Discourse Phenomena in Tutorial Dialogs on Mathematical Proofs

Christoph Benzmüller[1], Armin Fiedler[1], Malte Gabsdil[2], Helmut Horacek[1],
Ivana Kruijff-Korbayová[2], Dimitra Tsovaltzi[2], Bao Quoc Vo[1], Magdalena Wolska[2]

[1]Fachrichtung Informatik [2]Fachrichtung Computerlinguistik
Universität des Saarlandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany
{chris,afiedler,horacek,siekmann,bao}@ags.uni-sb.de, {gabsdil,korbay,pinkal,dimitra,magda}@coli.uni-sb.de

Abstract. Dialogs in formal domains, such as mathematics, are characterized by a mixture of telegraphic natural language text and embedded formal expressions. Due to the lack of empirical data for such environments, we have collected a corpus of dialogs with a simulated tutoring system for teaching proofs in naive set theory. The analysis of this corpus enabled us to identify genre-specific variants of linguistic phenomena which impose specific requirements on natural language dialog management.

1 Introduction

Dialogs about problem solving in mathematics are characterized by a mixture of telegraphic natural language text and embedded formal expressions. Behaving adequately in such an environment is extremely important for tutorial systems, as follows from Moore's empirical findings which show that flexible natural language dialog is needed to support active learning [9]. However, most state-of-the-art tutorial systems are only able to process limited forms of dialogs, either menu-based or requiring exact wordings [10, 2, 6].

Motivated by the lack of empirical data, we have collected a corpus of dialogs with a simulated tutoring system for teaching proofs in naive set theory, to identify genre-specific variants of linguistic phenomena which impose specific requirements on natural language dialog management. This work is embedded in a project whose goal is to develop a mathematical tutoring system with flexible natural language dialog.

The outline of this paper is as follows. We first present the aims of our project. Next, we describe an experiment in which we collected a corpus of natural language tutorial dialogs. We follow with an analysis of the phenomena observed. Finally, we discuss challenges for natural language dialog management.

2 The DIALOG Project

The goal of the DIALOG project [1] [11] is (i) to empirically investigate the use of flexible natural language dialog in tutoring mathematics, and (ii) to develop an experimental prototype system gradually embodying the empirical findings. The experimental system will engage in a dialog in written natural language (and later also in multimodal forms of communication based on diagrams, spoken language and animated mathematical function displays) to help a student understand and construct mathematical proofs. The overall scenario for the system is illustrated in Figure 1, including the following components:

- *Learning Environment* Students take an interactive course in the relevant subfield of mathematics with the web-based system ACTIVEMATH [8].
- *Mathematical Proof Assistant* It is used for checking the contribution of user specified inference steps to the problem-solving goal, which is based on reasoning carried out by the ΩMEGA system [14].
- *Proof Manager* In the course of the interactive tutorial session, the user may explore alternative proofs, or make various attempts at constructing a valid proof, involving both valid and invalid inference steps.
- *Dialog Manager* We employ the Information-State (IS) Update approach to dialog management developed in the TRINDI and SIRIDUS projects [13].
- *Knowledge Resources* The static knowledge in our scenario includes *pedagogical knowledge* (generic and domain-specific teaching strategies), and *mathematical knowledge* (in our MBase system [7]). The dynamic knowledge includes the student model, as well as the information state maintained by the dialog manager.

[1] The DIALOG project is part of the Collaborative Research Center on *Resource-Adaptive Cognitive Processes* (SFB 378) at University of the Saarland [12].

