Data-driven Modeling of Human Tutoring in Calculus

Myroslava O. Dzikovska and **David Reitter** and **Johanna D. Moore** [1] and **Claus Zinn** [2]

Abstract. We describe results from analyzing a corpus of human-human tutorial dialogue, which are aimed at identifying, formalizing and automating natural tutoring in the domain of calculus, in particular symbolic differentiation. We analyzed a corpus of human-human tutoring dialogues, focusing on three areas important to system development: interleaved symbolic and natural language, domain modeling, and tutorial dialogue strategies. We provide empirical verification of previous results on interleaving natural and symbolic language, and show that the properties of interleaving are highly dependent on the input modality. We describe a task model for our domain, and provide corpus data to show that the model must cover basic algebra skills as well, which are involved in differentiation. We verify the applicability of an existing annotation scheme for tutoring algebra in our domain, and propose that it be extended to cover student initiative.

1 Introduction

This paper provides an analysis and discussion of language and dialogue phenomena involved in tutoring mathematics, aimed at building BEEDIFF, a tutorial dialogue system in symbolic differentiation. In this case study, we assume that implementing tutorial dialogue requires modeling of at least three different aspects of dialogue: *conversational expertise* (describing the language students and tutors use), *domain expertise* (describing the reasoning strategies necessary for interpreting student input and generating intelligent feedback), and *pedagogical expertise* (describing tutorial strategies that tutors apply to help students learn). There has been a substantial amount of research in these three areas (e.g. [2, 18, 15, 5]), but it is not clear how well results from one tutoring domain can transfer in a different one.

We show how the analysis of a sample of tutorial dialogue in the domain at hand can provide unexpected insights into the nature of the domain and heavily influence decisions taken when implementing a tutoring system. We focus on studying the following phenomena in our corpus: the role of interleaving natural and symbolic language, frequency of referring expressions, a task model and the contribution of different subtasks to the dialogue, and a high-level model of tutoring moves.

Previously, Wolska and Kruijff-Korbayová [18] identified interleaved natural and symbolic language, various reference phenomena, and informal or imprecise natural language descriptions as key challenges in interpreting language in a corpus of tutoring algebraic proofs. The corpus study presented here analyzes these phenomena quantitatively. We show, in particular, that natural language is used much more frequently by tutors than by the students, and that the interleaving of natural and symbolic language is heavily dependent on the interface modality. Thus, additional analysis is needed before deciding whether and how to support interleaved language and mathematical input in a particular tutoring domain. We discuss the factors and possible solutions in Section 6.

Our analysis of subtasks involved in tutoring symbolic differentiation provides evidence that basic algebra skills, especially simplification of expressions, are of major importance in tutoring differentiation. While this is not particularly surprising, it was not obvious to us in the beginning of the study. Moreover, it is not taken into account in some e-learning systems which use computer algebra systems to support tutoring mathematics (for example, in interactive exercises in the LeActiveMath system [8]). Our study underscores the importance of using human data to appropriately design both tutoring and e-learning systems, and provides an example of how corpus analysis can be used to support domain modeling.

Finally, our study provides evidence demonstrating that an annotation scheme devised for tutoring algebra problem-solving [15] can be transferred to a different mathematical domain. We describe our corpus analysis to establish the applicability of the scheme, and also show where extension is needed, specifically, in accounting for student initiative. We propose a coarse-level classification of student utterances which can be further refined in order to be incorporated into this model.

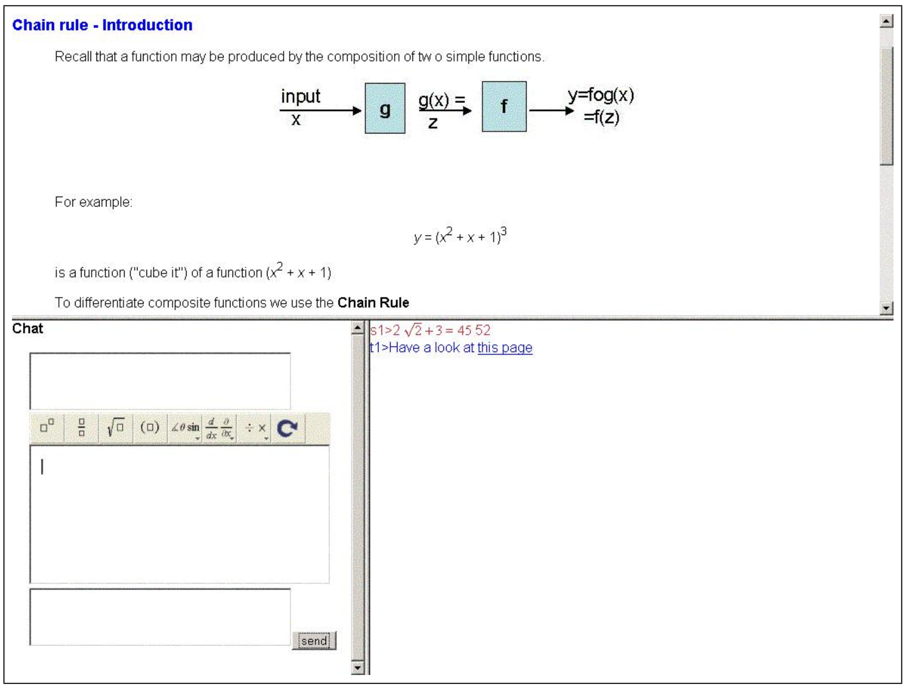
The remainder of the paper is structured as follows. In Section 2, we first describe our data collection. In the three following sections, we report on the results of our corpus analysis with respect to the language used by tutors and students (Section 3), domain modeling (Section 4), and tutorial dialogue modeling (Section 5). We discuss our findings in Section 6, and follow with a discussion of future work and evaluation based on our analysis in Section 7.

