An annotated corpus of tutorial dialogs on mathematical theorem proving

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

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

Abstract

Our goal is to develop a flexible dialog system for tutoring mathematical problem solving. Empirical findings in the area of intelligent tutoring show that flexible natural language dialog supports active learning. Therefore, we focus on the development of solutions allowing flexible dialog. However, little is known about the use of natural language in dialog settings in formal domains, such as mathematics, due to the lack of empirical data. We designed and performed an experiment with a simulated tutorial dialog system for teaching proofs in naive set theory. To investigate the correlations between (i) domain-specific content and its linguistic realization, and (ii) the use, distribution, and linguistic realization of dialog moves, we are annotating the corpus with (i) dependency-based semantic relations that build up the linguistic meaning of the utterances and (ii) with dialog moves.

1. Introduction

In the DIALOG[1] project (Benzmüller et al., 2003a), we are investigating and modeling semantic and pragmatic phenomena in dialogs focused on tutoring problem solving skills in mathematics. The goal is to (i) empirically investigate the use of flexible natural language dialog in tutoring mathematics, and (ii) develop an experimental prototype system gradually embodying the empirical findings. The experimental system will engage in a written dialog to help students understand and construct mathematical proofs.

The central component of the system is the Dialog Manager (DM) implementing the Information State (IS) based approach (Traum and Larsson, 2003). Proofs proposed by the student are represented and maintained by a Proof Manager that communicates with an automated theorem prover, ΩMEGA (Siekmann et al., 2002), to perform domain reasoning and evaluate the student's proof contribution.

Empirical findings in intelligent tutoring show that flexible natural language dialog supports active learning (Moore, 1993). In our project, therefore, the focus has been on solutions allowing flexible dialog. However, little is known about the use of natural language in a dialog setting in formal domains, such as mathematics, due to the lack of empirical data. To collect a corpus of data, we designed and performed an experiment with a simulated tutorial dialog system for teaching proofs in naive set theory.

In order to investigate systematically correlations between (i) domain-specific content and its linguistic realization, and (ii) use, distribution, and linguistic realization of dialog moves in our domain, we are annotating the corpus with dependency-based semantic relations that build up the linguistic meaning of the utterances and with dialog moves.

In this paper, we present our corpus and an ongoing annotation effort guided by a preliminary corpus analysis. The corpus annotation proceeds in three directions: at the discourse level, we are annotating the deep semantics of the dialog utterances; at the dialog structure level, we are annotating the dialog moves; finally, at the tutoring task level, the information related to the content of the utterances evaluated as proof-steps (in case of the student) and realizing a teaching method (in case of the tutor utterances).

The goal of the annotation is to inform the development of the dialog management and the input analysis modules of the system. Further domain-level annotations will give us insights into such issues as the relations between the structure of human-constructed proof viewed as an argumentative discourse on the one hand, and as a series of logical inferences on the other. Moreover, we want to learn about the interactions between the three levels (e.g. correlations between semantic content of the utterances and their dialog move functions, particular to our domain).

This paper is organized as follows: in Section 2., we present the setup of the data collection experiment; in Section 3., we describe our corpus of linguistic data; in Sections 4. and 5., we discuss linguistic meaning and dialog move annotation; in Section 6., finally, we present directions in which we are planning to extend the annotations.

2. The Experiment

24 subjects with varying educational background and little to fair prior mathematical knowledge participated in a *Wizard-of-Oz* experiment. The subjects were told they were evaluating a tutoring system with natural language dialog capabilities. At the tutoring session, they were asked to prove 3 theorems[2]: (i) $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$; (ii) $A \cap B \in P((A \cup C) \cap (B \cup C))$; (iii) *When* $A \subseteq K(B)$, *then* $B \subseteq K(A)$. The subjects were instructed to enter proof steps, rather than complete proofs at once, to encourage dialog with the system, and were free in their linguistic expression.[3]

[1]The DIALOG project is a collaboration between the Computer Science and Computational Linguistics departments of University of the Saarland as part of the Collaborative Research Center on *Resource-Adaptive Cognitive Processes*, SFB 378 (`www.coli.uni-sb.de/sfb378`).

[2]K stands for set complement and P for power set.

