, independently of any need for that context in the update process for the second belief report.

It is also not hard to state the truth-conditional contribution of the anaphoric relation, provided there will be a day when a good truth conditional semantics for counterfactuals is available: if both beliefs were true, the referent of the antecedent would be the same as the antecedent of the pronoun. This is what is predicted by the complicated account in (**?**), but that account lacks a satisfactory intuitive foundation.

The account in this paper is not just another philosophical theory. It is a cognitive science hypothesis. In constructing a context of interpretation for beliefs, by overlaying a local context on top of the embedding context, a picture is formed of what it would be like if the belief were true. The pronoun picks up a referent from the picture. The interpretation of that referent is conditioned by the counterfactual obtained by counterfactually assuming the truth of the antecedent belief. The operation of overlaying one context with another is a natural ingredient of the semantics of counterfactuals. To the extent that the Ramsey test is not by itself the correct story about counterfactuals, it has a reflex on the account of this paper: Gazdar's operation is not a final truth about overlaying either. Any improvement should apply to both overlaying and counterfactuals.

The account straightforwardly extends to modal subordination and is not in conflict with the existing accounts of that, which however cannot deal with cases with two different

operators, like (3).

3. A wolf might come in. Bill thinks it is prowling about in the neighbourhood.

The counterfactual is: if a wolf would come in it would be the one that Bill thinks is prowling about. This forces specificity on both Bill's belief and the speaker's modal, but not existence. Both may come from an unfounded rumour, as in the Hob-Nob example.

More interesting are the like the paycheck sentence and Landman's famous (4).

4. If a farmer owns a donkey, he beats it. If he owns a horse, he treats it well.

Assume the second clause is interpreted with respect to a context of interpretation that still has: "a farmer owns a donkey" on top. Overlaying "he owns a horse" now functions as a correction and we can take the result of the correction as giving the content of the condition. This may be the same mechanism that operates in contrast under parallelism, as in the paycheck cases and squares well with the observation that both kinds of anaphora are limited to parallelism.

I started from the observation that discourse representation theory has been forced to give up one of the intuitions on which it is founded: a context of interpretation that is an information state and which supplies information to the interpretation process. I have explored a way to keep this information in and gave up instead monotonicity. It should not be controversial to assume that human cognition is able to do corrections and consider counterfactual states of affairs. It should therefore not be a surprise that one gains a new perspective on some quite old puzzles.

References

Walter Edelberg. 1992. Intentional identity and the attitudes. *Linguistics and Philosophy*, 15:561–598.

Gerald Gazdar. 1979. *Pragmatics: Implicature, Presupposition and Logical Form.* Academic Press, New York.

Peter Geach. 1962. *Reference and Generality*. Cornell University Press, Ithaca, New York.

Hans Kamp. 1981. A theory of truth and semantic representation. In Jeroen Groenendijk, Theo Janssen, and Martin Stokhof, editors, *Formal Methods in the Study of Language, Part 1*, volume 135, pages 277–322. Mathematical Centre Tracts, Amsterdam. Reprinted in Jeroen Groenendijk, Theo Janssen and Martin Stokhof (eds), 1984, *Truth, Interpretation, and Information; Selected Papers from the Third Amsterdam Colloquium*, Foris, Dordrecht, pp. 1–41.

David Kaplan. 1989. Demonstratives. In J. Almog, J. Perry, and H. Wettstein, editors, *Themes from Kaplan*, volume 135, pages 481–566. Oxford University Press, New York.

Lauri Karttunen. 1976. Discourse referents. In James McCawley, editor, *Syntax and Semantics 2: Notes From the Linguistic Underground*, pages 363–385. Academic Press, New York.

Robert Stalnaker. 1979. Assertion. In Peter Cole, editor, *Syntax and Semantics*, volume 9. Academic Press, London.

Henk Zeevat. 2001. Demonstratives in discourse. *Journal of Semantics*, 16:279–314.

Let's You Do That: Enquiries into the Cognitive Burdens of Dialogue

E. G. Bard,
TAAL, HCRC
U. of Edinburgh
Edinburgh EH8 9LL
ellen@ling.ed.ac.uk

A. H. Anderson
Dept of Psychology, HCRC
U. of Glasgow
Glasgow G12 8QB
anne@psy.gla.ac.uk

Y. Chen
TAAL, HCRC
U. of Edinburgh
Edinburgh EH8 9LL
yiya.Chen@let.ru.nl

H. Nicholson
TAAL, HCRC
U. of Edinburgh
Edinburgh EH8 9LL
hannele@ling.ed.ac.uk

C. Havard
Dept of Psychology, HCRC
U. of Glasgow
Glasgow G12 8QB
c.havard@psy.gla.ac.uk

Abstract

Most discussions of audience design assume that it rests on speakers' uptake of information about listeners' knowledge. The cognitive difficulty hypothesis (Horton and Gerrig, 2004 in press a) proposes that speakers provide less tailored design when the cognitive cost of uptake or recall increases. Yet the principle of mutual responsibility implies that cognitive load should to be shared efficiently: listeners should provide information which would be difficult for speakers to discover themselves. Two map task experiments examine speakers' uptake of information about listeners' knowledge and their responses to listeners' difficulties. Both experiments show that uptake is poor where it would be most useful: speakers attend very little to feedback in the form of simulated listener eye-tracks which directly indicate discrepancies between participants' knowledge. The second experiment shows that verbal feedback, though harder to interpret than gaze, generates more helpful responses in the form of Dialogue Transactions which correct listener errors and in the form of Game Moves which focus on listener knowledge. We propose that the instructor's priority is relating her own knowledge and that she will be deflected only when overtly called on to acknowledge a discrepancy between her own knowledge and the listener's.

1 Introduction

Recently experimental psycholinguists have given a great deal of attention to dialogue, with particular emphasis on the extent to which speakers design utterances for the benefit of their interlocutors. Audience design of this kind is taken to validate the notion of common ground in a psychological model of the process of conducting dialogues: if speakers maintain a model of their interlocutor's knowledge as well as their own, the intersection, the knowledge held in common, can be estimated[1]. A parallel line of research addresses common

[1] Strictly, common ground is only that shared knowledge which is mutually acknowledged as shared (Clark & Marshall, 1981; Barr & Keysar, 2004). We deal here with shared knowledge, both because it is usually what is at stake in the experimental literature and because it appears to be central to the view that we develop.

ground from the listeners' perspective, examining how a listener's knowledge about the what the speaker knows can affect that listener's interpretation of the speaker's referring expressions.

These experiments are based on several predictions involving the notion of common ground. The first gives every speaker responsibility for discovering what information is in common ground. To do this, it is predicted, each must at least attend to clues to the other's knowledge (Clark & Carlson, 1982, Clark & Krych, 2004). The second requires each speaker to exploit these cues when framing her own utterances. The third invokes the theory of mind in interpretation: it predicts that, as a listener, any interlocutor will consider only those candidate referents which he knows to be in or derivable from knowledge held in common.

Underlying this research is the assumption that common ground, the knowledge held mutually, will be established when each interlocutor performs two tasks: modeling the other's knowledge and maintaining her own. Clearly, one of these record keeping tasks is easier than the other: a participant's own experience can be recorded in episodic memory and can function via computationally inexpensive associative processes like priming (Pickering & Garrod, 2004) or resonance (Horton & Gerrig, in press b). The upkeep for a model of the interlocutor's knowledge can be much more costly (Bard & Aylett, 2004; Carletta & Mellish, 1996; Pickering & Garrod, 2004) and may involve chains of inferences about the interlocutor's actions, intentions or conceptions (Clark and Marshall, 1981). For this reason, dialogue is a joint project, a game for two players which can best be played if each player makes the contributions that keep the other player's task feasible. Though the principle of least collaborative effort (Clark & Wilkes-Gibbs, 1986) allows both players to make gradual contributions to the establishment of common ground, it is possible to go a step further.

Studies of audience design take the notion of joint responsibility for creating common ground to mean that each participant has full responsibility for maintaining and embellishing the models of both speakers. In many ways, this amounts to cost-duplication. The principle of least collaborative effort means that joint responsibility should be a kind of cost sharing, with players assuming not identical, but complimentary responsibilities (Carletta & Mellish, 1996). Each should attend to his or her own knowledge and present it to the other when necessary. In this view of joint responsibility, no interlocutor need be responsible for information which the other can provide more economically. This latter interpretation of joint responsibility seems to come close to Clark, Schreuder and Buttrick's (1983) definition of optimal design.

Thus, audience design, in the sense of adjusting one's contributions to what the interlocutor knows, is not an absolute requirement; nor is listener modeling principally the responsibility of the speaker. Instead, speakers can design their utterances as suits their current personal knowledge or the currently known common knowledge, without actively seeking additional detail about the listeners. It is the their listeners' responsibility to provide them with indications of their own share of common ground, drawing on cheap and cheerful own-knowledge record keeping. The Monitor and Adjust model of dialogue (Horton & Keysar, 1996), under which speakers monitor both their own output and their interlocutor's feedback, is similar in spirit. It makes slightly stronger assumptions about self-monitoring than this position does, and it follows Clark and Schaefer in concentrating on listeners' rejection particular utterances, rather on their own contribution to common ground.

In summary, then, the theory of dialogue as joint activity makes contradictory predictions. Where joint responsibility is duplicated responsibility, speaker A is responsible for tracking speaker B's knowledge. Where joint responsibility is

shared responsibility, B is responsible for revealing his pertinent knowledge to A. There is evidence for both positions.

On the one hand, Speakers monitor listeners' activity and gestures while speaking (Clark & Krych, 2004). Speakers maintain forms of referring expression with a particular interlocutor (Brennan & Clark, 1996) and are disrupted if that interlocutor chooses a different expression (Metzing & Brennan, 2003). Speakers initially provide more detail in description, particularly atypical detail, for listeners who cannot see the picture described (Lockridge & Brennan, 2002). Speakers incrementally supply descriptive phrases in the order in which they can most conveniently be used by listeners (Haywood, 2004). Listeners will interpret referring expressions as if addressed to them (Hanna, Tanenhaus, & Trueswell, 2003).

On the other hand, speakers may provide egocentric descriptions initially and audience-related descriptions somewhat later (Dell & Brown, 1991); habitually utter syntactically ambiguous structures, where unambiguous paraphrases are available (Ferreira & Dell, 2000); describe objects when under time pressure in ways which are unhelpful to listeners (Horton & Keysar, 1996); perform faster production adjustments egocentrically and slower ones with only modest care for the listener (Bard et al., 2000; Bard & Aylett, 2004); interpret referring expressions as naming objects salient to themselves but patently unknown to the speaker (Keysar, Lin, & Barr, 2003); require experience as an addressee in an object selection task before providing evidence of audience design in their own utterances (Haywood 2004).

To deal with these contradictions, Horton and Gerrig (2004 in press a) have recently proposed a difficulty model of common ground construction, under which listener modelling is subject to effects of the cognitive effort involved. Modelling will be slow or less complete when it is more difficult. Horton and Gerrig show that interlocutors adhere more closely to principals of audience design in a later dialogue when it is simpler to distinguish their co-participants in terms of the task pursued in an earlier dialogue.

The present paper asks whether audience design, cognitive difficulty, or joint responsibility controls behaviour in dialogue. We investigate this question in a route communication task where two variables affect cognitive load. One is the source and specificity of the information about the listener's knowledge state. One source is the direction of the listener's gaze. If A says "Go to the large oak tree" and sees B looking at the bridge instead, little inference is needed to devise a correction (get B from the bridge to the oak tree). The other source is typical verbal feedback. If, when told to go to the large oak, B replies 'Don't follow', A will not know whether B lacks the oak, has two small oaks, or cannot understand the instruction. Even 'Don't have it' could mask a mismatch of map landmarks or a misunderstanding of instructions. A chain of inferences and investigations are required to tailor a solution to the listener's problem. The second measure of cognitive load, time pressure, is used because remarkably egocentric behaviour can occur when speakers are pressed to respond quickly (e.g., Horton & Keysar, 1996), and much better audience design when they respond at leisure.

Listener modeling, the difficulty model, and joint responsibility make different predictions here. An assiduous modeler of common ground will attend to all sources of information about the listener's knowledge: a speaker who says 'Fine!' but is looking in the wrong place needs to be told that he has a problem. One who says 'Can't see it' and is apparently lookiing in the right place needs a different kind of help. This attention should be maintained as long as time pressure permits competent dialogues to be completed. If common ground is cultivated more when the cost of cultivation is less, speakers should attend

to visual feedback at least as assiduously as to verbal replies (Clark & Krych, 2004; Pomplun et al, 1997), should give proportionately more attention to feedback when unhurried than when rushed. Joint responsibility, however, predicts that processing cost, uptake, and audience design are not related. Instead, because listeners' verbal contributions to the construction of common ground are the key to joint action in dialogue, speakers may habitually ignore visual feedback and attend instead to their interlocutors' explicit demands.

2 Experiment 1

Experiment 1 tests the ability of visual feedback alone to supply the role of the listener. The direction of the interlocutor's gaze is an important component of co-presence. The ability to see where the interlocutor is looking greatly enhances the utility of virtual co-presence (Gale & Monk, 2000). Here, visual feedback is instantiated a way that allows us to determine when it is attended to: the simulated eye-track of a distant listener is projected onto the monitor showing the route which the participant describes and the participant's genuine eye-track is examined for time spent looking at the interlocutor's track.

2.1 Method

Materials: Four different maps of fictitious locations each included a route defined by a number of labeled cartoon landmarks. Eight or 9 route-critical landmarks were designated *correct* and 4 non-adjacent items were to be *missed*. Other, irrelevant landmarks assured that the Instruction Giver (hereafter 'IG') always had to distinguish a route-critical landmark from a number of others. To simulate the gaze of an Instruction Follower (hereafter 'IF'), a red square was superimposed on a sequence of landmarks with saccades of random length and direction outward from each fixation target. For *correct* landmarks the fixation target was the route-critical landmark itself. For *missed* landmarks, the fixation target was a *wrong* landmark,elsewhere on the map. Though target sequence was preprogrammed, migration was initiated by the experimenter as soon as the participant named the next route-critical landmark. To create a usesable trial response, the experimenter had to advance the IF feedback square between the IG's instruction to move to the new landmark and any instruction to correct the IF or to move to the following landmark. The feedback square returned to the route after a detour only when the participant gave the appropriate instructions or moved advanced to the next landmark on the route.

Apparatus: Participants viewed maps on a flat screen monitor at a distance of 60 cm. Eye movements were recorded on an SMI remote eye-tracking device placed on a table below the monitor and using Iview version 2 software. Speech was recorded in mono using Asden HS35s headphone /microphone combination headsets. Video signals from the eye tracker and the participant monitor were combined.

Design: All participants served as IG for all 4 maps, with 2 under a 1-minute time limit and 2 without limit. One map in each time pressure condition included visual feedback. The Time Pressure and Feedback combinations were applied to maps by Latin Square.

Procedure: Participants were met with a confederate and asked to take the role of IG while the confederate worked as IF in another room. IGs were asked to describe the route on each map to the IF so that the latter could reproduce it by using her mouse to traverse a similar screen displaying a similar but not identical map. The feedback and timing conditions were explained and announced before each trial. Participants were fully debriefed. None suspected the true nature of the experiment.

Participants were Glasgow University students (aged 17-24), all with normal or corrected to normal vision, all native English speakers, and all paid £5 for partici-

pating. Participants were rejected from the final set if eye-tracking capture fell below 80% of experiment time on any map or if the experimenter missed the critical time-window for moving the IF feedback square for any wrong item on a map or for more than 30% of the correct items. Testing continued until 24 participants passed these criteria and filled a balanced design.

2.2 Results

Interactive behaviour: To discover whether participants engage in something other than monologue with purely visual feedback, we coded their transcribed speech as Transactions and Conversational Game Moves (Carletta et al., 1997). A Transaction is as section of a dialogue which achieves an identifiable subgoal of a non-linguistic task. A new type of Transaction, a Retrieval, was identified, in which IG explicitly directed a lost IF back to the route. ANOVAs were calculated by subjects (F_1) and/or by items (F_2) as appropriate. Absent in the no feedback condition, Retrievals were found in the trials with visual feedback. The usual route-advancing 'Normal' Transactions accordingly fell in frequency between no feedback and visual feedback conditions (Feedback: $F_1(1,23) = 24.68$, $p < .001$). Conversational Game Moves are stages of the linguistic task which manipulate information and common ground. Moves which were specifically interactive in that they would not be expected in monologue (queries, aligns, acknowledges) were significantly more common with visual feedback than without (F_1 $(1,23) = 21.48$, $p < .001$). Time pressure affected only gross length of dialogues and amounts of gaze.

Attention to interlocutor knowledge: As the participants' speech had become more like dialogue in the feedback condition, listener modeling ought to be encouraging good uptake of cues to listener knowledge. Both where the 'IF's' gaze rested on the correct landmark and particularly where it digressed to an off-route landmark, IG should look longer at the targeted landmark than in the control condition, which lacked feedback. IG should look less at the route-critical landmarks which the IF missed, because her attention should be diverted to the landmark where IF appeared to be mistakenly gazing. In fact, a very different pattern emerged from mean total time spent gazing in the region of a landmark. As predicted, IGs looked longer at landmarks attracting correct IF gaze than at the same landmarks in the no-feedback condition, an average increase of 1.4 sec per landmark. Contrary to prediction, IG also looked significantly longer at on-route landmarks which IF missed (610 msec) but not at the distant 'wrong' landmarks under IF's gaze (430msec) (Landmark type: $F_1(2,46) = 4.10$, $p = .023$; Correct v wrong, $p < .05$). Relatively little time, then, was absorbed by attending to discrepancies between IG's and IF's knowledge. Instead, participants gazed at the on-route landmarks, whether IF's attention was directed to them ('correct landmarks') or not ('missed landmarks').

3 Experiment 2

Experiment 2 tests ease of absorbing listener-knowledge by comparing verbal to visual feedback.

3.1 Method

Materials comprised 6 maps, 4 derived from those used in Experiment 1 and 2 created to the same model.

Design: All participants used all 6 maps, 2 maps in each of three feedback conditions: no feedback, single channel (verbal for Group A participants, visual for those in Group B), and dual channel (verbal and visual). One trial in each modality condition was performed within a time limit of 2 minutes, while the other had no time limit. The order of feedback conditions was as just described in each time pressure condition. The order of time pressure conditions and the assignment of maps to condition were counterbalanced over the design.

Feedback was delivered once the subject had introduced each route-critical landmark. Verbal feedback was provided by the confederate according to a script: in negative replies, the confederate claimed not to be able to see the landmark, or follow the instruction, but did not explicitly describe any guess, location, or difficulty. Visual feedback was delivered as in Experiment 1. On each map, 7 to 9 route-critical landmarks received correct visual feedback with (concordant) positive verbal feedback (the IF 'gaze' was on the correct landmark and the IF said that it was); 3 landmarks had correct visual feedback and (discordant) negative verbal feedback, 3 had wrong visual feedback and (discordant) positive verbal feedback, 3 had wrong visual feedback and (concordant) negative verbal feedback.

Procedure and apparatus were as for the Experiment 1, with the addition of the extra conditions. Again, subjects were debriefed and the two participants who were suspicious about the true nature of the experiment were replaced.

Participants were 36 Glasgow University undergraduates, 18 per group, each paid £5. An additional 13 participants had been replaced because one of their 6 trials fell below the eye capture criterion.

3.2 Results

Results were coded as in Experiment 1 with the additions of new conditions.