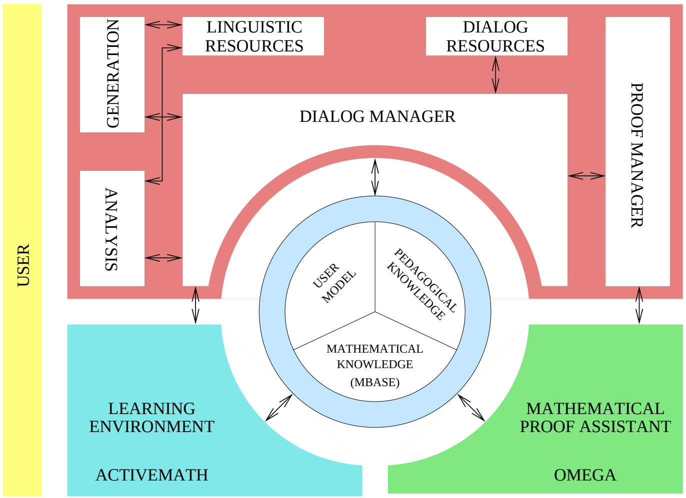

Fig. 1. DIALOG project scenario.

3 Empirical Study

We conducted a *Wizard-of-Oz (WOz)* experiment, supported by a tool [4], in order to collect a corpus of tutorial dialogs in the naive set theory domain. By choosing a WOz approach rather than human tutoring, the dialog data we collect (i) represents the users' behavior in interactions following the algorithms incorporating dialog strategies and (ii) provides early feedback on these algorithms.

24 subjects with varying educational background and prior mathematical knowledge ranging from little to fair participated in the experiment. The experiment consisted of several phases: (1) *preparation and pre-test* on paper, (2) *tutoring session* proper, and (3) *post-test and evaluation questionnaire*, on paper again. The subjects had to prove theorems, by applying the De Morgan laws $(K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D)))$, and by making use of the concepts power set $(A \cap B \in P((A \cup C) \cap (B \cup C)))$ and complement (When $A \subseteq K(B)$, then $B \subseteq K(A)$, where K signifi es *K*omplement in German).

The interface enabled the subject to type text or insert mathematical symbols by clicking on buttons. The subject was instructed to enter steps of a proof rather than the complete proof as a whole, in order to enable a dialog with the system. The tutor-wizard's task was to respond to the student's utterances following a given algorithm. The wizard fi rst classifi ed the completeness, accuracy, and relevance of the subject's utterance with respect to a valid proof of the theorem at hand. Then, the wizard decided what dialog moves to make next and verbalized them. The wizard was free to mix text with formulas [5].

4 Preliminary analysis of the linguistic data

In this section, we present an examination of the linguistic data obtained through the WOz experiment, focusing on the phenomena present in the student utterances. Examples appear in Figure 2 (the original German versions of utterances together with their English translation). We have divided the phenomena observed into three categories: (1) references, (2) ambiguities, and (3) imprecision, according to the dominating linguistic concept. All of these are associated with multiple interpretations on general linguistic grounds. However, handling these examples adequately within tutorial dialogs requires identifying the intended reading whenever possible by exploiting domain knowledge and expections.

4.1 References

Two types of references were identifi ed in this setting: general references to the prior dialog context (anaphora, discourse references in the classical sense), and specifi c references to the domain concepts (domain-specifi c references for the interpretation of which domain knowledge is required):