[3]Buttons were available in the interface for inserting mathematical symbols, while literals were typed on the keyboard. The

The wizard's task was to evaluate the subject's contribution as to its appropriateness. The subjects were split into three groups exposed to different conditions. In the *minimal feedback* condition, the wizard gave information only on whether answers were correct and complete. In *didactic tutoring*, the wizard reacted to incorrect answers by giving the correct answer and explaining it. In *socratic tutoring*, the wizard hinted at the correct answer. What kind of hint should be produced was determined by an implemented hinting algorithm (Fiedler and Tsovaltzi, 2003).

The experiment sessions were recorded by a *Wizard-of-Oz* support-tool (Fiedler and Gabsdil, 2002). In addition, we asked the subjects to think aloud while working on the problems, and we video- and audio-taped them. The corpus contains the data of 22 subjects. The experiment setup is presented in more detail in (Benzmüller et al., 2003b).

3. The Corpus

The corpus comprises 66 sets of dialog session logs with 12 turns on average. There are 1115 in total, of which 393 are student sentences. An average student and tutor turn consists of 1 and 2 sentences respectively.

During each session, the following material was collected: dialog session log file, think-aloud audio recording log, and video log file of the subject-system interaction. The log files have been translated into English. Both literary and word-to-gloss translations of student and tutor turns have been added to the dialog log files as separate entries.

Raw log files contain *in-line* session data recorded automatically by the *Wizard-of-Oz* support-tool and tutoring-related information entered by the wizard during sessions. The *session level data* include: experiment number, turn label, time-stamp information, and state of a simple finite-state model of IS changes. The in-line *tutoring-related information* include the answer-category and, in socratic tutoring sessions, the category of the hint the wizard realized during tutoring. The answer category represents the contribution's evaluation (cf. Section 5.1.2.), while the hint category represents the hint type selected by the wizard while following a pre-designed algorithm (cf. Section 5.1.1.).

The think-aloud recordings were transcribed and annotated with simple speech act categories such as "signaling emotions", "self-explanation", "addressing experimenter".

The main body of annotations concerns aspects of (i) language, (ii) dialog, and (iii) tutoring. Dialog and language level annotations are performed using the MMAX tool that supports multi-level annotation (Müller and Strube, 2003). The annotated tutoring level information is encoded in the log files as ASCII text. We describe the annotations at the three above levels in sections below.

The corpus is made available upon request.[4] Presently, only raw dialog data is available both in ASCII format and as LaTeX-formatted text of turns. After the annotation is completed, we will make the annotated files also available.

dialogs were typed in German. In the reminder of this paper, we will present only English translations of example utterances for brevity of presentation.

[4]To obtain the corpus, send email to dialog@ags.uni-sb.de.

4. Linguistic Meaning Annotation

In order to guide construction of a grammar for parsing, we are annotating the linguistic meaning of the sentences in terms of semantic dependency relations.

By linguistic meaning (LM), we understand the dependency-based deep semantics of a sentence in the sense of the Prague School, as employed in the Functional Generative Description (FGD) (Sgall et al., 1986; Kruijff, 2001). It represents the literal meaning of an utterance rather than a domain-specific interpretation. In FGD, the central frame unit of a sentence/clause is the head verb which specifies the *tectogrammatical relations* (TRs) of its dependents (*participants*). Further distinction is drawn into *inner participants*, such as Actor, Patient, Addressee, and *free modifications*, such as Location, Means, Direction. Using TRs rather than surface grammatical roles provides a generalized view of the correlations between domain-specific content and its linguistic realization.

We generalize and simplify the Prague Dependency Treebank[5] collection of TRs described in (Hajičová et al., 2000). The reason for the simplification is, among others, to distinguish which roles must be understood metaphorically given our sub-language domain. In order to allow for ambiguity in recognition of TRs, we organize them hierarchically into a taxonomy. The most commonly occurring roles in our context are Cause, Condition, and Result-Conclusion (which coincide with the rhetorical relations in the proof's argumentative structure), for example[6]:

As $[A \subseteq K(B)$ holds$]_{<\text{CAUSE}>}$, all x that are in A **are** not in B
As $A \subseteq K(B)$ **holds**, [all x that are in A are not in $B]_{<\text{RES}>}$
When $[A \subseteq K(B)]_{<\text{COND}>}$, then $A \cap B = \emptyset$
When $A \subseteq K(B)$, then $[A \cap B = \emptyset]_{<\text{RES}>}$