Attention to interlocutor knowledge: Assiduous listener modeling regardless of difficulty would demand that IG track IF's gaze as well as attending to IF's verbal feedback**.** If difficulty discovering pertinent listener knowledge is critical, IG should track simpler visual information, especially where the IF's and IG's interpretations apparently diverge, and the feedback square moves to the wrong landmark. When visual and verbal feedback disagree (as they do in discordant conditions) visual feedback should take precedence: speakers should look at the wrong landmarks, even if verbal feedback is positive. Timed dialogues should show proportionately less attention to IF-only information. If, however, it is not the speaker's task to track the listener's knowledge, there is no particular attraction in divergent listener gaze.

The effects of both feedback and time pressure showed this last pattern. In total, IGs looked less at landmarks under time pressure (F_1 (1, 34) = 48.08, $p < 0.001$), but the reduction applied to all landmarks *except* the IF-specific wrong landmarks ($F_1(3, 116) = 13.83$, $p < 0.001$; $F_2(5, 126) = 11.773$, $p < .001$) where gaze durations were minimal throughout (< .295sec vs 5 to 6sec for all others). IGs looked longer as each feedback channel was added (F_1 (2, 59) = 329.95, $p < 0.001$), but feedback did not prolong gaze on the wrong landmarks ($F_2(10, 244) = 3.26$, $p < 0.01$). We checked Transactions including feedback for any examples of speaker gaze at listener position, no matter how brief. Table 1 shows that speakers more often than not (59% of trials) failed to look at the Follower feedback square at all when it targeted the wrong landmark, though they more often looked at it (55%) when it targeted the correct landmark which they were in the course of describing ($F_1(1,34) = 7.70$, $p = .009$).

Verbal feedback	Visual Feedback	
	Correct	Wrong
Positive	.51	*.45*
Negative	*.59*	.37

Figure 1. Proportion of feedback episodes attracting speaker gaze to feedback square: Effects of combinations of visual and verbal feedback in dual channel conditions (Italics represent discordant feedback)

Interactive behaviour: If speakers always tailor their output to interlocutors, any feedback should encourage interactive behaviour which solves listener problems. If cognitive difficulty affects audience design, then interactive contributions should be more common when speakers can ac-

cess simply processed visual indications of the listener's knowledge. If verbal feedback is key, as joint responsibility suggests, then it should attract interactive talk more than visual cues do. As Figure 2 below reveals, the third pattern holds.

Retrieval Transactions, which bring errant Followers back from places that can be seen with visual feedback, are far less common with unambiguous visual feedback alone (7% of opportunities) than with only ambiguous verbal feedback or with visual and verbal information that may conflict (27%) ($F_1(1,34) = 90.80$, $p < .001$, cell comparisons at $p < .05$). A similar pattern is found for Interactive Moves (6% v 30%: $F_1(2,68) = 36.53$, $p < .001$).

Dependent variable	Single channel	Feedback channels 0	1	2
Retrieval transactions	Verbal		.27	.27
	Visual		.07	.27
Interactive moves	Verbal	.00	.31	.34
	Visual	.01	.06	.25

Figure 2. Effects on rate of interactive behaviours from feedback channels and modality of single-channel condition

4 Conclusion

Two experiments have shown, first, that visual feedback alone can make speakers' instructions more like a dialogue and, second, that speakers did not pay close attention to direct visual evidence for their listener's problems, however simple it might have been to interpret. In fact, they took up this feedback only when it fell on route-critical landmarks which they were already fixating in order to describe the route. They avoided looking at the spots where their listener's gaze had mistakenly focused. In the second experiment, gaze showed a similar pattern. Moreover, though visual feedback gave clear evidence for the location of the lost IF, it was not sufficient to launch a rescue: both interactive Moves and Retrievals depended on the presence of verbal feedback either alone or in combination with visual.

The results do not sit comfortably with a model which demands continuous uptake of listener information. Nor do they show the responses to time pressure or ambiguity that might support a cognitive load model. Instead the results point to joint responsibility: Verbal feedback seems to be required to draw participants' attention to the problems at hand. Perhaps verbal feedback has this quality because an intentional signal of distress is needed to derail IG's inadequate descriptions. Or perhaps visual feedback is ignored because IG can simply wait for the IF to re-appear without knowing where or how he is lost. Clearly in this paradigm, responsibility for designing adequate instructions was jointly held.

Acknowledgments

This work was supported by EPSRC (UK) grant GR/R59038/01 to E. G. Bard and grant GR/R59021/01 to A. H. Anderson. Dr. Chen is now at Radboud Universiteit, Nijmegen.

References

Bard, E. G., Anderson, A., Sotillo, C., Aylett, M. Doherty-Sneddon, G., & Newlands, A. (2000). Controlling the intelligibility of referring expressions in dialogue. *JML, 42*, 1-22.

Bard, E. G., & Aylett, M. P. (2004). Referential form, word duration, and modeling the listener in spoken dialogue. In J. Trueswell and M. Tanenhaus (eds.), *Approaches to studying world-situated language use*. Cambridge, MA: MIT Press, pp. 173-191

Brennan, S. E., & Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *JEP: LMC, 11*, 1482-1493.

Carletta, J., & Mellish, C. (1996). Risk-taking and recovery in task-oriented dialogue. *J Pragmatics, 26,* 71–107

Carletta, J., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G. & Anderson, A. 1997. The reliability of a dialogue structure coding scheme. *Comput Linguist, 23*, 13-32

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. Levine, & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 127-149). Washington, DC: APA

Clark, H. H., & Carlson, T. B. (1982). Hearers and speech acts. *Language, 58,* 332-373.

Clark, H. H., & Krych, M.A. (2004). Speaking while monitoring addressees for understanding. *JML, 50,* 62–8

Clark, H. H., & Marshall, C. R. (1981). Definite reference and mutual knowledge. In A. K. Joshi, B. Webber, & I. Sag (Eds.), *Elements of discourse understanding* (pp. 111–222). Cambridge: Cambridge University Press.

Clark, H. & Schaefer, E. (1987). Collaborating on contributions to conversations. *Lang Cognitive Proc, 2*, 19-41.

Clark, H. & Schaefer, E. (1989). Contributing to discourse. *Cognitive Sci, 13*, 259-294.

Clark, H. H., Schreuder, R., & Buttrick, S. (1983). Common ground and the understanding of demonstrative reference. *JVLVB, 22*, 245-258.

Clark, H. H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition, 22,* 1-39.

Dell, G., & Brown, P. (1991). Mechanisms for listener-adaptation in language production: Limiting the role of the "model of the listener". In D. J. Napoli & J. A. Kegl (Eds.), *Bridges between psychology and linguistics: A Swarthmore Festschrift for Lila Gleitman* (pp. 111–222). Hillsdale, NJ: Erlbaum.

Ferreira, V., & Dell, G. (2000). Effect of ambiguity and lexical availability on syntactic ambiguity on syntactic and lexical production. *Cognitive Psychol, 40,* 296-340.

Gale, C., & Monk, A. F. (2000). Where am I looking? The accuracy of video-mediated gaze awareness. *Percept Psychophys, 62*, 586–595.

Hanna J. E., Tanenhaus, M. K. & Trueswell, J. C. (2003) The Effects of common ground and perspective on domains of referential interpretation. *JML, 49*, 43-61

Haywood, S. (2004). *Optimal design in language production.* Ph.D. Dissertation. University of Edinburgh.

Horton, W.S., & Gerrig, R. J. (2004 *in press a*). The impact of memory demands on audience design during language production. *Cognition*.

Horton, W.S., & Gerrig, R. J. (2004 *in press b*). Conversational common ground and memory processes in language production. *Discourse Processes*.

Horton W.S. & Keyser B. (1996) When do speaker take common ground? *Cognition, 59,* 91- 117

Keysar, B, Lin, S., Barr, D. J, (2003). Limits on theory of mind use in adults. *Cognition*, *89*, 25–41

Lockridge., C. B, & Brennan, S. E (2002). Addressees' needs influence speakers' early syntactic choices *Psych Bull & & Rev*, 9, 550-557

Metzing, C., & Brennan, S. E. (2003). When conceptual pacts are broken: Partner-specific effects in the comprehension of referring expressions. *JML, 49*, 201–213

Pickering, M., & Garrod, S. (2004). Towards a mechanistic psychology of dialogue. *Behav Brain Sci, 27*, 169-190.

Pomplun, M., Rieser, H., Ritter, H. & Velichkovsky, B. M. (1997). Augenbewegungen als kognitionswissenschaftlicher Forschungsgegenstand. In Kluwe, R.H. (Ed.), *Kognitionswissenschaft: Strukturen und Prozesse intelligenter Systeme*, 65-106. Wiesbaden: Deutscher Universitätsver.

Robust Semantic Interpretation and Dialog Management in the Context of a CALL Application

Johan Michel
Avenue Laënnec
72085 LE MANS CEDEX 9
johan.michel@univ-lemans.fr

Jérôme Lehuen
Avenue Laënnec
72085 LE MANS CEDEX 9
jerome.lehuen@univ-lemans.fr

Abstract

This paper describes a work about dialog managing in the context of a Computer Assisted Language Learning (CALL) research. In this paper, we choose to focus on the dialog management which is modeled in terms of tasks and methods. The following sections describe the semantic analyzer, the dialog model, and a few results.

1 Introduction

We will present a double model (semantic interpretation and dialogue management) for human-computer dialog in the context of a computer-assisted language learning (CALL) system. In this environment, the learner is implicated in an interaction with a virtual partner around a task (a recipe) to perform in a virtual micro-world (a virtual kitchen) [Lehuen 00], [Michel & Lehuen 02]. Related works exist, we can mention [Hamburger 94]. The following figure shows the user interface of our system:

2 The analyzer

In this specific context, the system will have to deal with incomplete or ungrammatical utterances. So our constraints are the following ones:

- Modularity: it takes place in an existing CALL architecture. More particularly, it is in connection with a virtual environment implemented as a micro-world;
- Robustness: it has to deal with odd utterances. Moreover, even if the learner uses correct words which are not in the lexicon, the interaction has to be continued;
- Non-determinism: it has to be able to produce partial or multiple interpretations for one utterance in order to carry on the interaction. The context of the interaction has to complete or select them.

We start from existing robust methods like “skimming parsing” [Dejong 82] and “chart parsing” [Winograd 83]. But the semantic analyzer we implemented is able to generate lexical hypothesis when unknown words impede its process [Michel & Lehuen 04]. Then, these hypotheses are used to engage a dialogic recovery strategy using the words recognized by the analyzer.

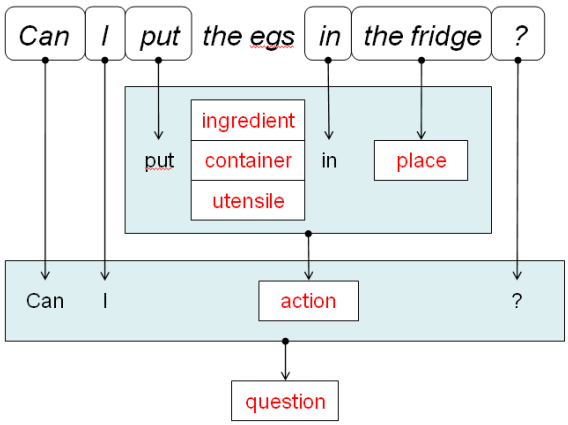


Fig. 1: Example of a syntax-driven hypothetico-deductive analysis

The figure 1 shows how the sentence *"Can I put the egs in the fridge?"* is analyzed as a question on the basis of one (triple) hypothesis about the unknown (and wrong) segment *"the egs"*. In this syntax-driven hypothetico-deductive analysis, *"the egs"* can be an ingredient, a container or a utensil. The analysis has three different steps: the lexical cover checking, the syntactic cover checking and the syntactic recovery. The lexical cover checking verifies if all the words in the utterance belong to the lexicon. The syntactic cover checking verifies if a syntactic pattern can be applied to the utterance. The syntactic recovery reorganizes the utterance to find a syntactic pattern. This recovery succeeds if after having reorganized the sentence, a syntactic pattern is found.

<table>
<tr><td rowspan="6">Good lexical cover</td><td>good syntactic cover</td><td colspan="2">« put the eggs in the fridge » (1)</td></tr>
<tr><td rowspan="3">bad syntactic cover</td><td>Syntactic recovery succeeds</td><td>« put in the fridge the eggs » (2)</td></tr>
<tr><td rowspan="2">Syntactic recovery does not succeed</td><td>« the fridge » (3)</td></tr>
<tr><td>« open » (3)</td></tr>
<tr><td rowspan="5">Bad lexical cover</td><td rowspan="2">good syntactic Cover</td><td>With hypothesis</td><td>« <u>foo</u> put the eggs in the fridge » (4)</td></tr>
<tr><td>Without Hypothesis</td><td>« put the <u>foo</u> in the fridge » (5)</td></tr>
<tr><td rowspan="3">bad syntactic cover</td><td>Syntactic recovery Succeeds</td><td>« put in the frige <u>foo</u> the eggs » (6)</td></tr>
<tr><td rowspan="2">Syntactic recovery does not succeed</td><td>« <u>foo</u> the eggs <u>foo</u> » (7)</td></tr>
<tr><td>« <u>foo</u> » (8)</td></tr>
</table>

Table 1. Examples of cases the analyser must handle

Different interaction strategies can be chosen given the analysis results. The first step is checking if the analysis's results correspond to the applicative (state of the applicative task) and interactive context, in this case, the analysis is validated. The second step is creating the partner's reaction from the applicative and interactive context.

3 The dialog model

The second part of our work focus on the knowledge of the dialog and the domain levels which are both modeled as tasks and methods. This approach is coming from research on generic mechanisms for problem-solving. It enables to rationalize

the behavior of the system and provides a framework to design an abstract, implementation-independent description of problem-solving process (fig. 2). In our case, the domain level is only a pretext to engage dialog situations and to make rise linguistic problems. So, the dialog level is weakly connected with the domain level: the repair strategies are more about language and less about the task going on.
Figure 3 contains some tasks and methods to engage a dialog. You can see four tasks, one decomposition method, one iteration method and two execution methods.

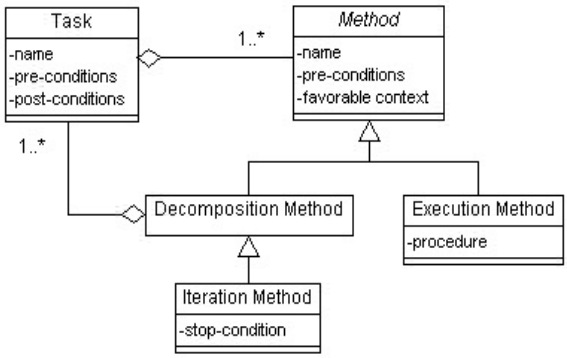


Fig.2: Task-Method framework as an UML class-diagram

.

```
(task (name     T-structure-dialog)
      (methods  M-structure-dialog))

(method-decomp (name   M-structure-dialog)
               (tasks  T-open-dialog T-dialog T-close-dialog))
```

```
(task (name     T-open-dialog)
      (methods  M-open-dialog))

(method-exec (name      M-open-dialog)
             (function  F-open-dialog))

(deffunction MAIN::F-open-dialog ()
   (assert (to-write "Hello, can you explain to me how to make a
                      chocolate cake?"))
```

```
(task (name     T-dialog)
      (methods  M-dialog))

(method-iter (name       M-dialog)
             (stop-cond  "(stop dialog)")
             (tasks      T-listen T-respond))
```

```
(task (name     T-listen)
      (methods  M-listen))

(method-exec (name      M-listen)
             (function  F-listen))

(deffunction MAIN::F-listen ()
   (assert (waiting-learner-utterance))
   (focus IHM))
```

Fig. 3: Some tasks and methods to engage a dialog

According to [Trichet & Tchnounikine 99], a Task-Method model can be decomposed into four layers, each layer manipulating data from the lower one. The highest layer, named control, deals with the way the modelized task is performed. It can be a simple algorithm, a classical expert system, or even a higher-level Task-Method engine (fig. 4). In our case, the control of the dialog level is performed by a simple independent algorithm, whereas the control of the domain level is performed by specific methods of the dialog level.

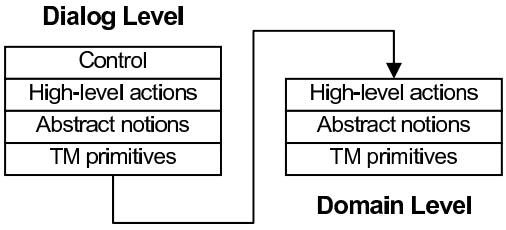


Fig.4: Task-Method model based on 4 layers

These models are fully implemented in Jess, a rule-based environment for the Java platforms. A first experiment has been conducted to test the system and its

robustness. In the interaction logs, we have observed the different awaited dialog situations.

4 Examples of dialogs

In the logs, we found the different interaction cases we expected. For a given group, on forty-six exchanges, we counted thirty-two exchanges linked to the task without any problems (Figure 5), fourteen exchanges with linguistic problems (Figure 6), and one exchange with a problem about the task (Figure 7). The exchanges are translated from French.

```
Learner> Take the milk
Partner> I take the milk from the fridge.
Learner> Take the cream
Partner> I take the cream from the fridge.
```

Fig. 5: Exchanges without problems

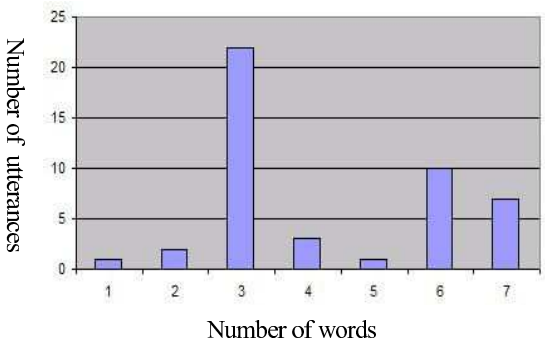
```
Learner> Open the cupboard at the top on
the righ
Partner> I do not understand when you write
« Open the cupboard at the top on the
righ». Which cupboard do you want to open?
Learner> Open the cupboard at the top on
the right
Partner> I open the cupboard.
```

Fig.6: Exchanges with linguistic problems

```
Learner> Take the milk
Partner> I understand, but it has already
been done
```

Fig.7: Exchange with a problem about the task

The learner's utterances were relatively short (from one to seven words) but this is mainly due to the nature of the task. The figure below illustrates this phenomenon for a given dyad:



References

[Dejong 82] G. Dejong 1982, *An overview of the FRUMP system*, In: "Strategies for natural language processing", Lehnert & Ringle, Erlbaum, Hillsdale, pp. 149-176.

[Delorme & Lehuen 03] F. Delorme and J. Lehuen 2003, *Dialog Planning and Domain Knowledge Modeled in terms of Tasks and Methods - A Flexible Framework for Dialog Managing*, In: ISMIS'03, Maebashi City (Japan), Springer-Verlag "Foundations of Intelligent Systems", Lecture Notes in Computer Science, pp. 689-693.

[Hamburger 94] H. Hamburger 1994, *Foreign language immersion: Science, practice, and a System*, In: JAIED Special Issue on Language Learning, vol. 5(4) pp. 429-453.

[Lehuen 00] J. Lehuen 2000, *A Dialogue-Based Architecture for Computer Aided Language Learning*, In: AAAI Fall Symposium on "Building Dialog Systems for Tutorial Applications", pp. 20-27.

[Lehuen & Luzzati 99] J. Lehuen and D. Luzzati 1999, *Acquisition coopérative d'une compétence langagière interprétative en dialogue homme-machine*, In: TALN'99, Cargèse (Corse), pp. 357-362.

[Michel & Lehuen 04] J. Michel and J. Lehuen 2004, *Un analyseur hypothético-déductif non déterministe pour l'apprentissage et la pratique d'une langue*, In: TAL & Apprentissage des Langues, Grenoble (France), pp. 13-22.