	Utterance	Translation
References	(1) Potenzmenge enthält alle Teilmengen, also auch $(A \cap B)$	*A power set contains all subsets, hence also $(A \cap B)$*
	(2) $K((A \cup B) \cap (C \cup D)) = K(A \cup B) \cup K(C \cup D)$ de Morgan Regel 2 auf beide Komplemente angewendet	*de Morgan rule 2 applied to both complements*
	(3) $P(A \cap B), P(C) \subseteq P((A \cap B) \cup C)$ Als nächstes stelle ich die rechte Seite unter Anwendung der Eigenschaften von Mengenoperationen so um, daß $P(A \cap B)$ vereinigt mit einer anderen Menge herauskommt.	*Next, I will recast the right side by applying properties of set operations in such a way that this results in $P(A \cap B)$ union some other set*
Ambiguities	(4) de Morgan Regel 1 gilt auch für $K(C \cup D)$ de Morgan Regel 2 besagt $K(A \cap B) = K(A) \cup K(B)$. In diesem Fall z.B. $K(A) =$ dem Begriff $K(A \cup B)$ und $K(B) =$ dem Begriff $K(C \cup D)$. Deshalb ist dann $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$	*de Morgan rule 1 also holds for $K(C \cup D)$ de Morgan rule 2 means $K(A \cap B) = K(A) \cup K(B)$. In this case e.g. $K(A) =$ the term $K(A \cup B)$ and $K(B) =$ the term $K(C \cup D)$. Therefore $K((A \cup B) \cap (C \cup D) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$*
Imprecision	(5) $(A \cup B)$ muß in $P((A \cup C) \cap (B \cup C))$ sein, da $(A \cap B) \in (A \cap B) \cup C$	*$(A \cup B)$ must be in $P((A \cup C) \cap (B \cup C))$, since $(A \cap B) \in (A \cap B) \cup C$*
	(6) $(B \cup A)) \subseteq C$ $(B \cup A)) \subseteq D$. Wenn A Teilmenge von C und B Teilmenge von C dann müssen beide Mengen zusammen ebenfalls eine Teilmenge von C sein. Gleiches gilt mit D $K(C \cap D) \cup K(A \cap B)$ Anwendung der de Morgan Regeln. $((B \cup A)) \subseteq C$ $(B \cup A)) \subseteq D$.	*If A is a subset of C and B a subset of C, then both sets together must also be a subset of C. The same holds for D. $K(C \cap D) \cup K(A \cap B)$ applying the de Morgan rules. $((B \cup A)) \subseteq C$ $(B \cup A)) \subseteq D$.*

Fig. 2. Examples of dialog utterances. The predicates P and K stand for power set and complement, respectively.

- *Discourse references* are illustrated by the use of (1) pronouns (e.g. demonstrative: "*this* also holds vice-versa"), (2) discourse deictic anaphora ("the *above* expression", "apply *here* the axiom ... in opposite direction"), (3) noun phrase anaphora (e.g. "on *the left side*", "for the *inner parenthesis*"). Expressions (1) and (2) refer to an assertion and an expression, respectively, to be determined according to the discourse context. The references in (3) are incomplete specifications. "Left side" refers to an equation, and "inner parenthesis", which is metonymic, refers to the expression enclosed by it.
- *Domain objects* have known denominations which are referred to with their natural language names, e.g., "power set" as a natural language expression has the same denotation as the predicate P in a formal expression about sets. Additionally, such reference may be ambiguous between its *generic* aspect, where it denotes a domain object as a type vs. its use as a *specific* reference to a particular instance (or: token) of that type. Generic and specific references may appear within the same utterance (cf. (1), where "Potenzmenge" (powerset) is used as a generic reference, whereas $A \cap B$ is a specific reference to a subset of a specific instance of the power set). Likewise, the example (2) illustrates a combination of discourse and domain-specific references, where "both" is used in the pronominal function, and the noun "complement" refers to the domain concept in the formal expression. The whole expression, on the other hand, clarifies the scope of an operation performed on the formal expression.

4.2 Ambiguities

The ambiguities observed comprise structural ambiguities which may be due to the scope of domain relations or punctuation uses, as well as lexical ambiguities of connectives and domain relations:

- *Domain relations* comprise propositional logic junctors, logical derivation, and justifications, and they also can be expressed by natural language. The interpretation of descriptions in which these relations appear is problematic because of, among others, ambiguities concerning the scope. For instance, "A and B implies C" has two structurally different interpretations.