2 Data Collection

Our data collection environment separated tutees (henceforth: students) from tutors physically, and thus restricted their modes of communication. They could only interact using natural language via a chat interface where interlocutors could send each other text messages entered via a PC keyboard and use a special editor to enter complex mathematical expressions. Text and formulas could be intermixed. The messages sent by both parties were logged for later evaluation. The student's screen is shown in Figure 1.

The tutor could observe the student's actions in real-time on a second computer screen. Students and tutors were trained to use the interfaces prior to the data collection session.

[1] School of Informatics, University of Edinburgh, email:
`{ mdzikovs | dreitter | jmoore } @inf.ed.ac.uk`
[2] DFKI - German Research Centre of Artificial Intelligence, Saarbruecken, Germany, email: `Claus.Zinn@dfki.de`


Chain rule - Introduction

Recall that a function may be produced by the composition of tw o simple functions.

For example:

$y = (x^2 + x + 1)^3$

is a function ("cube it") of a function $(x^2 + x + 1)$

To differentiate composite functions we use the **Chain Rule**

Chat

s1>2 $\sqrt{2}$ + 3 = 45 52

t1>Have a look at this page

Figure 1. Data collection environment, student's screen.

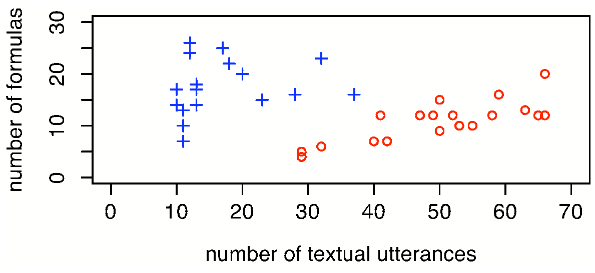


Figure 2. Types of student (crosses) and tutor (circles) utterances per dialogue.

Tutors helped students solve problems in symbolic differentiation, a domain with which participants have been made familiar in class. Students were able to see web pages containing previously prepared basic explanations of the rules of differentiation upon the tutor's request.

Two experienced mathematics instructors (as tutors) and 14 first-year mathematics or science undergraduate students of University of Edinburgh (as tutees) were paid to participate. The resulting corpus consists of 19 dialogues. It contains 1650 utterances (with textual and symbolic parts), 5447 tokens and 559 formulas.

3 Analyzing Mixed Language

3.1 Quantitative analysis of dialogue

As a first step in the exploratory analysis of our data, we were interested in the proportions of the dialogue attributed to tutors and students, i.e., who is talking and how long their utterances are. Such parameters influence the focus of a dialogue system implementation.

Figure 2 shows the number of formulas and textual utterances, separately for students and tutors, in each of the 19 sessions. Tutors, steering the dialogue, dominate the verbal discourse quantitatively, yet they let the students do the mathematical work. Students produced fewer utterances (mean $\mu = 16.6$ utt/session) than tutors ($\mu = 47.3$) but entered more formulas per session ($\mu = 17.4$) than tutors ($\mu = 10.2$).

Students' formulas are also significantly longer ($\mu = 13.8$ elements/formula) than tutors' ($\mu = 9.8$, $p < 0.001$). This is likely to be due to the problems that the students had to solve: the derivative of a term X tends to be longer than X.[3]

To investigate how participants combined mathematical and natural language, we looked at the use of inline mathematical notation, where formulas were written using a semi-standard textual notation rather than the dedicated formula editor provided. We marked up complete formulas such as `cos'(x) = sin(x)` as *inline* formulas, and also individual mathematical terms such as w used in sentences. Tutors wrote inline formulas more often than students did: in addition to 559 formulas written with the formula editor, we found 118 formulas written in textual form inline, of which 25 were written by the students and 93 by tutors, a reliable difference ($\chi^2 = 69.5,\ p < 0.001$).