[Michel & Lehuen 02] J. Michel and J. Lehuen 2002, *Conception of a Language Learning Environment based on the Communicative and Actional Approaches*, In: ITS'2002, Biarritz (France), pp. 651-660.

[Trichet & Tchounikine 99] F. Trichet and P. Tchounikine 1999, *DSTM: a Framework to operationalise and Refine a Problem-Solving Method modeled in terms of Tasks and Methods*, In: International Journal of Expert Systems with Applications, vol. 16, pp.105-120.

[Winograd 83] T. Winograd 1983, *Language as a Cognitive Process*, Syntax, Addison-Wesley Publishing Compagny.

Extensions to Speaker/ Hearer Representation in DRT

Yafa Al-Raheb
University of East Anglia
y.al-raheb@uea.ac.uk

1 Introduction

Two DRT representations are introduced in Kamp et al. (2005). The first of which deals with presupposition and the second with propositional attitudes. However, neither representation deals with degrees of belief nor with speaker/hearer representation. The paper proposes a reconciliation and extension to these two DRT variants in order to represent degrees of belief and enhance the link between the linguistic content (utterance) and speaker/hearer representation, thus achieving an enriched account of presupposition.

2 Examples

The 'strength' of beliefs held by speakers differs from one situation to another, and depends on whether the speaker is introducing the topic of the dialogue. A weaker form of belief, called acceptance, is introduced to permit a form of differing degrees of beliefs. Acceptance represents the grey area where information is put on hold, not yet believed, but not rejected (Al-Raheb 2004). To explain what is meant by belief and acceptance, here is an example:

(1)

S1: I must buy Vincent's wife a birthday present.

H1:I didn't know Vincent was married.

S2: Yes, he is. His wife likes chocolate.

H2: She may also like flowers.

S3: I'll buy her chocolates.

The speaker, S, presents the presupposition (here it is new information to the hearer, H) that Vincent has a wife. Initiating the topic of presupposition allows H to attribute a stronger degree of belief to S about the presupposition. If we contrast example 1 with example 2 below, the stakes of the strength of beliefs would be much higher for H when he is required to perform an action than when simply going along with the dialogue.

(2)

S1: You should buy Vincent's wife a birthday present.

H1: I didn't know Vincent was married.

S2: Yes he is. His wife likes chocolate.

H2: She may also like flowers.

S3:But she prefers chocolate.

H3: I'll get her some chocolate.

In example 1, where H was not required to perform an action, it is safer for S to assume that H accepts the presupposition, as H is not committing to doing any task, than to assume the stronger case, i.e. H believes the presupposition. However, in example 2, where H agrees to buy Vincent's wife a present in H3, i.e. H commits to perform an action for Vincent's wife, S concludes that H believes the presupposition and adds this to S's representation of H's beliefs. These two examples show that certain pragmatic conditions can have a bearing on 'strength' of beliefs, which go beyond truth conditions. Our focus is on the compatible representation of 'strength' of be-

lief arising in presupposition.

3 Speaker/ Hearer Representation

This section explores the relationship between Kamp et al.'s two DRT variants, which can be linked and extended to provide appropriate representation for Speaker/ Hearer interaction. Kamp et al. (2005) discuss two separate variants of DRSs (Discourse Representation Structures) for beliefs and presupposition in DRT. The first only includes presuppositional and non-presuppositional aspects such as those shown in figure 1, to represent the linguistic content of an utterance without mention of cognitive states. Figure 1 shows Kamp's separation of presupposition and assertion for 'The rabbit is white' without dealing with beliefs. The first two nested DRSs show presuppositional information. The third nested DRS contains the non-presuppositional information, the assertion.

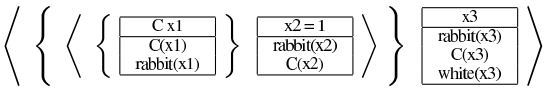


Figure 1: Linguistic Content DRS

The second DRS variant deals with beliefs, desires and intentions but not the presuppositional and non-presuppositional content. Figure 2 shows Kamp et al.'s (2005) formulation of someone who is shown a box full of stamps and told he can keep one. That person sees part of a stamp that he deems valuable, so he forms the belief, BEL, that there is a valuable stamp in the box, has the desire, DES, to possess the stamp, and the intention, INT, to pick the stamp from the box to fulfill his desire to own the stamp, 2d1840GB.

However Kamp et al. (2005) do not establish the link between these two DRT variants to explain the connection between speaker generation, speaker's utterance, and hearer recognition. To establish this link, a new

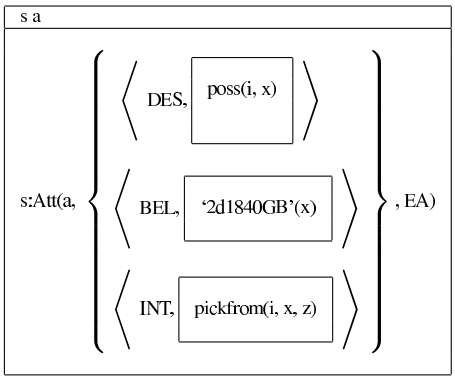


Figure 2: Propositional Attitudes DRS

DRS representing both the linguistic content and cognitive states of two participants is established. First of all, each DRS representing an agent's, (i.e. hearer's or speaker's) cognitive state includes the two personal reference markers 'i' and 'you'. When 'i' is used in a DRS, it refers to the agent's self within that entire DRS. To refer to the other agent, 'you' is used. To make the link between speaker generation, linguistic content, and hearer recognition more explicit, presuppositions are marked by a presupposition label 'p_n', 'n' indicating a number. The labels increase the expressive power from an essentially first-order formalism to a higher-order formalism. Assertions are marked by 'a_n'. Similarly, DRSs inside the main speaker or hearer DRS are labeled 'drs_n'.

The reconciled version of DRT employs both Kamp's intention and beliefs spaces. However, the belief space is expanded to include the speaker's beliefs about the hearer's beliefs. The beliefs of an agent give the motivation for making an utterance, and the recognition of an utterance gives the hearer an insight into the agent's beliefs.

Another space or DRS is introduced to represent weaker belief, or '`acceptance`' space. This also includes the speaker's acceptance space as well as what the speaker takes the hearer to accept. Provided the speaker has

sufficient information, the speaker can also have an embedded DRS within the acceptance space that represents what the hearer takes the speaker to accept. The same level of embedding is also introduced within the belief DRS when necessary.

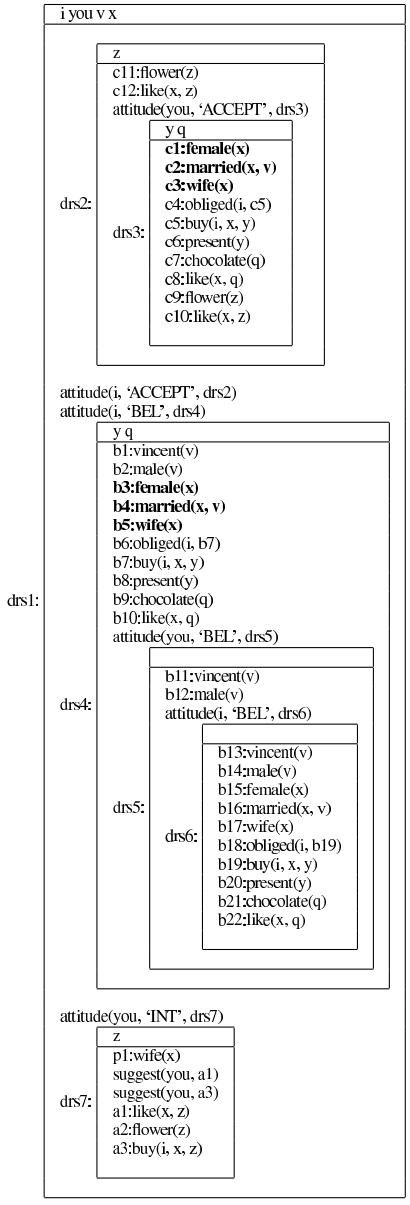


Figure 3: Speaker Recognition (After H2)

The intention space has been expanded to include the linguistic content provided by the current utterance, originally only represented in figure 1, to strengthen the link between an agent's intentions and the linguistic form uttered. The intention space also links the assertion with the presupposition that the particular assertion needs for linguistic realization and represents the dialogue act generated by making an assertion (Traum 1997). Believed information labelled 'b_n' inside a belief DRS or accepted information labelled 'c_n' inside an acceptance DRS can be either presupposed or asserted inside the intention DRS. As such, the labels in the intention DRS can only be 'p' or 'a'.

Intention space differs from the belief and acceptance spaces in that the intention space directly links to the current utterance being represented, whereas belief and acceptance spaces may include previous beliefs or accepted information. This gives the flexibility of being able to model information that the hearer has recognized but has not yet decided to accept or believe and, is therefore, not yet included in either the belief or acceptance space.

Figure 3 of speaker recognition in example 1 after H2 shows three embedded DRSs, acceptance DRS, drs2, belief DRS, drs4, and intention DRS, drs7. DRSs are referred to by the attitude describing them. For example, attitude(i,'BEL',drs4) refers to the DRS containing the speaker's beliefs, using the label for the belief DRS, drs4. The speaker's acceptance DRS, drs2, contains an embedded DRS for the hearer's acceptance space, drs3. Similarly, the belief space, drs4, contains space for the speaker's beliefs about the hearer's beliefs, drs5. The intention DRS, drs7, contains the recognized linguistic content of the utterance that the hearer made in H2.

Unlike example 2, the dialogue in example 1 has not provided the speaker with sufficient information to conclude that the hearer believes the assertions and presuppositions: the speaker has to buy a present, Vincent's wife likes chocolates, that the speaker will buy chocolates for Vincent's wife as a present, and

there is such a person as Vincent's wife. After H2, the hearer has just suggested flowers as a present for Vincent's wife, which, given that the speaker has reason to believe the hearer is cooperating, leads the speaker to assume the hearer accepts the presupposition, Vincent has a wife. The propositions are thus represented in the the hearer's acceptance space, drs3, rather than his belief space, drs5. However, as the speaker initiated the topic of the conversation and indeed the type of present that Vincent's wife may prefer, the hearer has stronger grounds to believe that the speaker believes her utterances, drs6.

Example 2, on the other hand, where the hearer first questions the presupposition, Vincent has a wife, but later on agrees to buy a present for her, shows greater strength of belief attached to the presupposition which affects that commitment. By virtue of that commitment, the speaker can attribute greater strength to the hearer's beliefs about the presuppositions and assertions, which are represented in the speaker's beliefs about the hearer's belief, drs5, in figure 4 .

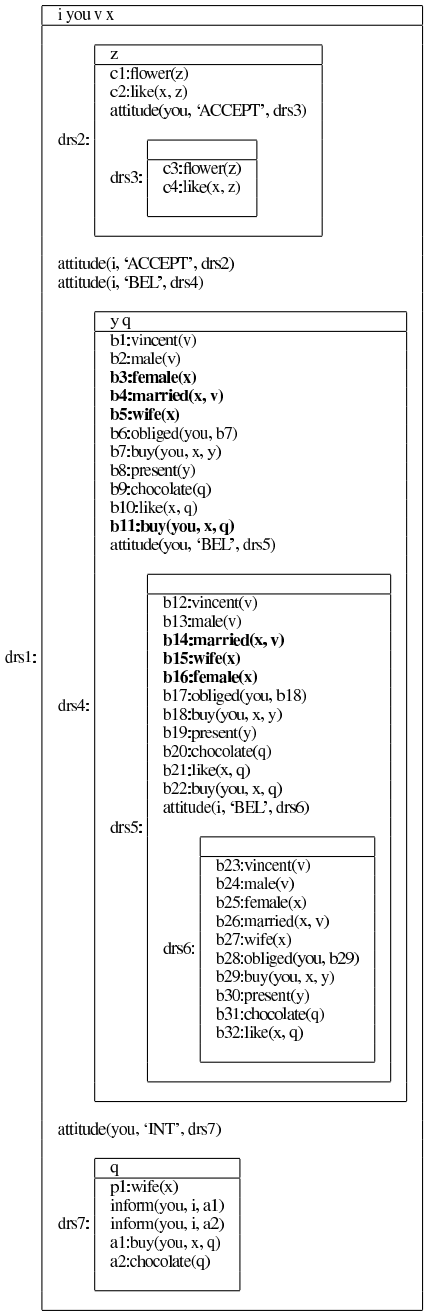


Figure 4: Speaker Recognition (After H3)

4 Conclusion

The paper attempted to represent the complex process of speakers recognizing utterances and using the linguistic information in forming mental representations of hearers' mental representations. This lead us to propose some modifications to DRT to offer compatible speaker/ hearer representations and handle examples where degrees of belief are needed.

References

Al-Raheb, Y. 2004. *Presupposition and Belief in DRT: Towards a New Implementation*. In: Ginzburg, J. and Vallduvi, E. Catalog 04: Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue, Barcelona, 144-145.

Kamp, H., van Genabith, J. and Reyle, U. 2005 *The Handbook of Logic*. Gabbay, D. and Guenthner, F. http://www.ims.uni-stuttgart.de/ hans/.

Traum, D. 1997. *Report on Multiparty Dialogue subgroup on Forward- looking Communicative Function*. Standards for Dialogue Coding in Natural Language Processing, Dagstuhl-Seminar Report no. 167.

WOZ experiments in Multimodal Dialogue Systems

Pilar Manchón Portillo
University of Seville
p.manchon@indisys.es

Guillermo Pérez García
University of Seville
g.perez@indisys.es

Gabriel de Amores Carredano
University of Seville
jgabriel@us.es

Abstract

This poster describes a new implementation of a multimodal dialogue system in the Home Machine Environment and the platform developed to conduct the set of experiments designed to model the system's behaviour in this scenario. The research carried out in this paper has been partially funded by EU Project Talk (Contract No 507802) and the Spanish Ministry of Science and Technology under Project TIC2002-00526.

1 Introduction

The objective of these experiments is to extend an existing spoken dialogue system integrating new input and output modalities. In order to achieve this goal, we have designed a WOZ platform where several experiments will be conducted. The experiments' design will be discussed and justified.

2 System Description

The original system is based on the Information State Update approach and has been especially designed to deal with Natural Command Languages. It consists of a number of OAA (Open Agent Architecture) agents which share information and perform different tasks according to a predefined overall dialogue strategy.

The dialogue history is recorded and taken into account in order to disambiguate subsequent utterances and achieve a more natural Human-Computer Interaction. The system can deal with multiple coordinated commands, spontaneous corrections, anaphoric reference resolution and several additional spoken-dialogue-related phenomena.

With regard to the chosen scenario, this particular system application can control several types of devices: lights, music, fan, dimmers, blinds, telephone (…).

The final objective is to integrate additional modalities in the system whereas at the same time allowing for greater flexibility, efficiency and naturalness in the overall interaction. This presents a great deal of additional complexity, since not only all modality-dependent issues have to be addressed independently but also the new issues arisen with the integration of modalities must be taken into account.

The system will deal with both graphical and spoken input, as well as a combination of the two:

- "Turn the lights on" (spoken input)
- Click (graphical input)
- "Turn **this** on" + Click (multimodal input)

3 Experiment Description

The objective of the experiments to be conducted is to record the interactions between human users and the wizard from different perspectives, in order to gather information to configure the basic system.

The experiments will take place in our labs and the special setting is described below.

Completely naïve subjects will provide reliable data about the first reaction of an untrained user before becoming familiar with the system. At the same time, as the subjects become more familiar with the system, we will learn about efficiency and learnability. The analysis will include among other issues:

- possible obstacles or difficulties to communicate
- biases that prevent the interactions from being completely natural
- corpus of natural language in the home domain
- modality preference in relation to task
- modality preference in relation to system familiarity
- task completion time
- combination of modalities for one particular task
- inter-modality timing
- multimodal multitasking

The subjects will initially be given just enough information to perform the tasks, but will not be given precise instructions as to how to proceed with the system. They will be given very general information such as "you may talk to the system", "you may select things by touching the screen" or "you may do both things at the same time". In subsequent phases, the subjects will be provided with more and more information as they become familiar with the system.

As far as the subjects are concerned, they will be interacting with an intelligent multimodal dialogue system and no other human will be involved. They will be provided with one task at the time that will appear on a computer screen. They will be alone in a room especially prepared for the experiment. There will be:

- a touch-screen
- a microphone
- speakers
- a camera
- several devices
- a list of tasks
- a general description of the situation

The interaction between subject and system will be recorded from all perspectives. The camera abovementioned will video record the experiment. Special software will be used to record the touch-screen activity and all agents in the experiment set-up will log all their actions

The wizard will be out of sight but will be able to hear what the subject says and see their touch-screen. Although the subject's input will also be processed and logged by a speech recognition engine, the wizard will pretend to understand everything (within a predefined set of guidelines), excepting a few artificially introduced recognition errors. In response to the subject's actions, the wizard will produce speech, display a written message or image, execute an action, or any combination of the former. When producing speech, the wizard will use synthetic speech or pre-recorded prompts.

4 Platform Description

4.1 Hardware:

a. Wizard computer
The wizard has two main roles: interaction with the user simulating the real system, and control of the physical devices.

b. User Tablet PC
The user will be requested to perform tasks within the home machine environment and this tablet PC will allow her to access the graphical display, speak or both.

c. WiFi router
The user's Tablet PC and the wizard computer will communicate by means of a WiFi router that will allow the user to move freely around the room.

d. Home devices
Our lab setting includes a number of lights and a blind connected to X10 modules. A security camera is simulated with a pre-recorded video. A telephone terminal is also simulated on the screen.

4.2 Software

4.2.1 Wizard Agents

a. Wizard Helper
This is basically a control panel that enables the wizard to:

- Talk to the user. The panel is connected to a TTS running on the user's computer. The wizard can either choose among a number of possible sentences (previously determined according to the possible actions of the user) or type an alternative answer if the user's behaviour differs from what had been foreseen.

- Remotely play audio and video files (to simulate the camera and telephone).

b. Device Manager
This agent connects the wizard computer with the physical devices and with the user's Home Setup. When the Wizard clicks on the "kitchen light – on" button, this agent sends an X10 message to the kitchen light and also updates the user's Home Setup.

4.2.2 User Agents

a. Home Setup
This is a modified version of the actual system agent that displays the current setting of the house and its devices. The user may use the mouse or pen to click on the devices. When the device is clicked, it blinks (so that the wizard can see it with his remote screen) and sends a log message to the Log Manager. The Home Setup is linked to the Device Manager, so as to ensure its immediate update.

b. Telephone Simulator
This is the telephone terminal access icon on which the user can click. It blinks when clicked on and sends a message to the Log Manager as the rest of the Home Setup devices.

c. ASR Manager
Although the wizard will just listen to the user and will not take into account the recognizer output, the ASR will be activated in parallel (word+word grammar) to provide additional data. We have implemented several wrappers for different commercial ASRs.

d. TTS Manager
This agent synthesizes the wizard text messages and allows the Log Manager to keep record of the utterance.

e. Log Manager
This agent keeps record of all the user-wizard interactions during the experiment. It includes the information sent by the GUI Agents (Home Setup and Telephone Simulator) and the voice Agents (TTS and ASR Manager)

f. Video Client
This specific agent is used to simulate the security camera.

4.3 Inter-agent communication.

The platform must obviously be distributed and suitable for real-time applications. Although several options were available (Corba, Darpa's Communicator and Stanford's OAA), our final choice was OAA.

