Running head: GURU: DESIGNING CONVERSATIONS WITH AN EXPERT TUTOR

Guru: Designing a Conversational Expert Intelligent Tutoring System

Andrew M. Olney

Institute for Intelligent Systems

& Department of Psychology

University of Memphis

Natalie K. Person

Department of Psychology

Rhodes College

Arthur C. Graesser

Department of Psychology

University of Memphis

Dr. Andrew McGregor Olney

365 Innovation Drive, Suite 303

Memphis, TN 38152

e-mail: aolney@memphis.edu

Abstract

Intentionally left blank

Guru: Designing a Conversational Expert Intelligent Tutoring System

Identification

There is substantial empirical evidence that one-to-one human tutoring is extremely effective when compared to typical classroom environments (Bloom, 1984; Cohen, Kulik, & Kulik, 1982; Graesser & Person, 1994). Unfortunately, a human tutor cannot be provided to every child because there are simply not enough tutors. However, a technological solution exists: intelligent tutoring systems (ITS), which mimic human tutors, are accessible to anyone with a computer. We have successfully modeled the strategies, actions, and dialogue of novice tutors (Graesser & Person, 1994; Graesser, Person, & Magliano, 1995; Person, Graesser, Magliano, & Kreuz, 1994) in an intelligent tutoring system with learning gains comparable to novice tutors (Graesser et al., 2004; VanLehn et al., 2007). While this progress is significant, Bloom (1984) has reported that accomplished human tutors can produce even greater learning gains than novice human tutors. Building an ITS that mimics the pedagogy of expert human tutors is an ambitious research goal. To address that goal, we are building Guru, an ITS designed to mimic expert human tutors using advanced applied natural language processing techniques.

Investigation

Recently, Person and colleagues have undertaken a rigorous, large scale study of accomplished, expert human tutors. They have recorded fifty expert tutoring sessions, have transcribed the dialogues between tutor and student, and have coded the dialogues on both a micro-level (speech acts) and macro-level (sub-dialogues or tutoring modes). Based on our coding schemes, we have extracted dialogue models from these tutoring

sessions that reflect the general underlying structure of the tutors' conversations on multiple levels (D'Mello, Olney, & Person, in press). These dialogue models are the foundation for our approach to building an ITS because they outline what happens in an expert tutoring conversation. However, because these models are structural they are an incomplete model of expert human tutoring in two ways. First, our structural models do not specify a dialogue move's propositional content or the choice of words within it. For example, our structural models specify dialogue move categories (e.g., `question`, rather than a specific dialogue move such as "What is mitosis.") Secondly, when alternatives are possible our structural models only specify the alternatives but do not indicate which alternative is most appropriate in a situation. For example, our structural models may specify that the next tutor dialogue move should be a hint, prompt, or pump, but selecting amongst them would require assessing a number of other dialogue features (e.g., the correctness of a student's response, the student's overall progress, etc.)

Resolution

In this chapter we describe our ongoing research efforts using our expert human tutor data to create the expert Guru ITS using applied natural language processing (ANLP) techniques, including natural language understanding, knowledge representation, and natural language generation. These ANLP techniques allow us to fill in specific gaps in our structural dialogue models and to create a functioning system. This chapter we will primarily focus on the tools and methodologies behind creating an expert ITS, but our ultimate goal is student learning. We believe that an expert ITS will enhance learning outcomes beyond current ITS technology by using the particular tactics, actions, and dialogue of expert human tutors. Therefore, the essence of our approach is to design conversations between the Guru tutor and the student to promote learning.

Expert Human Tutoring

In order to model the conversation of an expert tutor, a corpus of expert human tutoring is needed. However, the most current meta-analysis reveals that the majority of human tutoring studies reported in peer-reviewed sources have primarily included untrained or "typical" tutors (Cohen et al., 1982). Expert tutoring studies are comparatively scarce, and such studies have included only a handful of expert tutors. In this section we review the studies that are most frequently cited in the literature and note some of the problems that have contributed to our lack of expert tutoring knowledge. First, several studies fail to indicate how many expert tutors were included in the analyses (Aronson, 2002; Fox, 1993; Derry & Lajoie, 1993; Lepper & Woolverton, 2002). Second, although some studies have included five or six expert tutors (Derry & Potts, 1998; Graesser, Person, Harter, et al., 2001; Lepper, Aspinwall, Mumme, & Chabay, 1990; Lepper, Woolverton, Mumme, & Gurtner, 1993; VanLehn et al., 2007), the remaining included only one or two experts (Shah, Evens, Michael, & Rovick, 2002; Evens, Spitkovsky, Boyle, Michael, & Rovick, 1993; Glass, Kim, Evens, Michael, & Rovick, 1999; Lajoie, Faremo, & Wiseman, 2001; Jordan & Siler, 2002). These missing and negligible numbers call into question whether the findings generalize to all expert tutors. Third, many of the studies include the same sample of expert tutors. For example, the tutors included in Graesser et al. (2001), Jordan and Siler (2002), and VanLehn et al. (2007) are the same five tutors. Fourth, a significant number of the studies have focused on the motivational aspects of tutors rather than on the cognitive and pedagogical features that contribute to student learning (e.g., the studies by Mark Lepper and colleagues). A fifth problem with these studies involves the credentials of the experts. That is, it is unclear as to what constitutes an expert tutor. In some of the studies, the expert tutors are Ph.D.s with extensive teaching and/or tutoring experience (Evens et al., 1993; Glass et al., 1999; Graesser et al., 2001; Jordan & Siler, 2002), whereas in others the experts are graduate