The results show that our math input editor, though present, was only used when convenient. Even though tutors had more practice with it than students, they did not switch to the formula editor as often: one third of all formulas entered by tutors were entered by keyboard. A reason for this might be that tutors felt pressed to respond quickly and skipped the comparatively inefficient, mouse-driven formula editor. Also, as discussed above, students' formulas are generally more complex than the tutors', because tutors are giving students

[3] This analysis only included formulas entered with the math editor because we do not yet have a parser for formulas written inline (discussed below).

feedback and hints rather than giving step-by-step answers with complex structure, as students do.

The choice of input (formula editor or inline formula) has further effects for mathematical input, as discussed in the next section.

3.2 Linguistic realization of multimodal input

From a linguistic perspective, we were interested in the extent to which interlocutors integrated mathematical formulas into their language. This, again, has important consequences for a resulting dialogue system, impacting the analysis of linguistic input, and the (multi-modal) generation.

We found symbolic expressions were intermixed often with natural language, either as noun phrases or clauses, with varying degree of verbalization of mathematical content:

(1) the 6 is also multiplying the $\cos(6x - 2)$
(2) yes, $\sin'(x) = cos(x)$
(3) Remember that the derivative of $\sin$ is $\cos$

The fact that $\sin'(x) = cos(x)$ holds, for instance, can be expressed symbolically (2), or as an English sentence with words referring to mathematical operators and functions (3). These observations are compatible with results of a prior study in the domain of set theory proofs [18].

We found, however, that the choice of the input method had knock-on effects for linguistic integration. Through the intermixing of input methods (formula editor and text input), the dialogues were inherently multimodal, and users preferred not to integrate the modalities Formulas input with the special editor were most often standalone, even though (as seen in Figure 1) a student or a tutor could have easily continued the sentence after the formula. Instead, we observed that they used deictic pronouns to refer to the content of the formula input field:

Tutor: Try differentiating this example:
$y = (x^3 - 3x)^5$
Student: $5(x^3 - 3x)^4 \times 3x^2 - 3$
is this still the same thing?

Symbolic expressions written with the math editor were more likely to be standalone rather than integrated with an utterance. Formulas composed with the dedicated editor were integrated in a natural language utterance in 186 cases, compared to 368 cases where they were not. In contrast, keyboard-composed formulas and numerical expressions were typically integrated with natural language (130 utterances, compared to 6 standalone expressions). The correlation between linguistic integration and input modality is statistically reliable ($\chi^2 = 166.68, p < 0.0001$). We believe that this contrast between non-integrated multi-modal input with the formula editor and integrated language input with textual notation has implications for both interface design and development of interpretation components for tutorial systems in the domain of mathematics. We discuss those in more detail in Section 6.

To further study how formulas and informal descriptions of mathematical terms are used in our data, we first marked up domain-relevant utterances, defined as utterances which contribute to the task of solving the differentiation problem, or questions and answers relevant to the topic. These included both utterances which used symbolic math directly, and utterances where mathematical concepts were expressed in natural language, *e.g.* , *What is the rule for roots?*.[4]

[4] This excluded idle chit-chat, as well as greetings, acknowledgments, and feedback without explicit mathematical content.

There were 904 domain-relevant utterances in the annotated data, evenly distributed between tutors (468) and students (436).

The student utterances frequently contained only a formula which contributed to the solution. 330 (76%) student utterances consistent of only a formula. Out of the remaining 106 utterances with natural language, 59 (56%) contained only language, and 49 (43%) mixed language and symbolic expressions. For tutors, 425 out of 468 utterances (91%) contained natural language, and 247 (58%) of those contained mixed natural language and formulas. Thus interleaved language and symbolic expressions were used by both students and tutors, but students tended to use significantly less language in general.

As noted above, formulas represent common anchors for referring expressions. In addition to deictic references which point to the input field content, interlocutors often used referring expressions to pick out parts of formulas:[5]

(4) OK. The expression for z is right but do you really mean $\frac{dz}{dy}$ or $\frac{dy}{dz}$?
(5) Multiply right bracket by 5, not left

In the 106 student utterances we annotated as domain-relevant which also contained natural language, 36 (34%) contained a reference to a part of a previous formula. For tutors, out of 425 utterances containing language, 138 (32%) contained references to parts of other formulas. Thus referential expressions are used frequently by both students and tutors. This highlights the importance of handling reference in both interpretation and generation, discussed further in Section 6.