4.4 Inter-agent synchronization.

One of the goals of our experiment is to determine how the user interacts with the system when he uses a mixed mode (e.g. two inputs at the same time: Voice: "switch on this light": Pen: Click on the kitchen lamp icon). In order to expand on Oviatt´s results on multimodal synchronization [1] [2] applied to our environment, a logging system with a precision of less than one second is needed. All our agents are implemented in C or JAVA, programming languages whose libraries allow millisecond precision, and the computers are configured running the NTP protocol.

4.5 Logging

This is the information saved in execution time during the experiment. The logging is therefore focused on low-level information, and especially on the time at which each utterance occurs.
The following table resumes the information logged:

Modality	Information Logged
GUI Input	Icon clicked Time
GUI Output	Message Time
Voice Input	General: Recognizer, Grammar, Language. Hypothesis level: Sentence, Score, Time Init, Time End. Word Level: Word, Score, Time Init, Time End.
Voice Output	Message Time

In order to save all this information, we have chosen the W3C recommendations from Emma (still a working draft), with very few modifications.

4.6 Annotation

This is information saved in post-execution time. Since our goal is mainly focused on the user's behaviour at dialogue level, much of the important information will be annotated.

The NXT toolkit is our first choice. We will develop our own display and use it to process the information.

5 Conclusion

This experimental platform will allow us to conduct the necessary experiments with the appropriate accuracy and simulation efficiency to ensure the robustness of the final results. In addition to this, different languages can be used and results compared in order to find some potential language-based differences in modality integration in multimodal dialogue systems.

References

[1] Oviatt, S. L., Multimodal interactive maps: Designing for human performance, Human-Computer Interaction, 1997, 93-129 (special issue on "Multimodal interfaces").

[2] Oviatt, S. L., DeAngeli, A. & Kuhn, K. Integration and synchronization of input modes during multimodal human-computer interaction. In Proceedings of Conference on Human Factors in Computing Systems: CHI '97.

[3] Hofs et ali. A multimodal interaction system for navigation. In Proceedings of Conference Diabrück 2003, 7th Workshop on the Semantics and Pragmatics of Dialogue: 2003

Micro-analysis of the belief transfer in information dialogues

Roser Morante and Harry Bunt
Tilburg University
The Netherlands
{r.morante|harry.bunt}@uvt.nl

Abstract

This paper describes work in progress, aimed at providing detailed empirical evidence about the processes of creating and updating information states in dialogue participants as the result of the utterances they exchange.

1 Introduction

Formal and computational work on dialogue modelling much of the time relies on the modelling of beliefs, goals, and intentions, following the 'BDI' paradigm that goes back to the work of Perrault, Allen and Cohen (Allen and Perrault, 1980; Cohen and Perrault, 1979; Perrault and Allen, 1980). More recently, this approach has taken a new form known as the 'information-state' or 'context-change approach', which uses the representation of agent's states of beliefs and other information in relation to the systematic (often plan-based) use of dialogue acts, as e.g. by Allen & Schubert (1994); Bunt (1996); Larsson & Traum (2000); Traum & Hinkelman (1992). With very few exceptions,[1] this more recent work does not involve truly formal modelling of information states, nor is it based on much empirical research into the details of how dialogue acts create and update the information states of dialogue participants. This paper describes some of our ongoing research that is aimed at providing an empirical basis for modelling the dynamics of dialogue agents' information states, by studying examples of recorded dialogue fragments under a formal microscope, trying to indicate for each dialogue utterance in detail which information it creates, strengthens, or cancels.

In doing this analysis, we include a first stage which is intended to be largely theory-neutral, by simply looking at the utterances, deciding exactly what they mean, and trying to make explicit what information they convey to the addressee. In a later stage we will perform another analysis of the same material using the system of dialogue acts defined in Dynamic Interpretation Theory (DIT) to annotate utterances and apply the definitions of the dialogue act types. We then compare the two analyses. Already at this stage it seems evident that the two analyses will *not* give the same results, since some of the dynamics of agents' information states is determined by global properties of stretches of dialogue, rather than purely locally by the effects of individual dialogue acts. This is an interesting first result. Second, the analysis is throwing new light on the phenomenon of grounding, which we believe can be analysed fruitfully by applying a formal notion of *mutual belief* and

[1] An exception is the work by Poesio & Traum, (1998).

investigating how mutual beliefs about weak beliefs may get strengthened to strong mutual beliefs.

In section 2 of this paper we briefly introduce DIT; in section 3 we describe the analysis method we applied, and we provide an example. Finally, in section 4 we summarize our initial findings and indicate directions for future research.

2 Information states in DIT

The background of our work is the theoretical framework of Dynamic Interpretation Theory (DIT; Bunt 2000), which gives a central position to the notions of *dialogue context* and *dialogue act*. A dialogue participant's beliefs about the domain and about the dialogue partner form a crucial part of his information state which in DIT is also called his *context*; there is no objective notion of dialogue context in DIT, but only the contexts (information states) of each participant. Dialogue acts are defined in DIT as semantic operations, used by dialogue participants to influence each other's context.

This means that dialogue acts are situated conceptually between utterances and context-changing operations; utterances are assumed to encode multiple dialogue acts, and their context-changing effects are defined through these dialogue acts. In this paper, however, we analyze the beliefs created and/or changed by dialogue utterances *directly*, without the intervention of dialogue acts. This has several potential advantages.

First, an analysis of context change based on dialogue acts is in danger of paying too little attention to changes that are not due to local effects of individual dialogue acts. We will see examples of that, relating to the phenomenon that grounding often occurs through implicit positive feedback. Second, by relating on the one hand context changes directly to utterances and in a later stage on the other hand assigning dialogue acts to utterances (on the basis of utterance and context features), we will obtain evidence on the validity and limitations of the modelling of a dialogue in terms of dialogue acts. Third, for the same reason, such an analysis can provide detailed insight into the semantics of dialogue acts whose meaning is not so easily defined in terms of changing beliefs, such as positive and negative feedback acts or time management acts. Fourth, and finally, the analysis of how the participants' information states change during a dialogue can help us to identify (sub-)types of dialogue acts that have not been noted before.

In this paper we focus on the analysis in terms of beliefs and goals. We analyze the processes involved in the creation and maintenance of the agent's beliefs about the partner's current beliefs and goals. The development of this analysis should make explicit how the flow of information in a dialogue affects the beliefs of both participants, in particular in relation to grounding and to persistence (and strength) of beliefs and goals. The data we analyze is a collection of information seeking dialogues, in which a user interacts with a simulated interactive help assistant for a fax machine.

3 Analysis method

For every utterance we represent several types of effects of the utterance on the cognitive state of the speaker and the hearer: effects of understanding, effects of expectations of being understood, and effects of processing the information which is being transferred. We represent the effects by means of some operators.

We define four types of beliefs (weak belief ($\|\cdot\cdot$), strong belief ($\|\vdash$), knowing the value of ($\vdash$), strong mutual belief ($\|\vdash^{*}$)), and a notion of goal ($\mid\!\sim$).

- Weak belief: a belief that an agent is not certain about and that requires confirma-

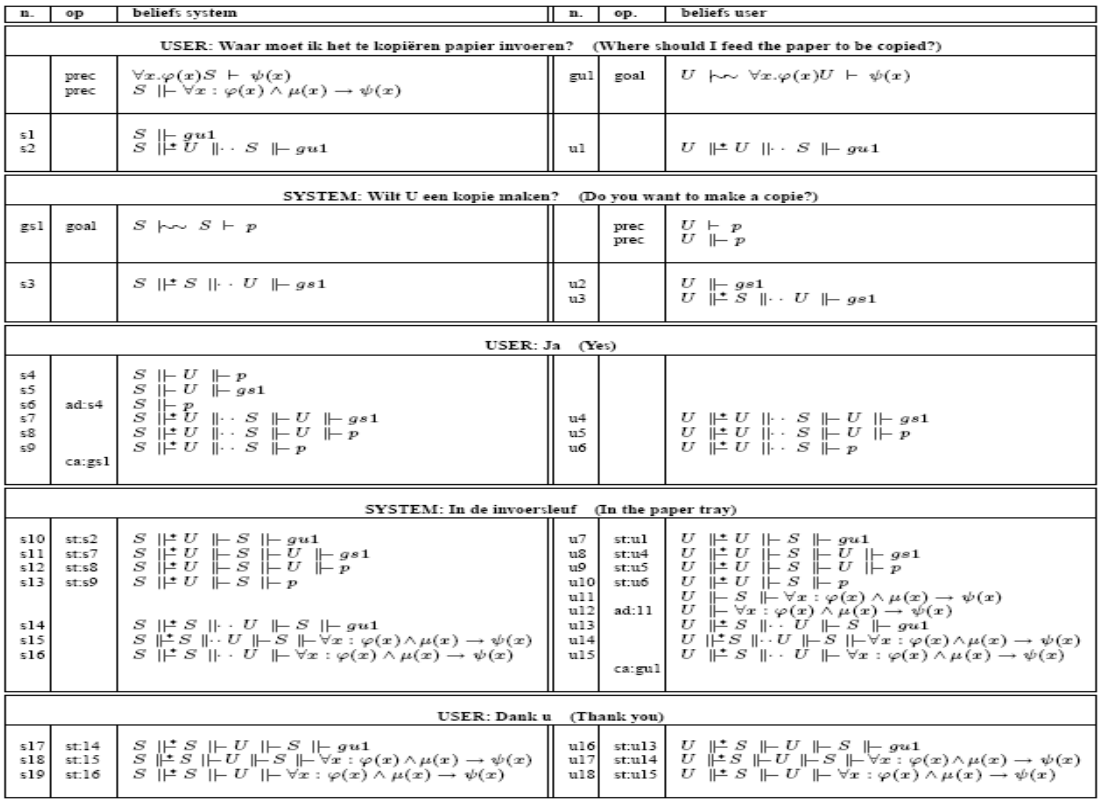


Figure 1: Simplified analysis of a dialogue fragment

tion to become strong belief. We assume that the hearer has a weak belief about the effects of his utterances, as long as he does not receive any feedback.

- Strong belief: a belief that the agent has no doubt about. We start from the assumption that the addressee of an utterance has no doubt about the appropriateness of the utterance. This is why the effect of an utterance in the addressee is represented by a strong belief.

- Knows value of: is formally defined as an abbreviation of a combination of strong beliefs.

- Mutual beliefs: these are the beliefs that both agents have about what is mutually believed (recursively).

We describe how beliefs change through operations on previous beliefs. As the dialogue evolves new beliefs are created and existing beliefs might change, or get cancelled. In order to model the changes we define the following operations that update beliefs and goals:

- Strengthening (st): A belief of S as an effect of U's utterance-1 will be strengthened when U emits another utterance related to utterance-1 that allows S to think that his belief was right. If there is negative evidence the belief is cancelled.

- Cancellation (ca): A belief is cancelled when it is disconfirmed or negated, or when new beliefs have been created that cancel its persistence. A belief can be cancelled if there is negative evidence about the belief. A goal to obtain information is cancelled when that information is provided.

- Adoption (ad): A belief is adopted when the agents incorporate it in their knowledge of the world. When adoption takes place it is often the case that a goal is accomplished, and thus cancelled.

We present a simplified example of analysis in Figure 1. Columns 1 to 3 contain the information related to the system's beliefs, and columns 4 to 6 contain the information related to the user's beliefs. In columns 1 and 4 the beliefs are numbered, in columns 2 and 5 the operations on beliefs are indicated, and in columns 3 and 6 the beliefs are formalized.

For every utterance we indicate the most important preconditions and goals, and the effects it causes in the hearer and speaker. Preconditions are conditions in the cognitive state of the speaker, that trigger or enable the emission of the utterance.

In this analysis grounding is interpreted as the coincidence of the same beliefs mutual beliefs in both participants.

4 Discussion and future work

The analysis shows that during a dialogue both participants for a while build up the state of beliefs about each other's beliefs and goals, as well as about the domain of discourse, and that after some time, when sufficiently much implicit or explicit positive feedback has occurred, the belief states become simpler and converge to a state when some common ground is established.

Interestingly this process is determined to some extent by nonlocal effects of sequences of utterances rather than by purely local effects of individual utterances.

Future research will involve an analysis that first, and independently, annotates all dialogue acts. Subsequently, the detailed local effects of dialogue acts, as predicted by DIT, will be compared with the present analysis.

The result will be feeded back into the theory and will be used to further investigate the interaction between dialogue acts and dialogue (belief) content.

References

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

James F. Allen and L. Schubert. 1994. The TRAINS project: A case study in defining a conversational planning agent. Technical Report TR 532, URCSD.

Harry Bunt. 1996. Interaction management functions and context representation requirements. In S. LuperFoy, A. Nijholt, and G. Veldhuizen van Zanten (eds.). *Dialogue Management in Natural Language Systems. Proc. of 11th Twente Workshop on Language Technology*. University of Twente, Enschede, pp. 187-198.

Harry Bunt. 2000. Dialogue pragmatics and context specification. In H. Bunt and W. Black, (eds.). *Abduction, Belief and Context in Dialogue*. John Benjamins, Amsterdam, pp. 81-150.

P.R. Cohen and C. R. Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science* 3: 177-212.

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

R.C. Perrault and J.F. Allen. 1980. A plan-based analysis of indirect of speech acts. *American Journal of Computational Linguistics* 6: 167-182.

M. Poesio and D. Traum. 1998. Towards an axiomatization of dialogue acts. *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogues (13th Twente Workshop on Language Technology)*, pp. 207-222.

D.R. Traum and E.A. Hinkelman. 1992. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence Special Issue: Computational Approaches to Non-Literal Language* 8(3): 575-599.

Multi-Party Interaction With Self-Contained Virtual Characters

Markus Löckelt
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
loeckelt@dfki.de

Norbert Pfleger
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
pfleger@dfki.de

Abstract

We describe a layered approach for coordinating interactions of human users and virtual characters in a multi-modal dialogue system.

1 Introduction

Contributions in face-to-face conversations convey not only propositional but also interactional content. Interactional information contributes to the structural organization of the conversation. It regulates the transitions between speaker and hearer, helps to avoid overlapping speech, and supports the identification of intended addressees of a contribution. We illustrate some aspects of multi-party discourse by an example of a quiz dialog. It includes a virtual moderator, a human user (Chris) and a virtual character (Frank):[1]

(1) *Moderator*: [⊙ both candidates] "The next question: Who scored the last goal at the world championship 1990?"

(2) *Chris*: [⊙ moderator] "Franz Beckenbauer"

(3) *Moderator*: [⊙ Chris; Frank shakes head and raises finger] "well, no . . . " [⊙ Frank]

(4) *Frank*: [⊙ Chris] "Oh dear, no" [⊙ moderator] "He was the coach." [Moderator nods] "The correct answer is Andreas Brehme."

(5) *Moderator*: [⊙ Frank] "Yes, that will be one point" [points at Frank] "for Frank!"

[1] ⊙ means "looks at".

Following (Duncan, 1972), conversations are organized in turns where participants coordinate their actions in order to achieve a smooth turn exchange. This takes place by means of a rule based signaling of what the individual participants want to do next. A hearer wanting to take the speaking turn can e. g. signal this by an upraised finger, sometimes accompanied by an audible intake of breath, see the beginning of turn (3). Even though there are several other ways to encourage a speaker to finish talking, a speaker who perceives these signals is able to infer the intention of the hearer and react accordingly (the moderator yielding the turn at the end of (3)). We model dialog exchange as being structured in rule-governed game-like sequences of dialog moves. When being addressed in a game move, a character has a choice of legal reply or followup moves, among which it selects one based on the current situation and its current goals. In turn (4), Frank determined that Chris has answered incorrectly. He decides to take over the pending response move; the moderator agrees by gazing at him.

2 Conversational Dialog Engines

Modules called Conversational Dialog Engines (CDEs) interact to realize the dialog capabilities of our system. All actions of a single virtual character are controlled by a dedicated CDE representing it. Human users of the system are also represented by their own CDEs, resulting in two classes of CDEs: CDEs creating the behavior of the virtual characters (*Character-CDEs*) and CDEs recognizing and analyzing the contributions of a hu-

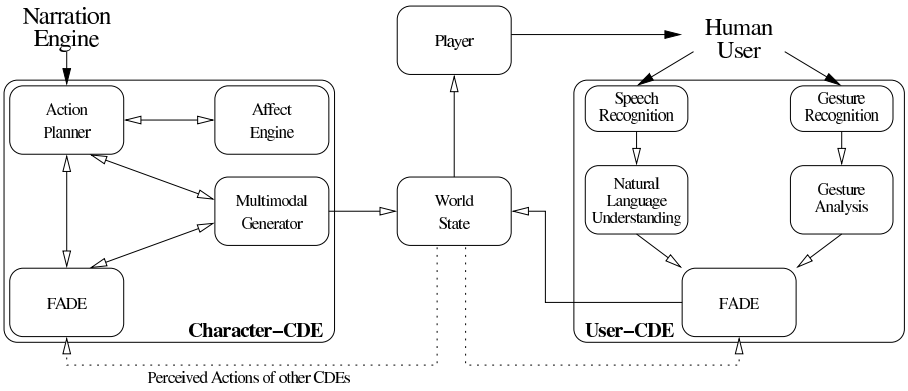


Figure 1: Components of Character-CDEs and User-CDEs

man user (*User-CDEs*).

Both CDE classes perceive, process and generate all character actions represented in the same ontology based data format. An abstraction of the actual state of the world can be perceived and manipulated by the CDEs. The abstract world state is interpreted by the 3D player to produce the actual visualization. A participant's contribution to an interaction is represented by an instance of a dialog act, e. g., *request*, and an embedded semantic representation of that utterance. Also, the internal knowledge of the virtual characters is represented in terms of this ontology. The Character-CDEs (as depicted on the left side of Fig. 1) consist of a fusion and discourse modeling engine (FADE), an affect engine, an action planner, and a multimodal generation component. In contrast, a User-CDE basically serves as a perception and translation module converting the actions of a human user into the ontology based representation the CDEs employ to communicate. A User-CDE comprises an ASR, a natural language understanding component, a gesture recognizer and analyzer, and a fusion and discourse modeling engine (FADE).

FADE The discourse modeling component of a CDE is responsible for interpreting the interactional contributions of the dialog participants and for maintaining a coherent discourse representation. It comprises a short-term local turn context based on a production rule system and a long-term, three-tiered discourse context representation. It models the flow of the interaction from the perspective of an individual dialog participant. Moreover, the interpretation of perceived events is based on the participant's current conversational role (e. g., speaker, addressee, overhearer).

The local turn context provides a comprehensive model of the current conversational situation. It models all participating co-interactors with respect to their current role and their respective internal states. This enables a CDE to interpret the perceived interactional contributions with respect to the current state of the conversation. One example would be a virtual character raising a finger into the visual field of another agent. This could mean either that the agent wants to take the turn (if its current role is that of an addressee, or overhearer) or that it wants to prevent another agent from taking the turn (if its current role is that of a speaker). The discourse history keeps track of the ongoing discourse and provides a comprehensive history of the individual discourse contributions. This enables the generation component to produce referring or elliptical expressions.