students who worked in tutoring centers (Fox, 1993). These are just some of the problems that warranted our recent collection of expert human tutoring data.

Our expert human tutoring corpus is the largest collection of expert tutoring sessions to date. Our expert human tutoring corpus includes twelve expert math and science tutors who were screened carefully and recruited to participate in the project. The focus on math and science reflects the emphasis on STEM (science, technology, engineering, and mathematics) by the U.S. government. All experts had a minimum of five years of one-to-one tutoring experience, a secondary teaching license, a degree in the subject that they tutor, an outstanding reputation with schools as a private tutor, and an effective track record (i.e., students who work with these tutors show marked improvement in the subject areas for which they receive tutoring). The students in our study were all students having difficulty in a science or math course and were either recommended for tutoring by school personnel or sought professional tutoring help. Fifty one-hour tutoring sessions were videotaped, transcribed, and annotated.

Three coding schemes were developed to classify all tutor and student dialogue moves. A dialogue move was either a speech act, an action (e.g., student reads aloud), or a qualitative contribution made by a student (e.g., partial or vague answer). Multiple dialogue moves can occur within one conversational turn. The Tutor Pedagogical Moves scheme includes 14 categories and was inspired by previous tutoring research on pedagogical strategies and dialogue moves (Cromley & Azevedo, 2005; Graesser et al., 1995). The Tutor Motivational Moves scheme includes 8 categories that were either reported previously in the literature or were extrapolated from the INSPIRE model (Lepper & Woolverton, 2002). All tutor moves were classified as either motivational or pedagogical and then assigned to a particular pedagogical or motivational category. A coding scheme was also developed to classify all student dialogue moves into 16 categories. Four trained judges coded the 50 transcripts on the three dialogue moves schemes. To

determine the reliability of their judgments, Cohens Kappas were computed (.96 for Tutor Motivational Scheme; .88 for Tutor Pedagogical Scheme, and .88 for Student Move Scheme). Approximately 57,000 dialogue moves were coded. The Tutor Motivational and Pedagogical Schemes are presented in Table 1 and Table 2.

[TABLE 1 AND TABLE 2 ABOUT HERE]

We also developed a coding scheme for larger units of the tutoring session that we call modes (Cade, Copeland, Person, & D'Mello, 2008). Two trained judges coded the 50 transcripts and found eight modes, including *Introduction, Lecture, Highlighting, Modeling, Scaffolding, Fading, Off Topic,* and *Conclusion*, with Kappa above .80 for each mode. Each mode can be characterized by a specific kind of interaction. For example, *Introduction* contains greetings and establishes an agenda, *Lecture* is predominantly direct instruction, *Highlighting* draws attention to a problem solving step, *Modeling* occurs when the tutor works a problem for the student, during *Scaffolding* the tutor and student solve a problem together, *Fading* involves the student predominantly solving a problem alone, *Off Topic* contains non-tutoring related conversation, and *Conclusion* mirrors *Introduction* as the social glue at the end of the session. An individual mode can span dozens of turns, and so represents a major unit in the structure of a tutoring session.

We have recently analyzed the expert human tutoring corpus using data mining techniques and have discovered significant patterns of dialogue moves (D'Mello et al., in press). In that work we determined two-step transitions, i.e., move to move, that occured at rates significantly greater than chance and had effect sizes greater than the median effect size. In *Lecture*, for example, only 34 transitions met these criteria; a visual inspection revealed several meaningful dialogue move clusters. The first *Lecture* cluster is the information *transmission* cluster, in which the tutor primarily engages in direct instruction with superficial monitoring of student attention and understanding. The second *Lecture* cluster is the information *elicitation* cluster where the tutor elicits

information from the student using direct questioning, e.g. forced choice, prompts, pumps, etc., the student tries to answer, and the tutor gives feedback on the student's answer. Additional *Lecture* clusters include an off-topic cluster, e.g. humor, and a student-initiated questioning cluster, e.g. common ground questions and knowledge deficit questions. Each cluster is essentially a subgraph, ie. a smaller graph contained within the larger graph defining *Lecture*; alternatively, one can view each subgraph as a subdialogue nested in the larger *Lecture* dialogue.