4 Domain Modeling

4.1 Differentiation

The students were tutored primarily in the application of the Chain Rule, defined as follows:

$$\frac{d}{dx}\left[f(g(x))\right] = \frac{d}{dg}\left[f(g(x))\right] \cdot \frac{d}{dx}\left[g(x)\right]$$

Its application involves several steps, which can be formalized as a *task model*. The task model, which describes the task of building a derivative in terms of simpler, partially ordered sub-tasks, helps to identify which parts of the task a student is currently tackling, and how tutors address student actions to provide (corrective) feedback or scaffolding help in case a student is stuck.

The task model for derivatives for a given function $y = f(g(x))$ is as follows: (1) Rewrite y to recognizable form $\overline{y}$ (equivalence transformations); (2) Identify $\overline{y}$ to be of form $f(g(x))$; (3) Identify "inner" ($z = g(x)$) and "outer" ($w = f(z)$) layers of $\overline{y}$; (4) Differentiate each of the layers: compute derivative of z, $\frac{dz}{dx}$, and of w, $\frac{dw}{dz}$; (5) Combine results $\frac{dy}{dx} = \frac{dw}{dz} \cdot \frac{dz}{dx}$; and (6) Normalize result (equivalence transformations) to bring the function into a canonical form.

To obtain a better picture on sub-task distribution in our corpus, we developed an annotation scheme which is a simplification of our task model, and which divides the task into applying the differentiation rules, and normalizing the result. We then segmented our dialogues according to the following scheme:

[5] Users also frequently named pointing to parts of formulas as a desirable feature in post-hoc interviews.

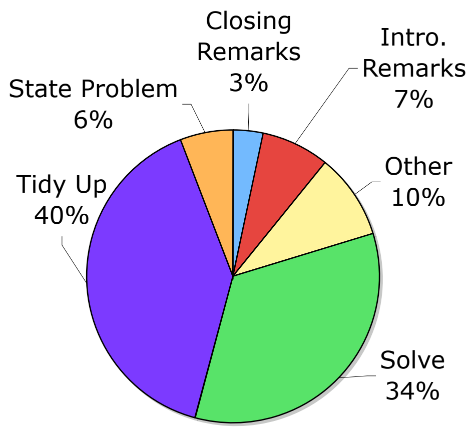


Figure 3. Distribution of the different task segments

Intro. Remarks: A sequence of tutor and student utterances at the beginning of a session, *e.g.*, social, student assessment. Similar remarks at intermediate points in the dialogue were labeled as *Other*.

State Problem: A sequence of tutor turns presenting the problem or exercise the student is supposed to solve.

Solve: A sequence of turns that starts after *State Problem*, and that attempts to construct the problem's solution. This includes student intermediate steps, proposed solutions, tutorial hints, and tutorial feedback on student actions. *Solve* may also include a transformation of the initial problem statement into a form that allows the application of derivation rules; for example, to transform $\frac{3}{\sqrt{x}}$ into $3x^{-1/2}$.

TidyUp: A sequence of student and tutor turns that are devoted to "tidying-up", *i.e.*, *simplifying*, or *normalizing* a given solution. This segment usually follows a *Solve* segment, but is optional in some cases (either because the student presented a reasonable term as part of *Solve*, or the tutor does not insist on term normalization).

Closing Remarks: A set of tutor and student utterances to end a tutorial session.

Other: All other utterances that do not fall into any of the above categories.

The results of this annotation are presented in Figure 3, which shows that term normalization (*TidyUp*) takes up the largest part of the overall dialogue effort, followed by *Solve* turns. This result was unexpected, forcing us to realize that any effective machine-based tutor must be able to teach differentiation *as well as* basic rules of algebraic manipulation (identifying common factors, operator precedence ordering *etc.*). This had direct implications on the design of our domain reasoning components discussed in Section 6.

The particularly large portion of *TidyUp* utterances is an artefact that results from the way differentiation is taught: normalization of equations is an important step, because the normalized formula allows the student to intuitively spot options to further transform the formula (e.g., produce another derivation) and to actually estimate the shape of the curve that the equation models. It also demonstrated that our proposed task model needed to be improved, by differentiating individual steps in the normalization stage as we do in the differentiation stage. We are currently conducting a more fine-grained annotation where all student contributions are marked with the appropriate step in the task model, and which will be used to refine the domain reasoning components.

For the general case of tutoring systems, the finding illustrates that an empirical evaluation of the domain using data collected in a realistic environment helps avoid pitfalls in the implementation of symbolic systems, which occur when formally interesting problems are addressed by the developer instead of the problems that often occur in real-life dialogue.

5 Student and Tutor Dialogue Strategies

So far, we have identified the language and the domain reasoning that tutors use when conducting one-on-one dialogue. Both resources inform the instantiation and selection of tutorial strategies. The goal of our corpus annotation and analysis is to discover strategies used by experienced tutors in our domain.