Other commonly found TRs (aside from the inner participant roles) include Norm-Criterion, for example:

$K(A \cup B)$ is by [DeMorgan-1]$_{<\text{NORM}>}$ $K(A) \cap K(B)$
$K(A \cup B)$ equals, according to [deMorgan-1]$_{<\text{NORM}>}$,$K(A) \cap K(B)$

We group other modifications into sets of HasProperty, GeneralRelation (adjectival and clausal modification), and Other (a catch-all category), for example:

then all A and B must be contained [in $C]_{<\text{PROP-LOC}>}$
all elements [from $A]_{<\text{PROP-FROM}>}$ are contained [in $K(B)]_{<\text{PROP-LOC}>}$
From $A \subseteq U \setminus B$ [with $A \cap B = \emptyset]_{<\text{OTHER}>}$ follows that $B \subseteq U \setminus A$

where PROP-LOC is a HasProperty relation of type Location, GENREL is a general relation, and PROP-FROM is a HasProperty of type Direction-From or From-Source.

The input understanding module employs a rich lexically based grammar for parsing input sentences. We are using a Multi-Modal Combinatory Categorial Grammar formalism (Baldridge, 2002) combined with Hybrid Logic Dependency Semantics (HLDS) representation of the LM constructed in parallel with the syntax through unification of HLDS terms (Baldridge and Kruijff, 2002). The LM annotation is guiding the development of the grammar.

5. Dialog Moves Annotation

In order to find out correlations between the dialog move dimensions and to build a dialog model, we are annotating dialogs moves in the corpus.

[5]http://quest.ms.mff.cuni.cz/pdt/

[6]The presentation of the annotation is schematic.

Based on a preliminary analysis of the corpus, we developed a taxonomy of dialog moves and a dialog move annotation scheme. We have adopted a commonly accepted high-level standard taxonomy of DAMSL dialog moves (Core and Allen, 1993; Allen and Core, 1997), which we extended for our specific domain of tutoring mathematical problem solving. A preliminary version of our annotation scheme in presented in (Tsovaltzi and Karagjosova, 2004).[7] To the DAMSL taxonomy, we added a new dimension which specifies an utterance's task-level function. The *task dimension* captures functions that are particular to the task at hand and its manipulation, and hence to the genre. There are two sub-dimensions, namely proof task and tutoring task. The particular separation of genre specific moves from genre independent ones captures the overall philosophy of a Dialog Manager easily reconfigurable to other tutoring domains and other dialog genres. At the task level, we annotate information pertaining to the proof-steps (for student utterances) and to the teaching method (for tutor utterances).

An instance of an utterance that has functions in three dimensions, potentially inter-related, is the following tutor turn:

Can you explain that in more detail?

It is an *info_request* in the forward looking dimension (it requests information), a *request_clarification* in the backward looking one (it requests clarification of a previous utterance) and a *check_or_problem* in the task dimension (the tutor is trying to diagnose a problem that needs treatment).

5.1. Tutoring task

In this section, we present the tutoring-related annotations. We are only concerned here with answers that attempt to bring the task forward. In the tutoring-related annotations, the most prominent are the categories of hints the tutor gave and the categories of the subjects' answers.

5.1.1. A Taxonomy of Hints

In this section we explain the philosophy and the structure of our hint taxonomy. The taxonomy includes more than the hint categories mentioned in this section. The full taxonomy can be found in (Fiedler and Tsovaltzi, 2003).

Philosophy and Structure Our hint taxonomy was derived with regard to the underlying function common for different surface realizations of hints, mainly responsible for their educational effect. To capture different functions of a hint we define categories across two dimensions, namely (i) active vs. passive, and (ii) domain-relation vs. domain-object vs. inference-rule vs. substitution vs. meta-reasoning vs. performable-step. Each of these classes consists of single hint categories that elaborate on one of the proof step attributes. The hint categories are grouped in classes according to the information they address in relation to the domain and the proof. By and large, hints of the passive function of a class in the second dimension constitute the hints of the active function of its immediately subordinate class, in the same dimension.

[7] This scheme is is still being tested.