Under the type of analysis performed by D'Mello et al. (in press), the expert human tutoring corpus provides a specification for Guru on multiple levels. On a mode level, we know that sessions typically shift from *Introduction* to *Lecture* to *Scaffolding*. So at the highest level, we can consider tutorial conversations in terms of mode transitions. Within each mode, we can use dialogue move transition information to both extract larger subdialogues (clusters) and to estimate the most probable tutor response to any given student move within a subdialogue. So our analysis of the expert human tutoring corpus has given use three levels of structure: mode, subdialogue, and move. Additionally the expert human tutoring corpus can help us with generating the content of some dialogue moves, since we can inspect the corpus to see the possible ways a particular dialogue move is manifested, e.g. Positive Feedback ("Yes", "Correct", "That's right", etc.).

While the expert human tutoring corpus is a useful description of what expert tutors do, there are many blanks that must be filled in so that the corpus can be used to build a functioning ITS. For example, when a tutor gives feedback, positive and negative feedback are not equally probable. Instead the tutor's response is based on some assessment of the student's knowledge, and the type of feedback is constrained by this assessment. Likewise when tutorial dialogue is generated, it cannot be probabilistically sampled except in a few cases such as unelaborated feedback, e.g., "Good job." Imagine how incoherent the tutor would be if we randomly sampled direct instruction moves from

50 different tutoring sessions! Instead the tutor's dialogue moves, while guided by the mode, subdialogue, and move transitions of the corpus, must be generated from some underlying knowledge representation that reflects the structure of the domain being tutored. In a nutshell, the problem of building a conversational ITS is no less than solving problems of natural language understanding (required to assess the student) and natural language generation (to produce tutor responses), both of which must be mediated by some knowledge representation of the domain (to provide coherent instruction). In the rest of the chapter, we describe our approaches behind these ANLP components for Guru.

Knowledge Representation

We begin our discussion of Guru's ANLP components with knowledge representation. Because both natural language understanding and generation are mediated by a knowledge representation, it is necessary to describe the representation before describing how it is used for understanding and generation. But before that, it is worthwhile to review how knowledge representation is characterized by the intelligent tutoring system community. The task demands of an ITS system, namely modeling expertise and modeling student knowledge, supply constraints on the properties of a useful knowledge representation.

In an ITS, modeling of expertise and modeling of student knowledge are typically called the domain model and the student model, respectively (Beck, Stern, & Haugsjaa, 1996). A domain model represents expertise in a domain, i.e. the relevant knowledge in that domain and its organization, which is often geared either towards problem solving or explanation. While it is beyond the scope of this chapter to describe the many differences among various ITS domain models, what they share in common is that they are typically structured to make tasks in the ITS more straightforward, whether that task is selecting the next problem for the student to solve or assessing the correctness of a student's

answer. Student modeling likewise has many implementations, and each of these implementations has its own underlying representation of the domain (Ohlsson, 1992). For example, overlay student models typically assume a domain decomposition where chunks of content can be marked as understood by the student, rather like checking items off a list. An overlay student model is so called because it lays over the domain model in a rather transparent way, i.e. each element of the domain model is on the checklist for the overlay student model. Overlay models are one of the more popular student models in ITS research, and our current choice of student model for Guru.

Clearly an overlay student model first requires a domain model. However, the creation of a domain model is sufficiently challenging that it requires special authoring tools to accomplish and still requires many man-hours to develop (Murray, 1998; Corbett, 2002; Aleven, McLaren, Sewall, & Koedinger, 2006). Thus in Guru we have been particularly interested in unsupervised and semi-supervised knowledge representation techniques that can extract semantic representations from raw text. The two primary techniques that we have used to date are latent semantic analysis (LSA) and concept map extraction, which we will briefly describe in turn.

Latent semantic analysis (LSA) is a machine learning technique capable of representing world knowledge (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Landauer & Dumais, 1997; Landauer, McNamara, Dennis, & Kintsch, 2007), see also Chapter X this volume. LSA has been shown to closely approximate vocabulary acquisition in children (Landauer & Dumais, 1997), grade undergraduate essays as reliably as graduate students (Foltz, Gilliam, & Kendall, 2000), and understand student contributions in tutorial dialogue (Graesser et al., 2000; Olde, Franceschetti, Karnavat, & Graesser, 2002). LSA works by projecting words into a vector space commonly referred to as an LSA space, which is constructed in a two step process (Martin & Berry, 2007). The first step is the construction of a term-document matrix which contains word frequencies

across a collection of documents. In the term-document matrix, the value at row i column j is the number of times term i appeared in document j. Weighting schemes can further be applied to this matrix to improve task performance (Dumais, 1991). In the second step, the term-document matrix is subjected to singular value decomposition, a common technique in linear algebra. The effect of singular value decomposition is to create a reduced version of the original term-document matrix, in which the first n dimensions of the reduced matrix are an optimal approximation of the original matrix in a least-squares sense (Eckart & Young, 1936).