A number of annotation schemes, for instance, [15, 7, 16, 12, 10], analyze tutorial dialogue at various levels of detail. In the following, we present a discussion of corpus annotation aimed specifically as a data source for building a tutorial dialogue system. We found that McArthur et al. [15] (henceforth McA) was the most useful for our purposes, because this study was conducted in a similar domain (remedial algebra tutoring) and focused on the transformations needed to solve equations. Moreover, the McA scheme is formulated using a notion of a task-model independent of the particular math domain. It provides a detailed catalogue of various tutorial strategies, and is based around a notion of a *current tasks* hierarchy, which fits well with the corpus analysis we discussed in the previous section.

In the McA scheme, a dialogue is structured around a sequence of top-level *solve-problem* tasks divided into smaller steps. A dialogue segment for each subtask is structured around a standard script, usually consisting of a task introduction, performing the task itself, performance assessment by the tutor, and a wrap-up stage. The tutor's decision making is governed by a tutorial planning process, which consists of first selecting a pedagogical purpose to be achieved (*e.g.*, diagnosis or remediation), and then selecting an appropriate tutorial technique to achieve the purpose.

While McA's scheme allows for a rich and detailed analysis of tutorial dialogue, we found that its complexity made it difficult to apply. Thus, as a first step, we made a performed coarse-grained analysis and identified a purpose for each utterance, without referring to McA. Our set of purposes quickly converged toward purposes consistent with McA, thus confirming its validity in our domain.

For tutor's utterances, the utterance purposes from McA we found in our analysis were: *Greeting/Goodbye*, *Task Management* – i.e. choice of problems and transitions between tasks; *Knowledge Assessment* – questions by the tutor aimed at discovering the general state of student's knowledge; *Task Introduction* - various tutor actions in introducing a new problem and relating it to previous problems, and *Performance Assessment* and *Remedial*, which cover the interactions related to problem solving, including giving positive and negative feedback on individual steps, hints and other kinds of remedial strategies.

These purposes covered most but not all of the dialogue. Based on our analysis, we extended the scheme with 3 additional purposes: *Self-Correction*: our tutors made mistakes sometimes and then needed to correct themselves; *Session Management*: since most students did more than one session, tutors discussed where to pick up a thread left off during a previous session, or mentioned that the session is ending soon; and *High-level Performance Assessment*: tutors sometimes gave feedback about the student's overall progress an abilities, which wasn't relevant to the immediate task, as in *"I'm*

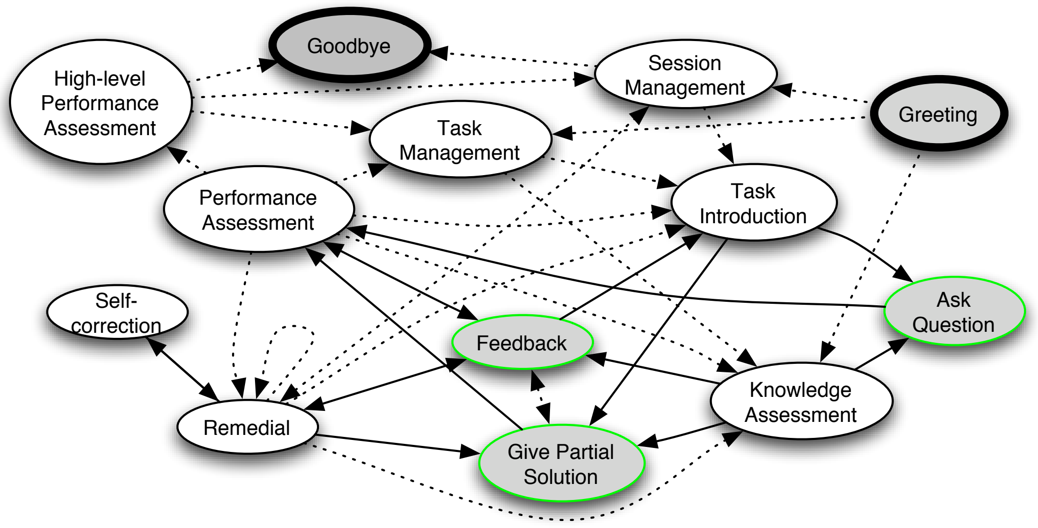


Figure 4. Transition diagram showing the various utterance types found in a subset of our corpus. Dotted transition lines indicate null-transitions, where no input from the student is needed. All other transitions are triggered by interaction. Shaded nodes represent the student's utterance types, others the tutor's ones.

sure you are now an expert in the Chain Rule." We classified those as a separate category, so that a future analysis may help uncover the patterns which lead tutors to give this kind of additional feedback. McA's *recapping* purpose did not appear in our dialogues.