Action Planning The action planner is the deliberative unit for a character that devises the actions that are necessary to stepwise achieve its narrative goals. Each action planner operates as an independent agent whose deliberative process roughly follows a cycle where the narration engine indicates plot goals for one or more characters, the CDEs enact dialog moves to fulfill them, and report back success or failure. When additional processes are spawned, this can happen either directly consequential of the original goal (e. g. an obligation to answer a question) or as

a result of the internal state of characters, (e. g. complaining that questions are too difficult). Dialog management usually adopts one of several established approaches, with specific advantages and disadvantages. Common variants are based on planning and/or logical inference, finite-state machines, and forms, in order of decreasing representational power, flexibility, but also computational complexity (see (Larsson, 2002)). The suitability of an approach depends on the characteristics of the application. The interactions for a simple ticket-ordering application might map quite naturally to form-filling fixed data structures, but complexer scenarios call for more versatile interactions and representations. Our domain shows mixed characteristics, and we also use a combination of methods. Our scenario contains elements that have little variation and can be scripted (e. g. greetings), but the user interaction and autonomous behavior by the virtual characters also allow for flexible deviations interweaved into the story controlled by the narration engine. Both types of tasks share a common task model, the process, but the dialog games can be initiated using either a finite-state model, or a plan-based approach which is adapted from the system described in (Wahlster, 2003) to work with multi-party dialogs.

3 Three Levels of Processing

Purely Unconscious Behavior The lowest level of behavior comprises reactive actions of the characters. If, e. g., a character perceives another character has just started to speak, it should react by gazing at the speaker. Another example is *idle behavior* a character displays when there is nothing else to do (e. g., short intakes of breath or self-adaptors). Idle behavior can be willfully suppressed if participants in a conversation want to show inattentiveness they can refuse to gaze at the speaker, and is triggered by FADE or the Affect Engine. FADE monitors the perceived changes in the environment and ensures that the character displays proper behavior. The Affect Engine in turn controls the idle behavior and facial expressions of the characters with respect to emotional state (e. g. angry facial expression). The respective actions of a virtual character are triggered by interfaces to the multimodal generation component (see Fig. 1).

Semi-Conscious Behavior The semi-conscious behavior comprises actions that are hard to control, e. g. displaying the individual understanding of the current state of the turn-taking process or displaying backchannel feedback. This behavioral class demands for some reasoning and inference processes in order to display appropriate behavior. An addressee displaying backchannel feedback needs to know: (i) the exact location of a *transition relevance place* (TRP) (the point within a turn at which an addressee can take over or can display backchannel feedback; see (Sacks et al., 1974)) and (ii) the current status of the understanding process to determine the most appropriate response. The generation of backchannel feedback is triggered by FADE while the actual action is generated by the multimodal generation component. FADE needs to constantly monitor the perceived actions of the speaker and the other participants in order to determine the TRP in the speaker's turn. It also needs to monitor the current status of the natural language understanding.

Another instance of semi-conscious behavior is related to the process of requesting the turn as displayed by Frank in example turn (3). Here Frank knows the answer but the moderator is holding the turn at the moment (to get the turn Frank raises his index-finger). When the moderator notices, he yields the turn to Frank by stopping to speak and looking at him. On the technical side, this display of a turn requesting signal is managed and triggered by the multimodal generation component. First, the generator receives a request from the action planner to generate turn (4) but before it starts to generate and output this sentence it checks with FADE who is holding the speaking turn. If it is the character itself, the action planner's request can be realized directly. However, in this case, FADE informs the generator that the moderator is holding the turn. Based on the initial generation job and the current affective state, the generator selects appropriate actions.

Deliberative Behavior The top level of behavior control executes *processes* to achieve goals, which can be triggered externally, e. g. by a narrative control instance. Characters will also autonomously adopt goals to fulfill social obliga-

tions, e. g. conforming to a dialog game, or to honor internal (e. g. emotional) state. Deliberative behavior itself decomposes into three levels: Dialog acts, dialog games, and processes.

The lowest level comprises a set of *dialog acts*, the atomic communicative units between CDEs. We use a set of acts similar to those in (Poesio and Traum, 1998); examples are *opening* (greeting), *info-request* and *answer*. The propositional content of dialog acts refers to ontological object instances. The dialog acts themselves do not carry interlocutor obligations. *Dialog games* form the middle level. They specify exchanges of dialog act moves governed by rules, and the alternative moves legal in a situation. An *InformationSearch* game, for example, states that an initial *info-request* may allow for an *answer* making an assertion in response, a statement that one does not know the answer, or a refusal to answer. Dialog games can be combined by several operations, e. g. appended or nested, to form composite games (see e. g. (McBurney and Parsons, 2002)). Dialog game specifications need not be the same across characters (e. g. an unfriendly character need not know how to respond to an *opening*, and may ignore it). If a character participates in a game, it accepts the obligation to make only legal moves according to (its own version of) the rules of the game. The conventional part of the game definition—stating which moves are legal to make at any point of the game—is shared among all characters, and takes the form of a finite-state-automaton, where transitions are labeled with preconditions and postconditions. From the narration engine's point of view, a *process* appears as a parametrized black box. A *QuizQuestion* process, for example, would be parametrized by (i) instances of ontological objects filling *roles* specifying the moderator, the contestants, the subject, and possibly the presentation style, (ii) narrative constraints, e. g. a timeout, (iii) the content of the dialog history, (iv) the character's private world view, and (v) a set of *traits* for the character, which can be static (e. g., an intelligence value) or dynamic (e. g., the affective state). The process also needs a method of evaluating the appropriateness of answers. As stated before, the internal process implementation can use a finite-state representation for simpler tasks, or be plan-based if more flexibility is necessary. A process goal from the narration engine can result in several sub-processes for other participating characters, as in our example. The contestants are obliged to answer the moderator: The question in turn (1) is not directed towards a specific character. Any dialog participant can decide to join the game, and *Chris* does so first.

4 Conclusion

Our four-year project has passed its halfway point, for which we completed a demonstrator system implementing our first scenario. In the second project phase, more than one human user will be able to simultaneously participate in the dialog, using separated input devices.

Acknowledgements

This research is funded by the German Ministry of Research and Technology (BMBF) under grant 01 IMB 01A (VirtualHuman).

References

Starkey Duncan. 1972. Some Signals and Rules for Taking Speaking Turns in Conversations. *Journal of Personality and Social Psychology*, 23(2):283–292.

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Department of Linguistics, G¨oteborg University, Sweden.

Peter McBurney and Simon Parsons. 2002. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11(3):315–334, Summer.

Massimo Poesio and David Traum. 1998. Towards an Axiomatization of Dialogue Acts. In J. Hulstijn and A. Nijholt, editors, *Proceedings of TWENDIAL workshop*, Enschede.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. 1974. A Simplest Systematics for the Organization of Turn-Taking for Conversations. *Language*, 50(4):696 – 734.

Wolfgang Wahlster. 2003. Towards Symmetric Multimodality: Fusion and Fission of Speech, Gesture, and Facial Expression. In A. G¨unter, R. Kruse, and B. Neumann, editors, *Proceedings of the 26th German Conference on Artificial Intelligence*, pages 1 – 18. Springer.

A new Metric for the Evaluation of Dialog Act Classification*

Stephan Lesch and **Thomas Kleinbauer** and **Jan Alexandersson**
DFKI GmbH
Stuhlsatzenhausweg 3, D-66123 Saarbücken
{janal,kleiba,lesch}@dfki.de

Abstract

The standard evaluation metrics for dialog act classifiers are based on the boolean outcome of the exact classification. For multidimensional tag sets, such as the ICSI-MRDA tag set, this is stricter than necessary, since the misclassification might be partial and this can be good enough for the application in which the classifier is embedded. We propose a new forgiving metric and show some preliminary results. Some future work is sketched.

1 Introduction

We are concerned with the evaluation of automatic classification of utterances for multidimensional tag sets. Contrary to one-dimensional tag sets, such as the one developed within the VerbMobil project (Alexandersson et al., 1998), multidimensional tag sets assign not only one tag per utterance segment but a combination of a general tag and zero or more additional tags. This is the case for the ICSI meeting recorder dialog act tag set (henceforth MRDA), see (Shriberg et al., 2004).

When faced with a real-life application using speech, the task of assigning the correct tags can be further complicated through the absense of sentence boundaries. In addition to the dialog act labeling, the classifier might have to determine the segment boundaries, too, that constitute each utterance to be labeled (see (Ang et al., 2005)). Evaluation of such a task therefore needs to consider both the segmentation performance and the tagging results.

*The research presented here is funded by by the EU under the grant FP6-506811 (AMI).

For the pre-segmented case, the performance of the tagger is usually measured with *precision*, *recall*, e. g., (Reithinger and Klesen, 1997), and sometimes their harmonic mean, *fScore*. All three metrics are based on a notion of a "correct" classification which usually means that the tagger returned the correct label. This makes evaluation a binary function—the tagger output is either correct or incorrect.

For multidimensional tag sets the case is a bit more complex: each dimension in a label should be evaluated independently. For example, if the correct label is $\{t_1, t_2, t_3\}$ and the tagger assigns $\{t_3, t_4\}$, then dimension $\{t_3\}$ was classified correctly, dimensions $\{t_1, t_2\}$ were missed and $\{t_4\}$ was hallucinated. To compute the above measures within such a tag set, the size of the intersection between the assigned label and the actual label is divided by the size of the classified set in case of precision and the size of the correct set for recall ($\frac{1}{2}$ and $\frac{1}{3}$ in the above example). The *fScore* is still the harmonic mean between these two metrics (here $\frac{2}{5}$).

If we investigate the behaviour of the *fScore* metric, we see that whereas the value of a correctly assigned label is 1, and a completely erroneously assigned label is 0. Partly correct labels receive a different value depending on the size of the set of tags in the true tag. This is caused by the asymmetric behaviour of precision and recall. To highlight this, we use a small artifical tag set consisting of a general tag, T, and a set of additional tags $\{t_1, t_2, \ldots, t_6\}$ (see figure 1).

Table 1 shows the values for two fixed instances of the true label (first column). In the first case, the truth is $\{T, t_1\}$—written Tt_1— and in the second we have Tt_1t_3. The second row shows possible tagger output, alongside the *precision*, *recall* and *fScore* values for each result. We can observe an

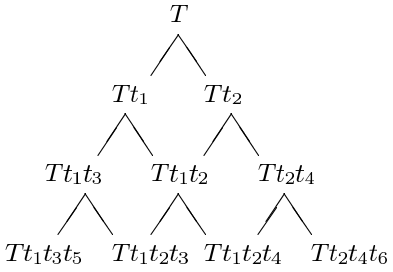


Figure 1: An excerpt of a made-up multidimensional tag set viewed as a lattice spanned by the subset relation. T is the general tag, and t_n are additional tags.

asymmetric behavior of *fScore* in rows 3 and 7. In both cases, the classified label contains one hallucinated special tag compared to the true label, but the *fScore* metric delivers different scores (0.8 and 0.86). A similar effect can be seen in rows 1 and 5, where in both cases the classified label misses one dimension in comparison to the ground truth while *fScore* yields values of 0.67 and 0.8.

Truth	Classified	*Prec*	*Rec*	*fScore*
Tt_1	T	1	0.5	0.67
Tt_1	Tt_1	1	1	1
Tt_1	Tt_1t_2	0.67	1	0.8
Tt_1t_3	T	1	0.33	0.5
Tt_1t_3	Tt_1	1	0.67	0.8
Tt_1t_3	Tt_1t_3	1	1	1
Tt_1t_3	$Tt_1t_2t_3$	0.75	1	0.86

Table 1: Values for *precision*, *recall* and *fScore* with different truth tags.

These effects occur because *fScore* takes the length of the true label into account (see also section 3): not only the absolute number of erroneously classified tags is relevant, but also the number of those that were classified *correctly*. In our example, row 3 yields two correct tags while row 7 has three—under this view, a higher *fScore* value in row 7 is justified. But it's also legitimate to ask for an evaluation metric that treats a deviation of one tag between classified label and truth equally, independent of

- whether the classified label contains one tag *too much* or *too little*.
- the length of the truth label, i.e. the position of this label in the hierarchy.

The rest of the paper is concerned with a new symmetric metric—Scorre— which addresses the above points. We compare the behavior of our new metric based on experiments on the ICSI meeting corpus. The paper is organized as follows: Section 2 discusses the hierarchical view of tag sets. We recapitulate the standard metrics precision, recall and *fScore* in section 3. Section 4 is devoted to our new metric. Before we conclude the paper and point at future directions, we present an experiment and compare our results in section 5.

2 Multidimensional Tag Set Hierarchies

Our MRDA taggers for the ICSI meeting corpus currently obtain around 50% correct classifications (i.e. the label produced by the tagger is identical to the human annotation). An examination of the result reveals that another 30% of the classifications are very similar to the human annotations.

Multidimensional labels can be regarded as sets of tags, and it is thus possible to compare them by looking at their intersection and the differences between them. Likewise, the labels can be organized into a hierarchy similar to figure 1. There, the number of edges between two labels, ancestor relations, in particular, whether two nodes have a common ancestor, play a crusial role. For a hierarchy on multidimensional labels defined by the subset relation between labels, there is an obvious equivalence to the set comparison.

In our approach, we use lattices as a more general structure to express other relations between tags not based on subset, and still use distances to measure similarity between labels.

In case of the MRDA tagset, there are labels which we regard as incompatible although they share some aspects. For instance, if the general tag is erroneously tagged, we want to consider the classification entirely wrong, even if the true and the classifier label share some additional tags.

Also, a metric based on distances can as well be used on one-dimensional labels which are ordered in a hierarchy. This is the case for the Verbmobil labels, which fall into several groups, such as, suggestions, feedbacks, informs, or politeness. Also, these group labels do not have to be actual DA labels, but can be introduced for the sole purpose of comparing more specific labels.

3 Classifier Evaluation

The performance of a classifier is usually measured with respect to two orthogonal aspects: the overall performance on a test corpus and the performance per tag. For both aspects, the common measures

recall, *precision* and *fScore* can be used. For the *per-tag* performance, three values have to be computed:

• *tagged*(*label*)—the number of times the label was assigned by the classifier,

• *occurs*(*label*)—the number of times the label occurs in the test corpus, and

• *correct*(*label*)—the number of times the label was correctly assigned by the classifier.

$$\begin{aligned} \mathit{Precision}(\mathit{label}) &:= \frac{\mathit{correct}(\mathit{label})}{\mathit{tagged}(\mathit{label})} \\ \mathit{Recall}(\mathit{label}) &:= \frac{\mathit{correct}(\mathit{label})}{\mathit{occurs}(\mathit{label})} \\ \mathit{fScore}(\mathit{label}) &:= \frac{2 * \mathit{Prec}(\mathit{label}) * \mathit{Recall}(\mathit{label})}{\mathit{Prec}(\mathit{label}) + \mathit{Recall}(\mathit{label})} \end{aligned}$$

To evaluate a classifier's overall performance on a test corpus, it is necessary to compute the overlap between the classified label (DA^C) and ground truth (DA^T) for each segment. In the case of multidimensional dialog acts, we regard each label as a set of tags, and thus define the intersection $DA^I := DA^T \cap DA^C$. Similar to the *per-label* case, *precision* and *recall* measure the amount of missed and hallucinated tags.

$$\mathit{Precision}(DA^T, DA^C) := \frac{|DA^I|}{|DA^C|} \quad (1)$$

$$\mathit{Recall}(DA^T, DA^C) := \frac{|DA^I|}{|DA^T|} \quad (2)$$

Next, we base our definition on the distance in the hierarchy and rewrite (1) and (2) using the subset relation: Let

$$\begin{aligned} \delta^C &:= |DA^C| - |DA^I| \\ \delta^T &:= |DA^T| - |DA^I| \end{aligned}$$

then

$$\mathit{Precision}(DA^T, DA^C) = 1 - \frac{\delta^C}{|DA^C|} \quad (3)$$

$$\mathit{Recall}(DA^T, DA^C) = 1 - \frac{\delta^T}{|DA^T|} \quad (4)$$

$$\begin{aligned} \mathit{fScore}(DA^T, DA^C) &= \\ & \frac{2 * \mathit{Prec}(DA^T, DA^C) * \mathit{Rec}(DA^T, DA^C)}{\mathit{Prec}(DA^T, DA^C) + \mathit{Rec}(DA^T, DA^C)} \quad (5) \\ &= \ldots \\ &= 1 - \frac{\delta^C + \delta^T}{|DA^C| + |DA^T|} \quad (6) \end{aligned}$$

Here, the reason for the asymmetrical behaviour of *recall*, *precision* and *fScore* is obvious: the denominators relate the distances to the total complexity of the labels, that is, the fraction of the total information missed by the classifier and how much information not present in the truth was hallucinated by the classifier respectively.

(3), (4) and (6) show that we can view *recall*, *precision* and *fScore* as distance metrics: tags missing in the classified label—δ^T—reduces *recall*, while tags hallucinated by the classifier—δ^C—reduces *precision*. *fScore* is a mixture of both distances.

4 A Hierarchy-Based Distance Metric

In a lattice of labels in which each pair of labels (DA^C, DA^T) has a least upper bound DA^{lub}, we define δ^T and δ^C using the shortest paths between the labels and DA^{lub}:

$$\begin{aligned} \delta^C &:= |\mathit{minpath}(DA^C, DA^{lub})| \\ \delta^T &:= |\mathit{minpath}(DA^T, DA^{lub})| \end{aligned}$$

For a lattice defined by the subset relation between tags (Y is a child of X iff Y contains all tags in X, and exactly one additional tag), DA^{lub} is equivalent to the intersection DA^I and the set-differences are equivalent to the distances between DA^T/ $DA^C$and DA^I.

We now define a metric with a constant denominator:

$$\textsc{Scorre}(DA^T, DA^C) := 1 - \frac{\delta^C + \delta^T}{2 * \mathit{depth}}$$

if DA^{lub} exists, 0 otherwise. The denominator is a constant, i.e., normalization is done with the distance between two labels into the range between 1 (DA^C=DA^T) and 0 (maximum distance between DA^C and DA^T, or no path at all).

Note, that *depth* must be large enough to prevent the metric from going below zero. One possible choice is the maximum possible path length (e.g. the maximum number of possible tags in a label). However, this number may be large, and in practice, a smaller value may be as appropriate, as long as no longer distances occur in a classification experiment.

Finally, we define Scorracy of a classifier on a test corpus with n segments, true labels DA_i^T and classified labels DA_i^C:

$$\textsc{Scorracy} := \frac{\sum_{i=1}^{n} \textsc{Scorre}(DA_i^T, DA_i^C)}{n}$$

Thus, Scorracy is the mean distance between the DA_i^T and DA_i^C normalized to the range between 1 and 0.

5 An experiment

When building a statistical tagger for MRDA labels, we have to choose between two basic approaches—one is to treat the labels as monolithic units (i.e. the roughly 118000 utterances in the ICSI corpus are annotated with ca. 1250 different labels), while the other is to decompose the labels into the 55 different tags, build one classifier for each tag (or for a group of mutually exclusive tags), and compose the results from these classifiers into labels.

Preliminary experiments indicate that the monolithic tagger performs better in terms of correct classifications (ca. 3%). For the combined tagger, however, the sum of exact + partial matches is slightly better. SCORRACY indicates that the mean distance between truth and classifier guess is nearly the same for both classifiers. (Clark and Popescu-Belis, 2004) reports a similar experiment with an abstraction of these labels (the MALTUS tagset), with similar results: they obtain 73.2% correct classifications with a simplified variant of the MALTUS tagset, and only 70.5% with a combined classifier.