First Dimension The first dimension distinguishes between hint's active and passive functions, where the difference is in how the information the tutor wants to refer to is approached. The distinction resembles that of *backward-* vs. *forward-looking* function of dialog acts in DAMSL. The *active* function looks forward and seeks, by means of *eliciting*, to help student access further information needed to come closer to the solution. The *passive* function refers to a small piece of information provided each time to bring the student closer to an answer; the tutor *gives away* information, e.g. previously unsuccessfully elicited.

Second Dimension *Domain-relation* hints address relations between mathematical concepts in the domain. *Domain-object* hints address domain objects (e.g., *give-away-relevant-concept* names the proposition's most prominent concept whose definition needs to be used). Since this class is subordinate to *domain-relation*, the hints in it are more revealing. The passive function of *domain-object* hints elicits the applicable inference rule, therefore, is part of the active function of the respective class. Finally, the *pragmatic* hints refer to pragmatic attributes of the expected answer. The active function hint *elicit-discrepancy*, for example, points at a discrepancy between the student's answer and the expected answer. It can be used in place of all other active hint categories.

5.1.2. Student Answer Categorization

In this section, we present a scheme for categorizing student answers (Tsovaltzi and Fiedler, 2003).

Proof Step Matching The student's answer is evaluated against an expected answer. The *expected answer* is the next proof step in the formal proof chosen by the system.

Parts of Answers and Over-Answering We define the following units relevant to the student answer categorization: A *part* is a premise, a conclusion, or an inference rule of a proof step. The two former must be explicitly mentioned for the proof step to be complete. An inference rule can either be referred to nominally or represented as a formula itself. In the latter case, we consider that formula as one of the premises. A *formula* is a higher-order predicate logic formula. Every symbol defined in the logic is a function. Formulae can consist of subformulae to an arbitrary degree of embedding. *Constants* are 0-ary functions.

We consider (accurate or inaccurate) over-answering as several distinct answers. If the student's answer has more than one proof step, we consider the steps as multiple answers and apply the categorization to each. There are cases where the order of presentation of the multiple answers is crucial. For example, we cannot count a correct answer that is inferred from a previous wrong answer since the correct answer would follow from a wrong premise.

Completeness vs. Accuracy An answer is *complete* if and only if all parts of the expected answer are mentioned. A part is *accurate* if and only if its propositional content is the true and expected one.

In categorizing an answer, we distinguish between getting the expected domain object right and instantiating it correctly. The latter does not follow from the former. Completeness is concerned with the object's presence, but not with its correct instantiation. That is, a place-holder for an

expected object in the answer is enough for attributing completeness, no matter if the object itself is the expected one. That issue is dealt with by accuracy, a binary predicate used in the same way as completeness. We intend to extend our categorization to include different degrees of accuracy.

The Categories The categories of students answers are:

Correct: A both complete and accurate answer.

Complete-Partially-Accurate: An answer which is complete, but some parts in it are inaccurate.

Complete-Inaccurate: An answer which is complete, but all parts in it are inaccurate.

Incomplete-Accurate: An answer which is incomplete, but all present parts are accurate.

Incomplete-Partially-Accurate: An incomplete answer with some inaccurate parts.

Wrong: An incomplete and inaccurate answer.

The following example shows the tutoring-related annotations. The tutor turns are annotated with the hint categories, the student turns with the student answer categories.

Tutor (1): *Please show: If $A \subseteq K(B)$, then $B \subseteq K(A)$!*

Student (1): (`wrong`) $A \subseteq B$

Tutor (2): (`give-away-relevant-concept`) *That is not correct! First you have to consider the if-then-relation.*

Student (2): (`wrong`) $A \subseteq K(K(A))$

Tutor (3): (`elaborate-domain-object`) *That is correct, but at the moment not interesting. Do you know how to deal with the if-then-relation?*

6. Conclusions and Further Research

In this paper, we presented a corpus of tutorial dialogs on mathematical problem solving, and a corpus annotation that is guiding the development of system components.

Besides the semantic sentence-level and the dialog annotation, we are planning to annotate other dialog aspects. At the discourse level, we are planning to annotate co-reference phenomena. We have observed that the same literals used for mathematical objects may not be co-referring. In the example below, the discourse entities for A_i and A_k do not co-refer:

> DeMorgan-Regel-2 means: $K(A_i \cap B_j) = K(A_i) \cup K(B_j)$ In this case: e.g. $K(A_i)$ = the term $K(A_k \cup B_l)$ $K(B_j)$ = the term $K(C \cup D)$

We are also preparing annotation of interpreted "sense" of utterances and inter-sentential rhetorical relations.