McA provides a detailed classification of student's utterances based on their correctness. However, it assumes that there is little student initiative in dialogue, and all student utterance codes are defined as responses to previous tutor utterances. This was not entirely consistent with our analysis. While in most cases the tutor indeed had initiative, in some cases we found students taking initiative by asking additional questions or stating generalizations. These led to exchanges difficult to account for satisfactorily with McA's classification. Therefore, we developed an initial classification scheme for students' purposes, classifying student utterances into 5 types:

- *Greeting, Goodbye:* (sub-dialogues are contained in one node for simplification).
- *Feedback:* The student informs the tutor whether a principle or a step in the solution has been understood.
- *Give Partial Solution:* The student solves an arbitrarily large part of the problem.
- *Ask Question:* The student takes initiative in steering the tutorial strategy.

Of the above types, only *Give Partial Solution* can be rated in terms of correctness, and we intend to use McA classifications to further refine the annotation of utterances marked with this purpose.

Our annotation shows relationships between tutoring purposes. A bigram analysis of the transitions between categories (Figure 4) can serve as a qualitative model of the tutoring process. It demonstrates communicative conventions and scheme: Neither do tutors terminate the dialogue without giving *Feedback* (and saying *Goodbye*), nor do they jump from a *Greeting* to a *Task Introduction* right away without assessing the student's abilities and/or letting the student know what to expect (*Task Management*).

Once the more coarse-grained analysis was complete, we annotated two dialogues with tutorial strategies based on McA. Consistent with the above analysis, we found that the entire scheme is well applicable to our dialogues, except in places where students took initiative. We also found some examples difficult to classify in terms of McA strategies, often the utterances which performed two functions at once: for example, there were a number of knowledge assessment queries which suggested a correct procedure in addition to querying student knowledge. Extending the scheme to account for those exchanges, and further annotation and analysis of the strategies encountered in our dialogues is part of our ongoing work.

6 Discussion

Although our data was specific to the domain of differentiation, the study points out a range of issues to be faced by builders of tutorial dialogue systems in general.

From the point of view of natural language processing, it is clear that the interleaving of natural language with formulas is a major requirement. Some of our findings confirm the qualitative observations gained on the DIALOG corpus [18] for set theory proofs. However, we provide a more detailed quantitative picture of interleaving symbolic and natural language, and its referential properties. Our analysis shows the difference between how frequently students and tutors used natural language, and we found an effect of the input method on interleaved language not noted in that study, namely that interleaving usually happened when formulas were written in informal "inline" notation rather than with a formula editor.

The difference may be due to the multi-modal nature of our input. The input in the DIALOG study consisted of language that was enhanced with elementary symbols from set theory. Moreover, the math expressions in their corpus could easily be written in one line, in contrast with fractions and exponentials, which are frequent in our corpus, and which require complex formatting to be readable, and cannot easily be laid out in the same line with text. In our case, students and tutors combined formulas and natural-language texts using their keyboards, and often treated the input editor as a separate, non-integrated modality while we originally expected it to be more integrated based on DIALOG study results.

When users are given the option to provide multimodal input, this does not mean that all of the input method modalities are actually used, even if the domain suggests this. Multimodal material is produced by users when it is felt to be appropriate, and with the input method that is the most convenient, as happened in our data collection. System developers cannot ignore the user's choice of modality in the subsequent linguistic processing. This means in our case that the system needs to be able to identify and analyze symbolic language in the textual notation[6] as well as deal with referential input and deixis in combination with the structured input.

The large percentage of mixed language for tutors that occurred in our corpus suggests that it is essential to implement interleaving of natural and symbolic language in generating tutorial feedback. On the input side, however, that most student utterances contained a stand-alone formula, and formulas from the input editor were rarely integrated with language when they were entered together. A useful strategy for domains with complex formulas may be to keep the formula editor separate from student input, used the same way as scratch paper or a blackboard in a human-human tutoring session. The main focus will then be on developing appropriate methods for resolving references to the contents of formula field rather than on interpretation of interleaved formulas and textual input as done in [19]. Introducing an ability to point parts of formulas instead of describing them may also be helpful. This result is also important for building other dialogue systems in the math domain, because it suggests that the complexity of the input changes the nature of the language and the best way to interact with the student.

Since a substantial percentage of both student and tutor utterances (33%) used references to parts of previous formulas, it is important to develop language and reasoning components capable of handling such references. To interpret referential expressions, a declarative model of term structure supporting references such as *the top part* is necessary, similar to [19]. To generate referential expressions, in addition to identifying a buggy rule or an incorrect assumption used by the student, the domain reasoner should be able to precisely describe the location of an error, for example, *There is a small mistake in the denominator*. The latter capability is perhaps more important, given that tutors used significantly more mixed language in our domain.

Additionally, the nature of dialogue may necessitate integration with a theory of task salience. Consider the following fragment:

(6) utt70: student: $15x^2/3(5x^3-6)^4$
(7) utt71: student: or is the 3 still negative?