In our experiments, we have used $depth = 5$, since labels deeper in the hierarchy did not occur. The advantage in this choice is that SCORRE is easier to interpret intuitively that way; for instance, 0.8 means that the distance between DA^T and DA^C is 2.

	monolithic		combined
	MALTUS	MRDA	MRDA
correct	67.1%	51.4%	48.5%
underspec.	11.2%	19.8%	25.8%
overspecific	2.7%	3.2%	2.9%
neighbours	2.1%	5.9%	4.1%
total	83.1%	80.3%	81.3%
precision	0.82	0.77	0.79
recall	0.77	0.68	0.67
fScore	0.78	0.70	0.70
total *fScore*	0.80	0.722	0.725
SCORRACY	0.81	0.76	0.77

Table 2: A single classifier for monolithic labels vs. a combination of classifiers for separate tags. Partial matches: underspecific classifications are e.g. `s^rt` classified as `s`; overspecific — `s` classified as `s^rt`; neighbours — `s^aa` classified as `s^bk`. *Precision*, *recall* and *fScore* are means over all classifications, total *fScore* is calculated from mean *precision*/*recall*

6 Conclusion and Future Work

We have presented a new metric for the evaluation of classifiers for multidimensional dialog act tag sets—SCORRE. We have shown that such tag sets can be arranged in a hierarchical manner and that the traditional metrics *precision*, *recall* and *fScore* can be understood as distance measures in this hierarchy. SCORRE is similar to *fScore*, but does not have its asymmetric property; SCORRE is independent on the position of the labels in the hierarchy.

Future work will include further experiments, in particular how adjustments in the classifier are reflected by the SCORRE values, in order to support optimization efforts for classification results.

References

Jan Alexandersson, Bianka Buschbeck-Wolf, Tsutomu Fujinami, Michael Kipp, Stephan Koch, Elisabeth Maier, Norbert Reithinger, Birte Schmitz, and Melanie Siegel. 1998. Dialogue Acts in VERBMOBIL-2 Second Edition. Technical report, DFKI Saarbrücken, Universität Stuttgart, TU Berlin, Universität des Saarlandes, July.

J. Ang, Y. Liu, and E. Shriberg. 2005. Automatic dialog act segmentation and classification in multiparty meetings. In *Proc. ICASSP 2005*, Philadelphia. To appear.

A. Clark and A. Popescu-Belis. 2004. Multi-level dialogue act tags. In *Proceedings of SIGDIAL '04 (5 th SIGDIAL Workshop on Discourse and Dialog)*, Cambridge, MA.

Norbert Reithinger and Martin Klesen. 1997. Dialogue Act Classification Using Language Models. In *Proceedings of EuroSpeech-97*, pages 2235–2238, Rhodes.

Elizabeth Shriberg, Raj Dhillon, Sonali Bhagat, Jeremy Ang, and Hannah Carvey. 2004. The icsi meeting recorder dialog act (mrda) corpus. In Michael Strube and Candy Sidner, editors, *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100, Cambridge, Massachusetts, USA, April 30 - May 1. Association for Computational Linguistics.

Automatic analysis of elliptic sentences in the Thetos system[1]

Nina Suszczańska
Institute of Informatics
Silesian Univ. of Technology
44-100 Gliwice, Poland
nsuszcz@polsl.pl

Julia Romaniuk
Institute of Linguistics NASU
Math.&Struct. Linguist. Dept.
01001 Kyiv Ukraine
rdmytro@i.com.ua

Przemysław Szmal
Institute of Informatics
Silesian Univ. of Technology
44-100 Gliwice, Poland
pszmal@polsl.pl

Abstract

The Thetos system translates Polish texts, both monologic and dialogic, into the Polish sign language. The system handles limited ellipsis cases in three main types, specific for parallel and non-parallel structures and for simple dialogues. A rich collection of Polish verbs with their valence schemes is used in this purpose. Our experiments suggest a possibility to reduce the simple-dialogue type ellipses to the remaining two ellipsis types. From another side, it is possible to adopt proposed methods of elliptic sentence processing to different languages.

1 Introduction

Thetos is an experimental system for translating written texts in Polish into the Polish sign language (Szmal and Suszczańska, 2001). It was primarily intended to be the sign language interpreter in the deaf's first contact with the doctor. Then we decided to charge it with interpreting fairy tales to deaf kids. (For this raison examples used in this paper are fragments of tale texts). Due to that dualism, in our research we have – among others – to practically solve problems connected with pronominal anaphora and elliptic structure both in dialogues and in monologic texts. In this paper our focus is the problem of automatic recognition of zero substitution and its reconstruction.

We distinguish three ellipsis types:

Anaphoric ellipsis appears in parallel structures of connected sentences, a complete and incomplete one, for example:

Najstarszy z braci otrzymał młyn, średni (e_1) (e_2) osła, a najmłodszy (e_1), Janek, (e_2) tylko kota. (The eldest of the brothers got the mill, the middle (e_1) (e_2) the donkey, and the youngest (e_1), Johnny, only (e_2) the cat.) (1)

Non-anaphoric (situational) ellipsis appears in non-parallel structures, for example:

Wraca kotek do domu – koguta nie ma (e_4). (The kitty returns home – there is no cock (e_4)). Note: in English, instead of adverbial, it is rather the predicate that would be dropped, giving in effect the sentence: *(e_4') No cock there.* (2)

Dialogic ellipses are specific for simple "question – answer" dialogues, e.g.:

— Kto tam (e_6)? — Słabym głosem zapytała chora babcia. (— Who (is) there? — The granny asked with a faint voice.)

— To ja (e_7) (e_8), kochana babciu, twoja wnuczka — odpowiedział wilk, udając głos Czerwonego Kapturka. (— It (is) me (e_8), dear granny, your granddaughter — the wolf answered imitating the Little Red Riding Hood's voice.) (3)

— Teraz masz własne mieszkanie? (Have you your own flat now?)

— Teraz (e_9) własne (e_{10}). (Now (e_9) own (e_{10})) (4)

In more sophisticated dialogues, all three types of elliptic sentences can be met. That is

[1] This work was supported in part by the Polish Committee for Scientific Research in 2003-2005 under Grant 4 T11C 024 24.

why we consider all of them.

Zero substitutions are governed by a set of rules, which we call "hidden grammar". Those rules allow for dropping components that are actually unessential, well then such ones, without which the whole sentence or its fragment stays fully comprehensible. They also say how to fill up sentences with dropped constructions (Romaniuk, 2001).

2 Ellipsis handling within translation process

In the Polish sign language both anaphoras and ellipses may appear, but rules of using them are a bit different than in phonic language. What we do translate now is a modeled text composed of sentences in so called canonical form (Suszczańska et al., 2004). To transform input sentences to this form we have – among others – to reconstruct the structure of full sentences on the basis of elliptic ones; indeed, it is translation within translation.

First steps involve automatic syntax analysis. The parser (Kulików et al., 2004) produces syntactic representation of the input sentence in the form of a labeled graph. Its nodes represent syntactic groups, and edges – syntactic relations occurring between them. During semantic analysis we transform the syntactic graph and get a predicate–argument structure. In this stage we reconstruct elliptic structures. For each ellipsis type we have to apply a specific algorithm.

Automatic ellipsis type classification is a problem for itself. Many complications for the analysis issue from the fact that the syntax of Polish allows for free sentence word order. It also complicates algorithms for reconstruction of elided components. This is why we haven't till now elaborated algorithms for finding constructions to supplement incomplete sentences but for some cases of ellipses only.

3 Ellipses in parallel structures

Parallel structures are well-known constructions that belong to the good writing-style canon; see e.g. (Nesbitt, 2002). It has been proven that formal structure of sentence where an anaphoric relation appears may be shortened only on condition that structures of connected complete and incomplete sentence are parallel (Gardent, 1993). Such reductions result in anaphoric ellipses, a specific kind of anaphoric connections. Each of them is a zero anaphora meant as a lexical zero.

Surface structure of a sentence mirrors its deep semantic structure. That's why the causes and possibilities of shortening sentences can be sought in their semantic structure; with that, one can refer to the communicative structure of sentences. While analyzing transitions between theme and rheme we may catch the content distributed in the whole text, not only in one sentence. In the case of anaphoric ellipsis of predicative center (PC) or PC's component, the rheme goes to peripheries of the structure of the content of text. E.g. predicate, which typically represents the rheme, in case of anaphoric ellipsis is known from the previous sentence.

Let's take a two-part compound sentence:

Point A lies on the line AB, and point B – (e_1) on the line CD.

In the second component sentence, the predicate (e_1) is in a peripheral position in relation to the rheme. Well now, at angle of the semantics of the sentence (of conveying new information in it), it is not important and – in consequence – it may be dropped.

Let's return to the problem of parallelism of complete and non-complete sentence. With that we will be considering both deep and surface structure. Due to frequently used rule of speaking effort economy, components which are on peripheries of the semantic structure of the sentence may be elided. Since a sentence should be understandable for the receiver even in case of being elliptic, then it should have a readable structure. It should repeat the structure of the previous (parallel) sentence. (Actually, there may be in a sentence more zeroes than PCs; we set up a hypothesis that with anaphoric connection, when structures are

parallel, all zeroes may by reconstructed.)

For the case of parallel structures, J. Romaniuk identified some rules for sentence abbreviation in Polish. They say that: 1^o PC or a component which is peripheral in relation to PC may be elided if the structures of the complete and the shortened sentences are parallel, because it is possible to rebuild the structure on the basis of context. 2^o In parallel structures, a sentence component that is dependent on the predicate is elided by stylistic reasons; if the missing component is signaled, it is reconstructed in effect of analysis of non-filled obligatory valence places.

These rules determine the way how to shorten a sentence and to leave it comprehensible in its context as well. On their basis we proposed an algorithm for reconstruction of structural and then lexical composition of elliptic sentences. It is only intended to analyze parallel structures of an incomplete sentence connected with a complete one.

It is easy to recognize two parallel structures in case where syntactic analysis gives an unambiguous parse of both sentences, from which the current one is elliptic, and the preceding one is not. Problems arise in case of ambiguous analysis.

We try to detect parallel structures: 1^o via searching for a dash „–" in the sentence (in such cases as shown in the example, the dash signals the position of ellipsis), 2^o via analyzing the morpho-syntactical traits of words as well as word order in the sentence, in that in parallel sentences the word order is preserved.

At the analysis of the deep structure of the sentence, we assume that in the valence scheme of the verb all obligatory places should be filled, and pretenders to an empty place should be searched in previous sentences. Trying to reconstruct the ellipsis, we limit the scope of searching for anaphora and antecedent by assuming that the anaphora is a PC or a component of PC, where PC should be meant as either subject, or predicate, or subject and predicate.

A similar algorithm works in a more general case, where lacking predicate has been detected in the sentence and the structure of the sentence has been established with using an adequate heuristics. An algorithm applicable for the deep sentence structure instead of the syntactic one is quite similar, too.

4 Ellipses in non-parallel structures

Non-parallel elliptic structures contain information about dropped components in the structure of the sentence and not in the context. Hence the context is useless for their repair. The most often dropped element is predicate that denotes a generalized movement or action. To resolve this type of ellipsis we add a verb (may be synthetic) of such kind to the structure of the sentence. The surrounding scheme for such verb should be fulfilled by any concrete verb of movement. For now we assumed a working variant of the verb and the scheme. It is an urgent task to examine all schemes of movement and action verbs in order to establish the set of common schemes and to define the desired generalizing one.

Evidently, the adopted solution is only the first of many possible steps in solving the problem. For example, there can be more than one dropped element, the verb can have a different meaning, etc.

In case of non-transitive verbs one can assume that the structure of elliptic sentences may be reduced to two subtypes:

$$subject - 0_{predicate} - adverbial$$

$$adverbial - 0_{subject} - 0_{predicate} - adverbial$$

Having inserted a generalized verb, we can try to reconstruct the dropped subject by using a generalized scheme. Obviously, in this case the analysis becomes unambiguous and imprecise.

5 Ellipses in dialogic texts

As it was mentioned above, in extended dialogues we can meet sentences that contain ellipses of both types discussed in the two preceding sections. Besides that, anaphoric

sentences with pronouns and other words intended to replace some elements are used. Our approach to analyzing anaphoric sentences and a method for searching antecedents was discussed in (Kulików et al., 2004).

In dialogues of „question – answer" type, the structure of both sentences is as a rule incomplete. For example:

— *Czy umiesz migać?* (*Can (you) sign?*)
// $0_{subject}$ – predicate *(5.0)*

— *Tak. (Yes.)* // $0_{predicate}$ – $0_{psubject}$ – adverbial *(5.1)*

or other answer variants:

— *Teraz tak. (Now yes.)*
// adverbial – $0_{predicate}$ – $0_{subject}$ *(5.2)*

— *Umiem. ((I) can)* // $0_{subject}$ – predicate *(5.3)*

— *Już umiem. ((I) already can)*
// adverbial – $0_{subject}$ – predicate *(5.4)*

In the sentence (5.0) subject can be easily rebuilt due to the grammatical form of the predicate, which obligatorily requires the subject *"ty" (you)*. The problem consists in detecting the type of shortening, and then – the corresponding reconstruction procedure.

The statement (5.1) is subject to anaphoric sentence analysis with substitutional word *"tak" (yes)*, whose antecedent is the preceding sentence as a total. That means that the structure of the sentence to be reconstructed, (5.1), will be entirely taken from the sentence (5.0), after its completion. There remains a problem with changing the subject expressed with the personal pronoun *"ty" (you)* into the pronoun *"ja" (I)*. The problem no more consists in mechanical change of the form of words, but in preserving both the formal representation of the content of the two utterances and the information that they all are concerned with the same person. In this case, for implementation purposes, we proposed to make use of the pronoun *"ten" (this)*. In consequence, our dialogue takes the following internal form:

— *Czy ten umie migać?* (*Can this sign?*) *(5.0')*

— *Tak. (Yes.)* ⇒ *Ten umie migać. (This can sign.)* *(5.1')*

In this point a new problem arises: transform the input sentence to a form which could be called a standard one.

So that, a possibility appears to reduce the third type of elliptic sentences to the precedent two. This our hypothesis requires additional research. It seems that analysis of the communicative structure of sentence could be helpful in this case.

6 Conclusion

There was no enough place to give a detailed description of algorithms discussed in this paper. The reader can find some additional information in (Kulików et al., 2004; Suszczańska et al., 2004). We have elaborated and implemented a part of exposed ideas. Experiments done in our Thetos translation system seem encouraging. We are intensively working upon accomplishment of remaining thoughts, since we find it necessary for the system to work satisfactorily.

References

Claire Gardent. 1993. A unification-based approach to multiple VP Ellipsis resolution. In: *Proc. of the 6th European Meeting of the ACL, Utrecht, The Netherlands.* Web page ftp://ftp.coli.uni-sb.de/pub/people/claire/multiplevpe.ps

Sławomir Kulików, Julia Romaniuk, and Nina Suszczańska, A syntactical analysis of anaphora in the Polsyn parser. In: *Proc. of the International Conference IIS:IIPWM'04*, Zakopane, Poland, 444–448

Scott Nesbitt. 2002. Parallelism. Web page http://www.unlv.edu/Writing_Center/Parallelism.htm, ed. A. Comeford, updated on 02 June 2002; accessed on 10 March 2005

Julia Romaniuk. 2001. Hidden Grammar of Anaphoric Ellipse. In: *Naukova spadshchyna prof. Semchynskogo i suchasna filologia.* Kyiv, 2:319-325 (in Ukrainian)

Nina Suszczańska, Przemysław Szmal, and Sławomir Kulików. 2004. Continuous Text Translation using Text Modeling in the Thetos System. *Int. J. of Computational Intelligence*, 1(4):338-341. Web page http://www.ijci.org/volumes/1304-2386-1.pdf

Przemysław Szmal and Nina Suszczańska. 2001. Selected Problems of Translation from the Polish Written Language to the Sign Language. *Archiwum Informatyki Teoretycznej i Stosowanej*, 13(1):37-51

Simplified MMAXQL: An Intuitive Query Language for Corpora with Annotations on Multiple Levels

Christoph Müller
EML Research gGmbH
Villa Bosch
Schloß-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
christoph.mueller@eml-research.de

1 Introduction

Growing interest in richly annotated corpora is a driving force for the development of annotation tools that can handle multiple levels of annotation.[1] Specialized query languages are employed for the exploitation of these corpora.

In order to make full use of the potential of multi-level annotation it is crucial that individual annotation levels be treated as *self-contained modules* which are independent of other annotation levels. This should also include the storing of each level in a separate file. If this principle is not observed, annotation data management (incl. level addition, removal and replacement, but also conversion into and from other formats) is made more difficult than necessary. Moreover, *multi-level querying* will be facilitated if annotation levels are independent of each other, because users can relate markables from all levels in a fairly unrestricted way, without having to consider representational issues that are irrelevant for their current query. This facilitates exploratory data analysis of annotated corpora for all users, including non-experts.

In our multi-level annotation tool MMAX2[2] (Müller & Strube, 2003) markable levels are independent of each other. The query language MMAXQL is rather complicated and not suitable for naive users. We present an alternative query method consisting of a more intuitive query language and an implemented method to generate MMAXQL queries from the former. The new, *simplified* MMAXQL can express a wide range of queries in a simple and compact way, including queries for discourse-level phenomena like coreference.

2 Simplified MMAXQL

A query in simplified MMAXQL consists of a sequence of *query tokens* which describe elements (i.e. either words or markables) to be matched, and *relation operators* which specify which relation should hold between the elements matched by two adjacent query tokens. Relations that can be queried include sequential, hierarchical, and associative relations. The **sequential** relation between two elements can be queried by means of the operator `before` (A ends before B begins) and `meets` (A ends when B begins)[3]. The **hierarchical** relation between two elements can be queried by means of the operator `in` (A is completely included/embedded in B) and `dom` (A completely contains/dominates B). The operators `starts` and `ends` combine the sequential and hierarchical relations, with `starts` standing for left alignment (A starts

[1] This description is based on Müller (2005).

[2] The current release version of MMAX2 can be downloaded at http://mmax.eml-research.de.

[3] This is the default operator.

when B starts and ends before B ends) and `ends` standing for right alignment (A starts after B starts and ends when B ends). In addition, **associative** relations like set membership can be queried by means of the relation operator `nextpeer` (cf. below for an example).

A query for **words** consists of regular expressions in single quotes. Each expression matches one word exactly. The query[4] `'[Yy]ou know'`, e.g. returns 59 hits in the form of 2-tuples, taking about 3 seconds to search the approx. 13.000 words of the document. *You know* is interesting in spoken dialogue because it can either have its literal meaning or be a lexicalised filled pause.

A query token for a **markable** is of the form `regExp/conditions`, where the (optional) `regExp` part specifies the text of a markable, and the `conditions` part defines matching conditions with respect to markable attributes and values. In the minimal form, a condition only specifies the name of a markable level. The sample document contains, among others, a *segment* level with 1398 markables roughly equivalent to speaker turns, and a *meta* level containing 1031 markables representing e.g. pauses, emphases, or sounds like breathing or mike noise. The query `/segment` retrieves a list of 1398 1-tuples in about 2 seconds. The query `.*pars.*/segment` returns the 3 segments which contain the string *pars* in about the same time.[5]

A more common way of query is in terms of attribute-value combinations. Simplified MMAXQL contains (optional) features which make this considerably easier. First of all, if the attribute name is unique across all markable levels, the level name can be left out, since the attribute name unambiguously points to it. Thus, a query like `/type=emphasis` can query markables from the `meta` level, granted that only one attribute of name `type` exists.[6] Furthermore, if an attribute is defined as having a closed set of possible values (as is the case for the *type* attribute on the *meta* level), and if the required value is unique across all values of all other attributes on all other levels, the attribute name can be left out as well. Thus, the above query can be reduced to `/emphasis`, which is shorter and more intuitive since what the user wants is finding cases of emphasis rather than particular attribute-value combinations. On the sample document, the query returns 265 hits in about 4 seconds.