– *Punctuation*, in the context of the mathematical domain, introduces structural ambiguity in interpretation of text chunks. A comma has been observed to be ambiguously used to mean enumeration, implication, and conjunction. For example, in (3), the comma is meant to be interpreted as a logical “and” that is followed by the logical consequence of the premise preceding it.
– *Lexical ambiguity* The meta-language vs. object language problem frequently occurs in our corpus with the interpretation of the connective “and”. Two uses of “and” have been observed: grammatical (“and” as a grammatical conjunction), and logical (as a logical connective). In the grammatical sense itself, “and” has been observed to be used in two functions: in enumeration (to mean “also”), and colloquially to express a “logical consequence” (or here: “logical derivation”), see: “$P(C) \cup P(A \cap B) \subseteq P(C \cup (A \cap B)$, *and* $A \cap B \subseteq P(A \cap B)$ *and* this implies $A \cap B \subseteq P((A \cup C) \cap (B \cup C))$”. Moreover, the mathematical relation “=”, which is technically used within equations or for indicating a value assignment, has also been found as an indicator for a substitution of a specific term for an expression in an axiom ((4), $K(A \cup B)$ to be substituted for $K(A)$, and $K(C \cup D)$ for $K(B)$, respectively).

4.3 Imprecision of natural language expressions

The imprecision of natural language use in our domain manifests itself in the imprecision of references to mathematical concepts. They are described by precisely defined symbols in the formal language of mathematics, and verbalized by agreed expressions of natural language (e.g. “powerset”, “implies”). As opposed to that, student utterances are sometimes flawed in domain terminological terms. For example, the conveyed relations in a mathematical expression may be imprecise in the sense that the natural language formulation fits several domain relations (e.g., within the domain of mathematical sets, “must be in” in (5) can be interpreted as “element” or “subset”; and “both sets *together*” in (6) as union or intersection).

On the other hand, imprecise use of formal domain concepts (and symbols) may also result in material error. A special case of an incorrect expression consists in the confusion of conceptually related operators or functions. Examples of commonly confused conceptually related operations in our domain include the use of “$=$” instead of “$\subseteq$” (too specific relation) or “$\in$” instead of “$\subset$” (sort incompatibility on similar relations). A different kind of incorrectly referred domain relation is the following: “The intersection of two sets is less or equal to the smaller one of *these* sets”, which, if taken literaly, results in a type clash (size relation is not applicable to sets). Nevertheless, the utterance is perfectly understandable for a human expert, when “less than” is (somehow metaphorically) interpreted as “subset of”.

5 Implications for dialog management

In view of the phenomena discussed, a typical dialog managemant component of a task-oriented system would invoke several clarification dialogs in order avoid even the slightest risk of misunderstanding. In a tutorial environment, however, the presence of accurate domain knowledge and task control potentially enables dealing without many otherwise unavoidable clarification dialogs in favor of elaborating the proper tutorial goal. For example, imprecise expressions as discussed in the previous subsection can be treated as “near misses” in the tutorial context, provided the system is able to derive the correct interpretation.

Considering the phenomena described above, we acknowledge the need for non-standard treatment of the linguistic data in our system. Besides the typical consequences of dealing with domain-specific linguistic data (such as specialized vocabulary with which the lexicon of, for instance, a morphological analyser must be augmented) we see immediate consequences in the following aspects of the analysis:

Content separation (mathematics vs. language) is required for utterance parsing. Mathematical expressions interleaved with natural language pose a challenge for automatic parsing. A simplistic solution could approach this in two stages: (1) identify and extract the formulas from the natural language, (2) parse the natural language text without the extracted formulas. However, problems here arise when elements of natural language are interwoven into what should be a mathematical expression, e.g. ”$A \cap B$ on the left side is $\in$ of $C \cup (A \cap B)$, which is only augmented by C”. This problem commonly occurs with focus function words such as in ”then A is also $\subset K(B)$”, or with quantifiers ”B contains no x $\in A$”. In the latter example, the scope of ”no” is not the full expression ”$x \in A$,” but rather ”x” alone. In order to handle such utterances adequately, natural language and mathematical expression analyses have to be interleaved more tightly. In addition, missing punctuation and the fragmentary and vague nature of the utterances poses a challenge for parsing both the natural language text and mathematical content.