There are two different instances of 3 in utt70: a power in $5x^3$ and a coefficient in the denominator. This reference was unambiguously understood as the latter, because in the course of problem solving only the denominator changed recently. These types of references were consistently produced and correctly interpreted by both tutors and students. We do not yet have a theory to help disambiguate such referential ambiguity, but one possibility would be to include a salience factor decaying over time for parts of the solution which change, similar to making recently visually changed objects salient in spatial reference resolution [11].

While our application domain is symbolic differentiation, our corpus study showed that a significant number of student errors are due to difficulties with algebraic transformations (*e.g.,* identifying common factors, rewriting square roots as polynomials) rather than difficulties with differentiation per se. In fact, such basic skills were tutored more frequently than differentiation itself (see the task distribution chart in Figure 3). While this may seem obvious, simplifications are not necessarily accounted for in implemented interactive differentiation exercise systems [8]. Moreover, existing computer algebra systems such as PRESS or MAPLE [6, 14] simplify the expressions differently than students are expected to do: for example, they may order the polynomials from the lowest to the highest monomial degree, and they do not perform factorization. Our corpus analysis highlighted teaching expression simplification as an important tutorial goal in the context of symbolic differentiation, and we are currently working on a domain reasoner which includes a model of the algebraic manipulations necessary in simplification. We believe this result may extend to other domains involving algebraic equations.

From the point of view of pedagogical expertise, our contribution lies in evaluating the applicability of McArthur's coding scheme to tutoring symbolic differentiation. We found that the scheme needs to be extended, in particular, to account for student initiative, session management and global performance evaluation performed by tutors in our domain. We proposed three new categories for tutor's purposes, and a classification for student utterance purposes, which will be further refined in ongoing research.

7 Future Work

This analysis is a first step toward building a tutorial dialogue system in the domain of differentiation, and building models of tutorial dialogue. To build the BEEDIFFtutorial dialogue system, we need to consider two separate but related questions. One is how to build computational models of tutorial dialogue which can be implemented in a computer system. Our study contributes to answering this question by quantifying the occurrence of various linguistic and tutoring phenomena in human-human dialogue. We will use the results to determine which features should be modeled in a tutorial dialogue system to make it capable of handling the phenomena common in human-human dialogue.

The question that we have not yet answered with our analysis is the role of the different features we have investigated in learning. Previous studies in conceptual domains demonstrated that the proportion of student language in dialogue is correlated with learning [9], and studies of human-human tutoring generally show effect sizes up 2 standard deviations in comparison effect size of 1 standard deviation achieved by computer tutors [4, 3]. It has been argued previously that producing explanations is important in learning geometry [1]. However, no conclusive evidence about the contribution of student input to learning is available at this point, and investigating the role of student-generated language in learning is an open research question.

A computer system provides a platform for experimentation with features of dialogue that is easier to control than human-human studies. Once the BEEDIFFsystem is built, we can evaluate several features which may be correlated with learning. First, we are planning to compare two versions of the system: a model-tracing version using only a formula input field and a limited set of input buttons such as "OK", "Yes", "No", "Help", and a system supporting full natural language input. This will allow us to determine the contribution of student language to learning (within the limitations of current language understanding technology).

We can also use the system to evaluate the effectiveness of different tutoring strategies, by comparing versions of the system using different strategies. A similar analysis has been conducted by Fiedler

[6] This can be simplified by introducing to users a standard syntax of "inline" expressions which they can use, similar to what we observed in the corpus but clearly formalized to make symbolic expressions easy to identify.

and Tsovaltzi [17], showing differences in learning gains between socratic and didactic strategies in a domain-specific hint ontology.

An appropriate annotation scheme is clearly necessary for evaluating the role of different tutoring strategies in dialogue. The McArthur scheme provides a detailed classification of possible remediations, such as compare this problem to a similar one, state the next goal, or tell the student to try again. Our analysis indicates that this classification is compatible with our domain, though we analyzed most dialogues using only the most coarse-grained level of McA, and more dialogues need to be annotated with the full scheme to verify its reliability.

The McArthur scheme is stated in terms not specific to a given domain (e.g. "state high goal"). This was important in transferring it successfully to our domain, but to implement these strategies in practice, we may need to develop a more detailed domain-specific ontology of goals and concepts, similar to the hinting ontology developed for proof domains [10]. We are planning to further analyze the corpus to better understand how general tutoring strategies of McArthur are realized in our domain as the next step.

The classification of student utterances we proposed is also coarse-grained in keeping with our current corpus analysis. We are working on developing a more detailed annotation scheme describing what students do in our corpus, including different kinds of help queries and explanations of their actions.