Elements from several levels can be mapped to each other by joining query tokens using relation operators. The result of such a query is a tuple with as many columns as the query contained query tokens. In the following example, the *meta* and *segment* levels and the word level are combined in a query to retrieve instances of *you know* that appear in segments spoken by female speakers[7] which also contain a pause or an emphasis.

```
'[Yy]ou know' in (/participant={f.*} dom /{pause,emphasis})
```

The following equivalent but much more verbose and complicated MMAXQL query is automatically generated from the above:

```
let $10=segment (*participant={f.*});
let $11=meta (type={pause,emphasis});
let $22=contains($10, $11);
let $20=basedata (*basedata_text={[Yy]ou});
let $21=basedata (*basedata_text={know});
let $2=during(meets($20, $21), $22);
display $2;
```

Our ICSI corpus does not yet contain coreference annotation, but that is in the process

[4]Unless noted otherwise, all examples come from document BDB001 of the ICSI Meeting Corpus, a corpus of spoken multi-party dialogue (Janin et al., 2003). The corpus was obtained from the Linguistic Data Consortium and completely converted into MMAX2 format, preserving all original information. Reported query times were on a Pentium Mobile III/800 with 512 MB RAM.

[5]The `.*` wild cards in the latter query are required since by default a query matches whole markables only.

[6]If this condition does not hold, the attribute name can be disambiguated by prepending the level name.

[7]The first letter of the `participant` value encodes the speaker's gender.

of being added. Therefore, the following example is taken from a different corpus which consists of a part of the Penn Treebank portion of the Switchboard corpus. This corpus was converted into MMAX2 format and subsequently annotated for coreference, using a markable level with name `coref` and a markable set attribute with name `member`. On this corpus, the following query can be used to retrieve pairs of anaphors (right) and their direct antecedents (left).

```
/coref nextpeer:member /coref
```

On sample document 0013_4617 from our corpus, the query returns 51 2-tuples in about 2 seconds. The `coref` level has a `npform` attribute describing the morphological form of the expression. Thus, the query can be modified as follows to return only anaphoric *pronouns* and their direct antecedents.

```
/coref nextpeer:member /coref.npform=prp
```

This reduces the number of hits on the sample document to 32.

The corpus also contains coreference annotation for non-NP antecedents, i.e. statements or propositions that are referred back to by means of pronouns (mostly by means of *that*). Non-NP antecedents are tagged with the value `utt` (for utterances) and `vp` (for verb phrases) in the `expressionstype` attribute. Thus, pairs of anaphors and their direct non-NP antecedents (either utterances OR verb phrases) can be retrieved with the following query.

```
/coref.expressionstype={utt,vp} nextpeer:member /coref
```

This query retrieves 10 2-tuples.

A single query can contain more than 2 query tuples. The following query retrieves 3-tuples of the initial markable in a coreference set and the next two mentions by just concatenating three query tuples.

```
/coref.member=initial nextpeer:member /coref nextpeer:member /coref
```

The corresponding MMAXQL query runs as follows:

```
let $10=coref (member={initial});
let $11=coref;
let $12=coref;
let $1=next_peer('member',
       next_peer('member',$10,$11),$12);
display  $1;
```

3 Future Work

The current experimental implementation does not yet include wild cards, which is particularly inconvenient for queries using the `nextpeer` operator, because without a wild card querying a chain of `n` markables requires that many literal repetitions of the query token. Thus, future work includes adding support for wild cards on the query token level. The query language also still lacks a means to express queries like '*coref* markables that are `n` *coref* markables apart.' Finally, we are also looking into ways of further optimizing query execution.

Acknowledgements

This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany.

References

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

Müller, Christoph (2005). A flexible stand-off data model with query language for multi-level annotation. In *Proceedings of the Interactive Posters/Demonstrations session at the 43rd Annual Meeting of the Association for Computational Linguistics,* Ann Arbor, Mi., 25-30 June 2005. To appear.

Müller, Christoph & Michael Strube (2003). Multi-level annotation in MMAX. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue,* Sapporo, Japan, 4-5 July 2003, pp. 198–207.

Presentation Strategies for Flexible Multimodal Interaction with a Music Player

Ivana Kruijff-Korbayová, Nate Blaylock, Ciprian Gerstenberger, Verena Rieser
Department of Computational Linguistics
Saarland University
Saarbrücken, Germany
korbay@coli.uni-sb.de

Tilman Becker, Michael Kaißer, Peter Poller, Jan Schehl
DFKI
Saarbrücken, Germany
tilman.becker@dfki.de

Abstract

We present an ongoing project building a multimodal dialogue system for a music player supporting natural, flexible interaction and collaborative behavior. Since the system functionalities include searching a big MP3 database, multimodal output is needed.

1 Introduction

In the larger context of the TALK project[1] we are developing a multimodal dialogue system for a music player application for in-car and in-home use. The system functionalities include playback control, manipulation of playlists, and searching a large MP3 database. We aim at a system that will engage in natural, flexible interaction and collaborative behavior. We believe that in order to achieve this, the system needs to provide advanced adaptive multimodal output.

To determine the interaction strategies and range of linguistic behavior that humans naturally use in the music player scenario, we have conducted Wizard-of-Oz experiments. Our goal was not only to collect data on how potential users interact with such a system, but also (and importantly) to observe what range of interaction strategies humans naturally use and how efficient they are. We therefore used a setup where the wizard had freedom of choice w.r.t. their response and its realization in single or multiple modalities.

When developing our system, we design the multimodal output presentation strategies and the range of linguistic realization options based on experience gathered during the experiment and an analysis of the corpus.

We briefly describe our experiments and the collected data (Section 2), present initial observations on the presentation of database search results in speech and on screen (Section 3), and sketch the main system components involved output generation (Section 4).

2 SAMMIE Data Collection

We conducted two series of data-collection experiments: SAMMIE-1 involved only spoken interaction, SAMMIE-2 was multimodal, with speech and screen input and output.[2].

In both experiments, the users performed several tasks, such as finding a song or an album and playing it or adding it to a playlist. In some tasks, the users were given rather concrete specifications, such as a name (e.g, *Play Crazy by Aerosmith*), in other tasks they got more vague characteristics, such as period,

[1]TALK (Talk and Look: Tools for Ambient Linguistic Knowledge; http://www.talk-project.org), funded by the EU 6th Framework Program, project No. IST-507802.

[2]SAMMIE stands for Saarbrücken Multimodal MP3 Player Interaction Experiment.

genre or type of music (e.g., *Play a pop song from 2004*, or *Make a playlist with 4 of your favorite songs*). This resulted in interactions where the users were exploring the database contents and adding search criteria depending on what was found.

In SAMMIE-1, there were 24 subjects, who each participated in one session with one of two wizards. Each subject worked on eight tasks, for maximally 30 minutes in total. Tasks were of three types: finding a specified title, selecting a title satisfying certain constraints and building a playlist satisfying certain constraints.

In SAMMIE-2, there were 24 subjects, who each participated in one session with one of six wizards. Each subject worked on two times two tasks.[3] The duration was restricted to twice 15 minutes. Tasks were of two types: searching for a title either in the database or in an existing playlist, building a playlist satisfying a number of constraints. Each of the two sets for each subject contained one task of each type. (See (Kruijff-Korbayová et al., 2005) for details.)

The wizards, playing the role of the music player, had access to a database of information (but not actual music) of more than 150,000 music albums (almost 1 million songs), extracted from the FreeDB database.[4] We used multiple wizards and gave them freedom to decide about their response and its realization in order to collect data with a variety of interaction strategies.

Both users and wizards could speak freely. The interactions were in German (although most of the titles and artist names in the database are English). In the multimodal setup in SAMMIE-2, the wizards could use speech only, display only, or to combine speech and display, and the users could speak and/or make selections on the screen.

[3]For the second two tasks there was a primary task using a *Lane Change* driving simulator (Mattes, 2003).

[4]FreeDB is freely available at http://www.freedb.org

Since the wizard cannot design screens on the fly, because that would take too long, we implemented modules supporting the wizard by providing automatically calculated screen output options the wizard could select from.

The types of screen output were: (i) a simple text-message conveying how many results were found, (ii) a list of just the names (of albums, songs or artists) with the Bcorresponding number of matches, (iii) a table of the complete search results, and (iv) a table of the complete search results, but only displaying a subset of columns. For each screen output type, the system used heuristics based on the search to decide, e.g., which columns should be displayed. The wizard could chose one of the offered options to display to the user, or decide to clear the user's screen. Otherwise, the user's screen remained unchanged.

We are currently analyzing and annotating the data w.r.t. the interaction strategies and other aspects. The interaction strategies observed in the collected data are driving the design of turn- and sentence-planning (cf. Section 4). We also interviewed both the 'subjects and the wizards after the experiments individually. Their feedback provides us with additional insight concerning the output generation decisions made by the wizards and how successful they were according to the users.

3 Search Results Presentation

Here we present preliminary observations on the presentation of database search results. In speech-only interaction, the wizards typically say the number of results and list them, when the number is small (up to approx. 10, cf. (1)). For more results, they often say the number, and sometimes ask whether or not to list them (cf. (2)). For very large sets of results, the wizards typically say the number and ask the user to narrow down the search, (cf. (3)).

(1) I found 3 tracks. Blackbird, Michelle and Yesterday.
(2) I found 17 tracks. Should I list them?
(3) I found 500 tracks. Please constrain the search.

In multimodal interaction, a commonly used pattern is to simultaneously display screen output and describe what is shown (e.g., *I'll show you the songs by Prince*). Some wizards adapted to the user's requests: if asked to show something (e.g., *Show me the songs by Prince*), they showed it without verbal comments; but if asked a question (e.g., *What songs by Prince are there?* or *What did you find?*), they answered in speech as well as showed the screen output.

"Summaries" A common characteristic in both setups is that the wizards often verbally summarize the search results in some way: most commonly by just reporting the number of results found, as in (3). But sometimes they describe the similarities or differences between the results, as in (4).

(4) 200 are from the 70's and 300 from the 80's.

Such descriptions may help the user to make a choice, and are a desirable type of collaborative behavior for a system. Their automatic generation provides an interesting challenge: It requires the clustering of results, abstraction over specific values and the production of corresponding natural language realization. We are working on static cluster definitions (e.g., production years, genre, album names, etc.), and define suitable ways of referring to them in the turn and sentence planners (e.g., reference to decades). Clusters could also be computed dynamically, which poses two challenges: (a) deciding which clusters are most useful to the user (depending, e.g., on a user model); (b) automatically generating cluster descriptions.

Screen Output Options There were differences in how the wizards rated and used the different screen output options: The table containing most detailed information about the queried song(s) or album(s) was rated best and shown most often by some, while others thought it contained too much information and hence they used it less or never. The screen option containing only a list of songs/albums with their length, received complementary judgments: some of the wizards found it useless because it contained too little information, and they thus did not use it, and others found it very useful because it would not confuse the user by presenting too much information, and they thus used it frequently. Finally, the screen containing a text message conveying only the number of matches, if any, has been hardly used. The differences in the wizards' opinions about what the users would find useful or not clearly indicate the need for evaluation of the usefulness of the different screen output options in particular contexts from the users' view point.

The subjects found the multi-modal presentation strategies helpful in general. However, they often thought that too much information was displayed. They found it distracting, especially while driving. They also asked for more personalized data presentation. We therefore need to develop intelligent ways to reduce the amount of data displayed. This could build on prior work on the generation of "tailored" responses in spoken dialogue according to a user model (Moore et al., 2004).

4 System Components

In this section, we briefly describe the components that are involved in output generation as part of the end-to-end dialogue system for the MP3 player domain we are developing.

Dialogue Management The dialogue manager is based on an agent-based model which views dialogue as collaborative problem-solving (Blaylock et al., 2003). It is implemented in the information-state update approach using DIPPER.[5] Utterances are viewed as negotiation of a shared collaborative problem-solving state, to do things such as determining joint objectives, finding and

[5]DIPPER is available at http://www.ltg.ed.ac.uk/dipper/

instantiating recipes to accomplish them, executing the recipes and monitoring for success.

Turn Planning In monomodal dialogue systems the propositional content is typically realized rather straightforwardly, producing written or spoken output w.r.t. to the issues of *what to say* and *how to say it*. In multimodal dialogue the relationship between the propositional content determined by the dialogue manager and the content realized as output is more complex as the content needs to be reasonably distributed over the available modalities in contextually appropriate ways. This also means that planning multimodal output needs to comprise the issue of *when to present what* according to the available modalities. To meet these challenges, our implementation of the turn planning component is based on a production rule system called *PATE*. Originally developed for the integration of multimodal input (Pfleger, 2004), this component provides an efficient and elegant way of realizing complex processing rules.

Sentence Planning and Realization Our sentence planner is also being implemented in PATE. One of its tasks is to plan the verbal summaries discussed in Section 3. It is also responsible for decisions pertaining to contextualized linguistic realization, such as information structure and referring expressions. Regarding sentence realization, the requirement of contextually appropriate spoken output calls for tools that allow for controlled variation in, e.g, syntactic structure and intonation. We use the OpenCCG system[6] for parsing and generation, and develop a German grammar for it (Gerstenberger and Wolska, 2005).

Speech Synthesis To produce spoken output in German we use the TTS system Mary[7], which enables us to produce contextually appropriate synthesized spoken output by controlling the intonation using a markup based on the German version of the ToBI standard.[8]

Screen Output We are using the generic table presentation tool we developed for the experiment to display tables, lists or text messages generated from the search results. The user can also graphically select items from the respective presentation. For use in the in-car system this table presenter is being adapted to the constraints of the driving situation, e.g., small display with large fonts and a limited number of rows. We are also adding a GUI for controling the MP3 player.

Later in the project, we will perform usability tests, where standard measures such as user satisfaction and task success will be used. The presentation strategies will be tested and evaluated in more specialized experiments with human judges comparing alternative outputs in specific contexts.

References

[Blaylock et al.2003] N. Blaylock, J. Allen, and G. Ferguson. 2003. Managing communicative intentions with collaborative problem solving. In *Current and New Directions in Discourse and Dialogue*, pages 63–84. Kluwer, Dordrecht.

[Gerstenberger and Wolska2005] C. Gerstenberger and M. Wolska. 2005. Introducing Topological Field Information into CCG. In Proc. of the ESSLLI 2005 Student Session, Edinburgh. To appear..

[Kruijff-Korbayová et al.2005] I. Kruijff-Korbayová, N. Blaylock, C. Gerstenberger, V. Rieser, T. Becker, M. Kaißer, P. Poller, and J. Schehl. 2005. An experiment setup for collecting data for adaptive output planning in a multimodal dialogue system. Submitted.

[Mattes2003] S. Mattes. 2003. The lane-change-task as a tool for driver distraction evaluation. In *Proc. of IGfA*.

[Moore et al.2004] J. D. Moore, M. E. Foster, O. Lemon, and M. White. 2004. Generating tailored, comparative descriptions in spoken dialogue. In *Proc. of the Seventeenth International Florida Artificial Intelligence Research Sociey Conference, AAAI Press.*

[Pfleger2004] N. Pfleger. 2004. Context based multimodal fusion. In *ICMI '04: Proc. of the 6th international conference on Multimodal interfaces*, pages 265–272, New York, NY, USA. ACM Press.

[6] OpenCCG is available at http://openccg.sourceforge.net/

[7] Mary TTS is available at http://mary.dfki.de/

[8] http://www.uni-koeln.de/phil-fak/phonetik/gtobi/

DJ GoDiS: Multimodal Menu-based Dialogue in an Asynchronous Information State Update System

David Hjelm, Ann-Charlotte Forslund, Staffan Larsson, Andreas Wallentin
GU Dialogue Systems Lab
Göteborg University, SE 405 30 Göteborg, Sweden
{sl,forslund,dhjelm,andreas}@ling.gu.se

Abstract

We present a demo where menu-based spoken dialogue is used simultaneously with a graphical user interface menu system as an interface to a mp3 player. Input is accepted in either modality, while output is presented in both modalities simultaneously. The system is implemented using a new asynchronous version of TrindiKit. For modality fusion and multimodal generation we are using multimodal GF grammars.

1 Introduction

In this demo we present work on multimodal menu-based dialogue currently being carried out at the Göteborg University Dialogue Systems Lab as part of the TALK project[1]. We are adding support for multimodality to GoDiS and TrindiKit, as well as making use of recent simplifications and improvements on TrindiKit. For modality fusion and multimodal generation we are using multimodal GF grammars.

The basic idea is to use menu-based spoken dialogue (Larsson et al., 2001) together simultaneously with a graphical user interface menu system. Input is accepted in either modality. Output is presented in either or both modalities (spoken interaction and GUI interaction) depending on whether each modality currently provides an information channel between system and user. We believe this is a simple yet very useful way of exploring the benefits of multimodal dialogue. In addition, it has the advantage of subsuming and extending the already familiar menu-based-GUI-style interface.

As a showcase we have implemented the DJ GoDiS application, a multimodal interface to a mp3-player.

2 GoDiS, TrindiKit and GF

GoDiS (Larsson, 2002) is an experimental dialogue system implementing a theory of Issue-Based Dialogue Management based on Ginzburg's concept of Questions Under Discussion (QUD). GoDiS is implemented using the TrindiKit, a toolkit for implementing dialogue move engines and dialogue systems based on the Information State approach (Traum and Larsson, 2003). GoDiS has been adapted to several different dialogue types, domains, and languages, including menu-based action-oriented dialogue when acting as an interface to a mobile phone (Larsson et al., 2001) or VCR. To enable multimodal interpretation and generation, we have

[1]Talk And Look, Tools for Ambient Linguistic Knowledge, EC Project IST-507802

recently integrated the Grammatical Framework (Ranta, 2004) into TrindiKit and GoDiS, using the Open Agent Architecture (OAA). GF is a grammar formalism based on type theory. The division between abstract and concrete syntax enables grammars to be written in parallel for different languages, sharing the same abstract syntax.

3 Asynchronicity in TrindiKit4

In TALK, all partners have agreed to use OAA as a common interface. Previous versions of TrindiKit were compatible with OAA but required dialogue systems to be run by a TrindiKit agent which would then call other agents when necessary. No solvables were offered by TrindiKit to the OAA agent community.

TrindiKit4 offers the possibility to distribute system components across several OAA agents. As a consequence, a number of input and output modules can run simultaneously, and update and inspect the TrindiKit Total Information State (TIS) independently. For instance, a module can listen continuously for speech and update the TIS only when speech has been detected without blocking the rest of the system. The capabilities of each TrindiKit component are published as solvables to the OAA community.

System coordination is done by a special purpose control module, which can be set up to monitor certain TIS variables and execute an associated control algorithm when they are set to a certain value. This can be used e.g. to notify the interpretation module that the input module has updated the TIS. The control algorithm can contain calls to TrindiKit modules or OAA agents as well as TIS updates and checks. Any number of control algorithms can be run in parallel. As OAA enables asynchronous processing, a previous implementation of asynchronicity in TrindiKit has been removed.

In DJ GoDiS, this architecture is exploited by having multiple input and output modules to allow for spoken and GUI-based (as well as written) communication in parallel. All input modules write to the same TIS variable, an input queue. After a certain amount of user inactivity, the contents of the input queue is copied into a string input variable and is considered to make up one user turn. The setting of this variable triggers interpretation, system update, dialogue move selection, generation and system output.