The *discourse model* needs to be dynamically updated not only with respect to the linguistic content, but also w.r.t. the domain (formulas). References, both to prior linguistic and mathematical elements of discourse need to be resolved and reflected in the discourse model. The challenge here is posed mainly by the latter as it is strictly dependent on the domain context, and domain proof-specifi c knowledge is required for resolution. Additionally, the discourse model needs to keep track of cross-turn references (student-turn content referring to tutor-turn content, and vice versa).

The *knowledge resources* for analysis need to be augmented by domain-specifi c information: (i) The lexicon of the analysis tools needs to include the special vocabulary (related to proving and sets theory). (ii) Considering the unconventional vocabulary encountered, methods of identifying non-standard synonyms need to be devised. (iii) Additional knowledge-bases of domain-specifi c information need to be constructed to reflect: (a) correspondences between the natural language expressions and mathematical symbols (e.g. includes, subset, etc.), (b) hierarchical relations between the names of domain concepts as expressed in natural language (to mirror the hierarchical structure of domain concepts [3]). In partiuclar, the domain-specifi c aspect of metonymic relations and metaphoric uses has to be captured by these representations.

6 Conclusions and Future Research

We briefly presented the overall framework of the DIALOG project which aims at the development of a mathematical tutoring system with flexible dialog. We reported on a WOz experiment in which we collected a corpus of tutorial dialogs with 24 subjects on several problems in naive set theory. The corpus also enabled us to identify interesting cases with genre-specifi c properties which pose challenges for natural language dialog management. In the next stages of the project, we will continue to investigate the interaction between various system parts and gradually implement the missing system components.

References

1. *Papers from the 2000 AAAI Fall Symposium on Building Dialogue Systems for Tutorial Applications*. AAAI Press, 2000.
2. V. Aleven and K. Koedinger. The need for tutorial dialog to support self-explanation. In [1], pages 65–73.
3. A. Fiedler, A. Franke, H. Horacek, M. Moschner, M. Pollet, and V. Sorge. Ontological Issues in the Representation and Presentation of Mathematical Concepts. Workshop on Ontologies and Semantic Interoperability at ECAI-2002, 2002.
4. A. Fiedler and M. Gabsdil. Supporting Progressive Refinement of Wizard-of-Oz Experiments. In Proceedings of the Sixth International Conference on Intelligent Tutoring Systems—Workshop W6: Empirical Methods for Tutorial Dialogue Systems, 2002.
5. A. Fiedler and D. Tsovaltzi. Automating Hinting in Mathematical Tutorial Dialogue. Proceedings of the EACL-03 Workshop on Dialogue Systems: interaction, adaptation and styles of management, Budapest, 2003.
6. N. Heffernan and K. Koedinger. Intelligent tutoring systems are missing the tutor: Building a more strategic dialog-based tutor. In [1], pages 14–19.
7. M. Kohlhase and A. Franke. Mbase: Representing knowledge and context for the integration of mathematical software systems. *Journal of Symbolic Computation*, 32(4):365–402, 2000.
8. E. Melis et al. ACTIVEMATH: A generic and adaptive web-based learning environment. *Artifical Intelligence in Education*, 12(4), 2001.
9. J. Moore. What makes human explanations effective? In *In Proc. of the Fifteenth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ. Earlbaum, 2000.
10. N. Person, A. Graesser, D. Harter, and E. Mathews. Dialog move generation and conversation management in AutoTutor. In [1], pages 45–51.
11. M. Pinkal, J. Siekmann, and C. Benzmüller. Projektantrag Teilprojekt MI3 — DIALOG: Tutorieller Dialog mit einem mathematischen Assistenzsystem. In *Fortsetzungsantrag SFB 378 — Ressourcenadaptive kognitve Prozesse*, Saarbrücken, Germany, 2001. Universität des Saarlandes.
12. SFB 378 web-site: http://http://www.ling.gu.se/projekt/siridus/.
13. TRINDI project: http://www.ling.gu.se/research/projects/trindi/.
14. J. Siekmann et al. Proof development with ΩMEGA. In *Proceedings of the 18th Conference on Automated Deduction*, LNAI 2392, Copenhagen, DENMARK, 2002. Springer Verlag.