8 Conclusion

We presented a corpus analysis aimed at informing the development of a tutorial dialogue system in the domain of differential calculus. We empirically verified the claims of [18] with respect to interleaved language and symbolic expressions. We showed that tutors use significantly more language than students, and that interleaving of natural language and mathematical expressions is highly dependent on input modality, which has to be taken into account in system design. We demonstrated the data-driven development of a task model, which helps determine common student errors, and quantify the time spent on various task stages. We verified the applicability of a tutorial dialogue annotation scheme proposed by [15], and extended it to cover cases of student initiative. Our study demonstrates that even though corpus analysis results with respect to language and tutoring can be transferred from different domains, additional study and extensions to existing schemes may be needed to cover a new domain.

ACKNOWLEDGEMENTS

We would like to thank Kaska Porayska-Pomsta, Manolis Mavrikis and Helen Pain for carrying out the data collection. This work was funded under the 6th Framework Programme of the European Commission, Reference IST-507826.

REFERENCES

[1] V. Aleven, K. R. Koedinger, and K. Cross, 'Tutoring answer explanation fosters learning with understanding', in *Artificial Intelligence in Education, Open Learning Environments: New Computational Technologies to Support Learning, Exploration, and Collaboration, proceedings of AIED-99*, eds., S. P. Lajoie and M. Vivet, pp. 199–206, Amsterdam, (1999).

[2] V. Aleven, O. Popescu, and K. R. Koedinger, 'Towards tutorial dialog to support self-explanation: Adding natural language understanding to a cognitive tutor', in *Proc. AI-ED 2001*, (2001).

[3] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, 'Cognitive tutors: Lessons learned', *The Journal of the Learning Sciences*, **4**(2), 167–207, (1995).

[4] B. S. Bloom, 'The two sigma problem: The search for methods of group instruction as effective as one-to-one tutoring', *Educational Researcher*, **13**, 3–16, (1984).

[5] John Seely Brown and Richard R. Burton, 'Diagnostic models for procedural bugs in basic mathematical skills', *Cognitive Science: A Multidisciplinary Journal*, (1978).

[6] Alan Bundy and Bob Welham, 'Using meta-level inference for selective application of multiple rewrite rule sets in algebraic manipulation.', *Artif. Intell.*, **16**(2), 189–212, (1981).

[7] M. T. H. Chi, S. Siler, H. Jeong, T. Yamau chi, and Robert G. Hausmann, 'Learning from tutoring', *Cognitive Science*, **25**(3), 471–533, (2001).

[8] Arjeh Cohen, Hans Cuypers, Dorina Jibetean, and Mark Spanbroek, 'Interactive learning and mathematical calculus', in *thematical Knowledge Management*, Bremen, (2005).

[9] Mark G. Core, Johanna D. Moore, and Claus Zinn, 'The role of initiative in tutorial dialogue.', in *Proceedings of EACL-03*, (2003).

[10] Armin Fiedler and Dimitra Tsovaltzi, 'Automating hinting in mathematical tutorial dialogue', in *roceedings of the EACL-03 workshop on Dialogue Systems: interaction, adaptation and styles of management*, pp. 45–52, Budapest, Hungary, (2003).

[11] Darren Gergle, 'What's there to talk about? a multi-modal model of referring behavior in the presence of shared visual information', in *Proceedings of the EACL-06 student research workshop*, Trento, Italy, (2006).

[12] S. Katz, 'Peer and student-mentor interaction in a computer-based training environment for electronic fault diagnosis', Technical report, Learning Research and Development Center, (1997).

[13] Maxim Makatchev, Pamela W. Jordan, and Kurt VanLehn, 'Abductive theorem proving for analyzing student explanations to guide feedback in intelligent tutoring systems', *J. Autom. Reasoning*, **32**(3), 187–226, (2004).

[14] Maple user manual.
http://www.maplesoft.com/products/maple/manuals/index.aspx?D=167.

[15] D. McArthur, C. Stasz, and M. Zmuidzinas, 'Tutoring techniques in algebra', *Cognition and Instruction*, **7**, 197–244, (1990).

[16] R. M. Pilkington, 'Analysing edicational discourse: The discount scheme', Technical Report 99/2, Computer Based Learning Unit, University of Leeds, (1999).

[17] Dimitra Tsovaltzi, Armin Fiedler, and Helmut Horacek, 'A multi-dimensional taxonomy for automating hinting', in *Proceedings of Intelligent Tutoring Systems - 6th International Conference*, Lecture Notes in Computer Science. Springer, (2004).

[18] M. Wolska and I. Kruijff-Korbayová, 'Analysis of mixed natural and symbolic language input in mathematical dialogs', in *ACL-2004*, Barcelona, Spain, (2004).

[19] Magdalena Wolska, Ivana Kruijff-Korbayová, and Helmut Horacek, 'Lexical-semantic interpretation of language input in mathematical dialogs', in *Proc. Workshop on Text and Meaning (at ACL-2004)*, (2004).