The presented dialog move taxonomy separates general dialog management attributes from genre and domain specific ones hence allowing for a reusable and reconfigurable DM. The annotation will serve to extract a dialog model for tutorial and genre independent dialog phenomena.

Finally, we plan to annotate the following domain concepts: proof steps and their justifications (i.e., inference method used and its parameters), premises and conclusions, and the direction of reasoning (forward or backward), as well as the theorem prover input representation. By looking at rhetorical relations and the above domain concepts, we hope to get insights into the structure of proofs viewed as rhetorical arguments on the one hand and as logical inference objects on the other.

7. References

Allen, J. and Core, M. 1997. Draft of DAMSL: Dialogue act markup in several layers. *DRI: Discourse Research Initiative*, University of Pennsylvania, PA.

Baldridge, J.M. and Kruijff G-J.M. 2002. Coupling CCG with hybrid logic dependency semantics. In *Proc. of the 40th Meeting of the ACL*, Philadelphia, PA.

Baldridge, J.M. 2002. *Lexically Specified Derivational Control in Combinatory Categorial Grammar*. Ph.D. thesis, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh, Edinburgh.

Benzmüller, Ch., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Pinkal, M, Siekmann, J., Tsovaltzi, D., Vo, B.Q., and Wolska, M. 2003a. Tutorial dialogs on mathematical proofs. In *Proc. of the IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, Acapulco.

Benzmüller, Ch., Fiedler, A., Gabsdil, M., Horacek, H., Kruijff-Korbayová, I., Pinkal, M, Siekmann, J., Tsovaltzi, D., Vo, B.Q., and Wolska, M. 2003b. A Wizard-of-Oz experiment for tutorial dialogues in mathematics. In *Proc. of AIED2003*, vol. 8: Advanced Technologies for Mathematics Education, Sydney.

Core, M.G. and Allen, J.F. 1993. Coding dialogues with DAMSL annotation scheme. In *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, MA.

Fiedler, A. and Gabsdil, M. 2002. Supporting pogressive refinement of Wizard-of-Oz experiments. In *Proc. of the ITS Workshop on Empirical Methods for Tutorial Dialogue Systems*, San Sebastián.

Fiedler, A. and Tsovaltzi, D. 2003. An approach to automating hinting in an intelligent tutorial system. In *Proc. of the IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, Acapulco.

Hajičová, E., Panevová, J., and Sgall, P. 2000. A manual for tectogrammatical tagging of the Prague Dependency Treebank. TR-2000-09, Charles University, Prague.

Kruijff, G-J.M. 2001. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. Ph.D. thesis, ÚFAL, Faculty of Mathematics and Physics, Charles University, Prague.

Moore, J. 1993. What makes human explanations effective? In *Proc. of the 15th Annual Conference of the Cognitive Science Society*, Hillsdale, NJ.

Müller, C. and Strube, M. 2003. Multi-level annotation in mmax. In *Proc. of the 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo.

Sgall, P., Hajičová, E., and Panevová, J. 1986. *The meaning of the sentence in its semantic and pragmatic aspects*. Reidel Publishing Company, Dordrecht.

Siekmann, J., Benzmüller, C., Brezhnev, V., Cheikhrouhou, L., Fiedler, A., Franke, A., Horacek, H., Kohlhase, M., Meier, A., Melis, E., Moschner, M., Normann, I., Pollet, M., Sorge, V., Ullrich, C., Wirth, C-P. and Zimmer, J. 2002. Proof development with ΩMEGA. In Voronkov, A. (ed.), *Automated Deduction — CADE-18*, number 2392 in LNAI, Springer Verlag.

Traum, D. and Larsson, S. 2003. The information state approach to dialogue management. In van Kuppevelt, J. and Smith, R., (eds.), *Current and New Directions in Discourse and Dialogue*. Kluwer.

Tsovaltzi, D. and Fiedler, A. 2003. An approach to facilitating reflection in a mathematics tutoring system. In *Proc. of AIED Workshop on Learner Modelling for Reflection*, Sydney.

Tsovaltzi, D. and Karagjosova, E. 2004. A dialogue move taxonomy for tutorial dialogues. In *Proc. of 5th SIGdial Workshop on Discourse and Dialogue*, Boston. To appear.