4 Menu-based multimodal dialogue

This approach offers what we believe to be a very flexible and intuitive multimodal interface to any device that can be operated using a standard menu-based GUI interface. Several modes of interaction emerge from a simple setup, including the following:

- The user may use the menu-based GUI in the normal way, without bothering with speech
- The user may use the spoken interface without bothering with the GUI
- The user may make menu choices using speech; these will have the same effect on-screen as making the menu choice by pointing and clicking
- The user may exploit GoDiS' flexible dialogue management to bypass the menu system and give commands and/or provide information as (s)he sees fit. Again, any spoken interactions will result in the corresponding menu options appearing on-screen, thus enabling the user to freely switch modality or combine modalities as desired at any point in the interaction.
- No user-modeling is done apart from keeping track of the shared dialogue state; user adaptation emerges instantly

```
playTask : Task;
volumeTask : Task;

play : Action playTask;
raise_volume : Action volumeTask;

madonna : Object playTask;
level : Object volumeTask;

makeRequest : (t : Task) -> Action t ->  Request t;
makeAnswer : (t : Task) -> Object t -> Answer t;

makePair : (t : Task) -> Action t -> Object t -> Pair;
```

Figure 1: Dependent types in the GF grammar for DJ GoDiS. The dependent type `Task` is used to make sure that actions and objects within the same utterance belong to the same task.

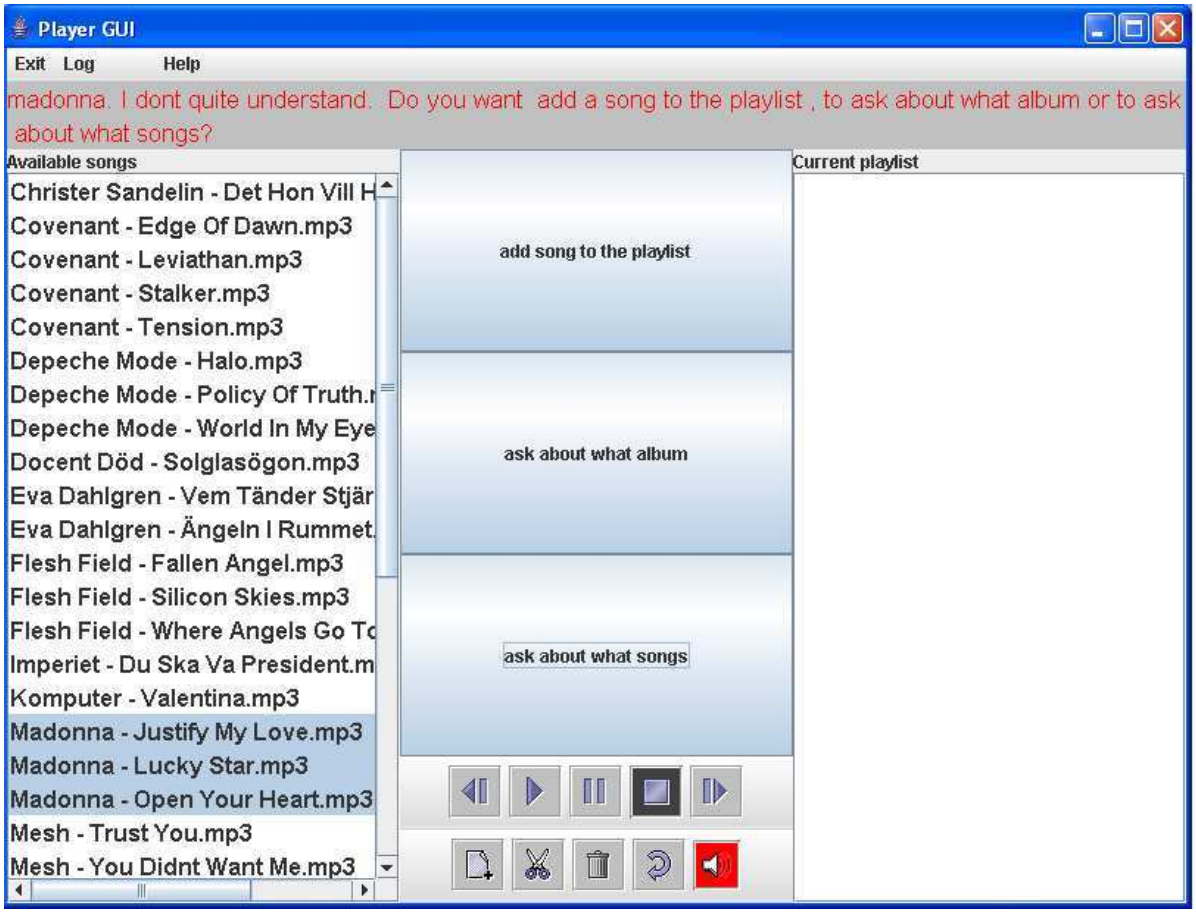


Figure 2: DJ GoDiS GUI. The user has just said "Madonna".

as an effect of the user's choice of modality from utterance to utterance

5 Multimodal grammars in GF and GoDiS

We distinguish between integrated multimodality, where modalities are combined in the same concrete syntax, and parallel multimodality, where each modality has its own concrete syntax (Bringert et al., 2005). In GoDiS we use an integrated multimodal input grammar to parse GUI and speech input and two parallel output grammars generating GUI and spoken output respectively.

The abstract syntax of the two grammars uses dependent types to pose constraints on dialogue move sequences. An excerpt from the abstract syntax is given in figure 1.

GF also support the generation of context free speech recognition grammars. We use this facility to generate a Nuance speech generation grammar from the natural language part of the input grammar.

6 DJ GoDiS functionality

The DJ GoDiS system is capable of performing standard mp3 player tasks, such as playing songs, creating playlists and controlling the volume. The user can also pose queries about e.g. what songs, artists and radio stations are available.

The following example dialogue shows how DJ GoDiS tries to figure out the user's goal by posing a clarification question. (This process is described in Larsson (2002) as "dependent accommodation"):

usr> (CLICKS ON A SONG IN PLAYLIST)
sys> `Do you want to play or remove from playlist?`
(A MENU APPEARS WITH TWO BUTTONS: [PLAY] AND [REMOVE])
usr> `Play`
sys> `OK, play.`
([PLAY] BUTTON IS HIGHLIGHTED)

The next example shows how the user can use two modalities at once to perform a task:

usr> `What songs are there by Madonna?`
sys> (DISPLAYS ALL SONGS BY MADONNA)
usr> `Play this one`
(CLICKS ON SONG 'LUCKY STAR')
sys> `OK, play`
(SONG IS ADDED TO PLAYLIST)
([PLAY] BUTTON IS HIGHLIGHTED)

Figure 2 shows the DJ GoDiS GUI. The user has just said "Madonna". All songs by Madonna are highlighted and three buttons are shown, representing menu choices: [ADD SONG TO PLAYLIST], [ASK ABOUT WHAT ALBUM] and [ASK ABOUT WHAT SONGS].

Acknowledgements

The research reported here was funded by TALK (Talk And Look, Tools for Ambient Linguistic Knowledge), EC Project IST-507802.

References

Björn Bringert, Robin Cooper, Peter Ljunglöf and Aarne Ranta. 2005. Development of multimodal and multilingual grammars: viability and motivation. Deliverable D1.2a, TALK.

Staffan Larsson, Robin Cooper, and Stina Ericsson. 2001. menu2dialog. In *Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 41–45.

Staffan Larsson. 2002. *Issue-based Dialogue Management*. Ph.D. thesis, Göteborg University.

Aarne Ranta. 2004. Grammatical framework. a type-theoretical grammar formalism. *Journal of Functional Programming, vol. 14:2. 2004*, 14(2).

David Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In Ronnie Smith and Jan Kuppevelt, editors, *Current and New Directions in Discourse & Dialogue*. Kluwer Academic Publishers.

Towards Ontology-based Pragmatic Analysis

Berenike Loos **Robert Porzel**

European Media Laboratory, GmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
{berenike.loos,robert.porzel@eml-d.villa-bosch.de}

Abstract

In this paper we describe an ontological model of pragmatic knowledge - using an example from the domain of navigation - that is based on the Descriptive Ontology for Linguistic and Cognitive Engineering and employs a specific ontological module called *Descriptions & Situations*. This framework establishes so-called *ontological patterns*. We employ such a pattern for modeling schematic knowledge of the pragmatics of spatial navigation.

1 Introduction

Spoken multi-modal dialogue systems equipped with the ability to understand and process natural language utterances commonly employ a formal, explicit specification of shared conceptualizations (Gruber, 1993) for machine encoding. At the same time the emerging Semantic Web (Berners-Lee et al., 2001) bases on such formal conceptualizations, called *ontologies* to add semantic information to textual and other data available on the Internet.

In the mobile multimodal dialogue system SmartWeb (Wahlster, 2004) a navigation ontology is necessary, which represents knowledge about the locomotion of the intended user to support car, motorcycle and pedestrian navigation. Existing navigation ontologies (Malyankar, 1999; Gurevych, 2003) describe route mereologies, which do not capture contextual dependencies. Given a single application-specific context, e.g. guiding only pedestrians - always on foot and always on the shortest path, we can employ such a *context-free* ontology. However, if we wish to make use of the many tunable parameters offered by today's route planning and navigational systems, as we will describe below, one must provide the means to find the right setting depending on the actual situation at hand in the least invasive way, i.e. minimizing the amount of parameters and role settings obtained by asking the user.

In the following we describe how the SmartWeb navigational ontology attempts to provide a principled approach to encode pragmatic knowledge about possible dependencies between the specific contextual factors, such as the actual weather, and other settings such as the choice of road type.

2 The SmartWeb Project

Mobile broadband communication technologies - ranging from wireless local area networks to UMTS - and the evolving semantic web technologies set the stage for intelligent web-based services. Together these ser-

vices provide the means for novel ways of interacting with and accessing semantically described information. Based on these developments the SmartWeb project seeks to realize ubiquitous interaction and semantic access via multimodal human-computer interfaces.

The goal of the greater research effort behind this work is to lay the foundations for multimodal user interfaces to access distributed and composable Semantic Web services employing a wide range of mobile devices.

3 The Need for Pragmatic Knowledge

In a mobile dialogue system context information is of high significance as the user expects the offer of topical services, while navigating through a dynamically changing environment (e.g. changing precipitation- and temperature levels and/or traffic- and road conditions), which makes the adequate representation of context knowledge inevitable for the task of natural language understanding (NLU).

In the field of NLU ontologies are a well established instrument for expressing domain knowledge and have been employed in state of the art multi-modal dialogue systems (Gurevych, 2003). Still, the following settings demonstrate the necessity of including extra-linguistic situative knowledge for the domain of human navigation in real space:

- For instance, a pedestrian might prefer public transportation over walking when it is raining even for smaller distances.
- A motor bicyclist might prefer to use winding country roads over interstate highways when it is warm and sunny, but not, when road conditions are bad.
- A car driver might like to take a spatially longer route if shorter ones are blocked or perilous.

4 Integrating Pragmatic Knowledge in the SmartWeb Foundational Ontology

The SmartWeb foundational ontology (Cimiano, 2004) employs the highly axiomatized Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)[1]. It features various modules, e.g. an ontology of plans and a module called *Descriptions & Situations* (Gangemi, 2003). As the focus of our work lies on an application and elaboration of the latter mentioned module, it will be described more closely in the next chapter. Additional to the foundational ontology a domain-independent layer is included which consists of a range of branches from the less axiomatic ontology SUMO (Suggested Upper Merged Ontology; (Niles et al., 2001)), which is known for its intuitive and comprehensible structure.

4.1 Pragmatic *Descriptions & Situations*

The module *Descriptions & Situations* (D&S) is an ontology for representing a variety of reified contexts and states of affairs. In contrast to physical objects or events, the extension of the ontology by non-physical objects poses a challenge to the ontology engineer. The reason for this circumstance is the fact that non-physical objects are taken to have meaning only in combination with some other entity. Accordingly, their logical representation is generally set at the level of theories or models and not at the level of concepts or relations (see (Gangemi, 2003)).

An example for a **situation** could be the instance of some specific person, e.g. `Rainer`, playing the *functional role* of a motorcyclist driving on the `B3` playing the role of a country road on a day whose weather region was valued by `sunny` and `warm`.

In our elaboration an example for a **description** would be a generalization over such

[1] More information on this descriptive and reductionistic approach is found on the WonderWeb Project Homepage: wonderweb.semanticweb.org.

instances, e.g. the description of locomotion would use roles - such as locomotor and path and a parameter such as environment, which adhere to the constraints established by D&S, i.e. that roles are played by endurants, e.g. physical objects and that they are parameterized by regions, e.g. the region encompassed by all weather conditions.

Figure 1 sketches out how this is realized in the D&S module.

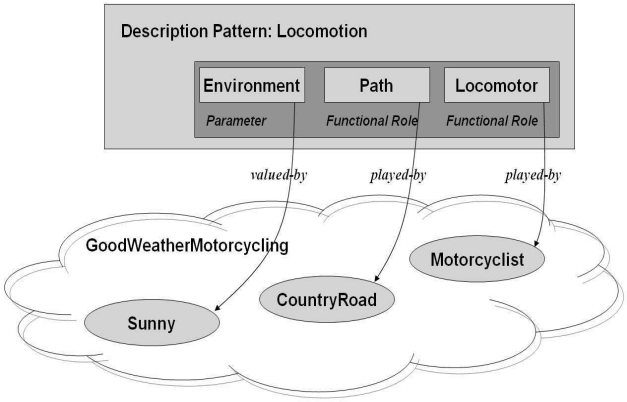


Figure 1: D&S example

One modeling choice that arises hereby concerns the question of how fine-grained such a description and relation hierarchy linked to corresponding roles and parameters should be or if a corresponding axiomatization should bear the burden of associating the pragmatically grouped items of the ground (domain) ontology, e.g. SUNNY, COUNTRYROAD and MOTORCYCLIST for describing the context in which country roads are the filler of choice for motorcyclists on sunny days. In the latter case the corresponding axioms would be the following in the context of GOODWEATHERMOTORCYCLING (GWM) using the predicate situationally_connected (s_c)):

$$\forall(x) \rightarrow GWM(x) \rightarrow$$
$$s\_c(GWM, Sunny) \wedge$$
$$s\_c(GWM, CountryRoad) \wedge$$
$$s\_c(GWM, Motorcyclist)$$

In either case this elaboration of the *Descriptions & Situations* module extends the notion of deriving an instance (situation) from a description by modeling a more general pattern of pragmatic knowledge. Figure 2 shows a corresponding simplified extract from the contextually enhanced ontology with the D&S plug-in.

4.2 Employment in the SmartWeb Project

As the described work will find practical employment in the SmartWeb Project our navigation ontology will be applied to:

- understanding navigational request
- context-dependent route planning.

5 Conclusion

Until now we have done a lot of work on finding the appropriate description for each situation in the D&S module. Unfortunately an axiomatization poses difficulties to most NLP systems and more systematic ways of populating the ontology with the needed description patterns, e.g. by means of learning, need to be found. The next step will be an appropriate quantitative evaluation as proposed by (Porzel & Malaka, 2004). In the future we will, therefore, be concerned with the semi-automatic learning of descriptions from collected situation instances and their consecutive population and quantitative evaluation in the ontology.

Acknowledgments

This work has been partially funded by the German Federal Ministry of Research and Technology (BMBF) as part of the SmartWeb project under Grant 01IMD01E and by the Klaus Tschira Foundation.

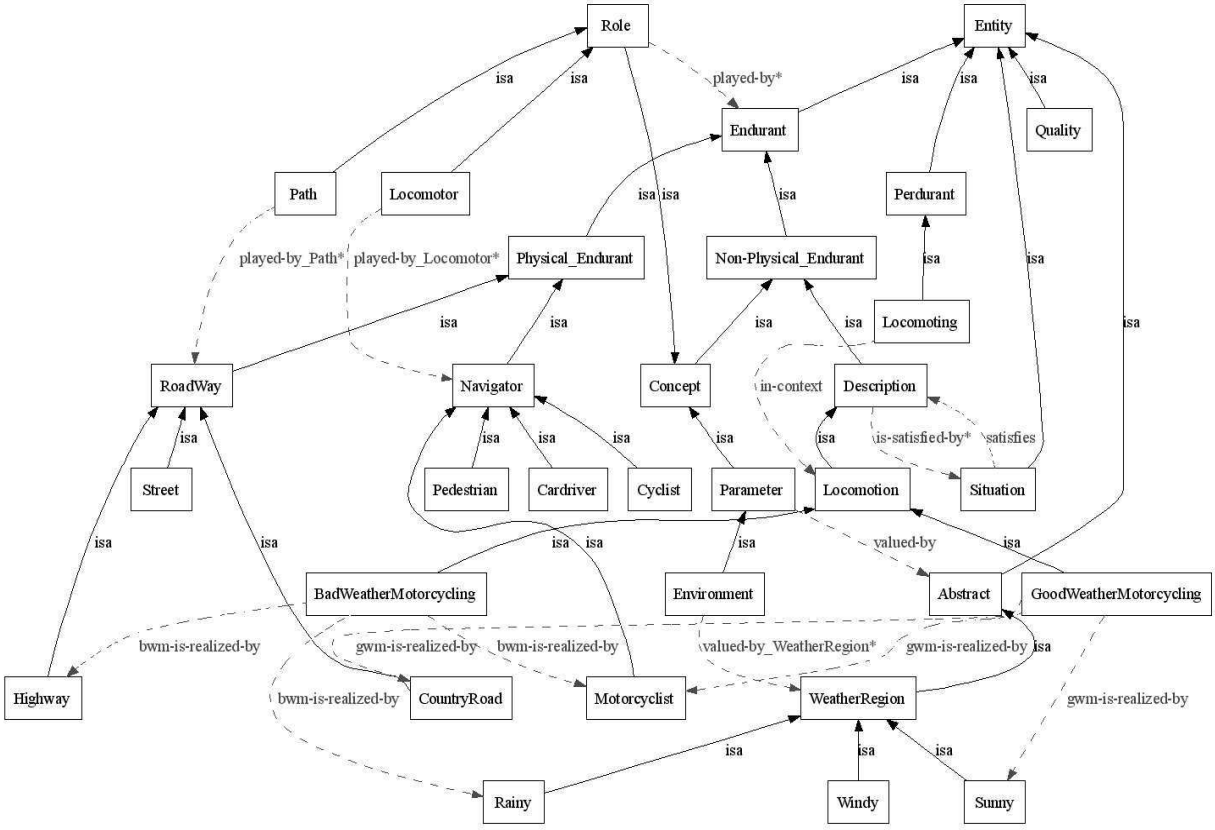


Figure 2: Navigation Ontology extract

References

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific American*, May.

Philipp Cimiano, Andreas Eberhart, Pascal Hitzler, Daniel Oberle, Steffen Staab, and Rudi Studer. 2004. The smartweb foundational ontology. *SmartWeb Project Report*.

Aldo Gangemi, Peter Mika. 2003. Understanding the Semantic Web through Descriptions and Situations. In *Proceedings of the ODBASE Conference*. Springer.

Thomas Gruber. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*, (5).

Iryna Gurevych, Robert Porzel, and Stefan Merten. 2003. Less is more: Using a single knowledge representation in dialogue systems. In *Proceedings of the HLT/NAACL Text Meaning Workshop*, Edmonton, Canada.

Raphael Malyankar. 1999. Creating a Navigation Ontology. Workshop on Ontology Management. AAAI-99, Orlando, FL. In *Technical Report WS-99-13*, AAAI, Menlo Park, CA.

Ian Niles and Adam Pease. 2001. Towards a standard upper ontology. In Chris Welty and Barry Smith, editors, Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine.

Robert Porzel Iryna Gurevych. 2003. Contextual Coherence in Natural Language Processing. In Blackburn, P., Ghidini, C., Turner, R., Giunchiglia, F. (eds). Modeling and Using Context, LNAI 2680, Springer, Berlin.

Robert Porzel & Rainer Malaka. 2004. A Task-based Approach for Ontology Evaluation. In *ECAI-2004 Workshop on Ontology Learning and Population*, Valencia, Spain.

Wolfgang Wahlster. 2004. SmartWeb: Mobile Applications of the Semantic Web. In *Proceedings of Informatik*, Ulm, Germany.