The resulting LSA space can be used to compare the similarity of two words by comparing the similarity of their corresponding row vectors. The standard comparison metric is cosine, which returns a value between -1 and +1 in theory, but between 0 and 1 in practice (since the original cell counts are positive). Thus LSA returns a cosine value of 1 for identical words and a low non-zero cosine for unrelated words. In a similar way, larger collections of words may be compared to other large collections by summing the corresponding word vectors for each collection. Since the sum of a collection of vectors is another vector of the same dimension, the same cosine metric can be used. Typically in tutorial dialogue, LSA would be used to compare a student's response with an expected response, i.e. an item in the domain model, which would be considered satisfied if the cosine were above a certain threshold. We will elaborate on this approach in the following section on natural language understanding.

Guru's second knowledge representation approach is concept mapping. Concept maps have been used by education, artificial intelligence, and psychological communities for decades, and as a result there are many different kinds of concept maps (Fisher, Wandersee, & Moody, 2000), so what we mean by "concept map" requires some clarification. Generally speaking, a concept map consists of a set of nodes (concepts) and edges (relations) describing a core concept or answering a core question (Novak & Canas,

2006). A given pair of nodes connected by an edge can be called a *triple*, i.e. a start node, edge relation, and end node. Our unique concept map definition combines previous work in both the psychology and education literatures (Graesser & Franklin, 1990; Gordon, Schmierer, & Gill, 1993; Fisher et al., 2000). From the education literature, we adopt a node formulation largely consistent with the SemNet map (Fisher et al., 2000). In our representation, only key terms can be the start of a triple (equivalently the center of a map). End nodes can contain key terms, other words, or complete propositions. This leads to maps with one layer of links radiating out of a core concept. From the psychology literature, we adopt an edge formulation largely consistent with conceptual graphs (Graesser & Franklin, 1990; Gordon et al., 1993). Our representation uses a restricted set of labeled edges that account for a large percentage of relationships. A restricted set is advantageous because having a prescribed set of edges facilitates both generating questions and answering questions from the map (Graesser & Franklin, 1990; Gordon et al., 1993). An example concept map with these specifications is shown in Figure 1.

[FIGURE 1 ABOUT HERE]

We recently developed a procedure for extracting concept maps from a textbook using a semantic parser and related post processing (Olney, in press). We use the LTH SRL Parser (Johansson & Nugues, 2008) to parse the textbook, outputting a dependency parse annotated with semantic roles derived from Propbank (Palmer, Gildea, & Kingsbury, 2005) and Nombank (Meyers et al., 2004). Thus the parse has a wealth of information including part of speech, lemma, head, relation to the head, verbal predicates, nominal predicates, and associated arguments.

Table 3 displays parse output for an example sentence, slightly simplified for length considerations. The root of the sentence is *is*, whose head is token 0 (the implied root token) and whose dependents are *abdomen* and *part*, the subject and predicate, respectively. Predicate `part.01`, being a noun, refers to the Nombank predicate `part`,

roleset 1. This predicate has a single argument of type `A1`, i.e. *theme*, which is the phrase dominated by *of*, i.e. *of an arthropod's body*. Predicate `body.03` refers to Nombank predicate `body`, roleset 3 and also has a single argument of type `A1`, *arthropod*, dominating the phrase *an arthropod's*. Potentially each of these semantic predicates represents a relation, e.g. *has-part*, and the syntactic information in the parse also suggests relations, e.g. `abdomen` *is-a*.

[TABLE 3 ABOUT HERE]

For each syntactic or semantic relation found by the parser, we require that a triple's start node be a key term in our domain. We define these as terms in the glossary and index of the textbook. Depending on the edge type, the edges are either handled syntactically or by using the semantic information returned by Propbank and Nombank. For example, edges that are handled syntactically include *is-a* via the *be* main verb, *has-property* via adjectival modifiers of noun phrases, and *location* via prepositions. Semantic edges derived from Propbank and Nombank require examination of multiple features including the lexical form of the predicate, the gloss for the roleset of the predicate, the label given to the argument, and the gloss given to the argument. These features are input to a manually designed decision tree, which inspects the features by priority and assigns a relation. We describe this process as semi-supervised because the node and edge definitions have been manually defined for our domain, but the rest of the procedure is unsupervised.

Natural Language Understanding

Now that we have defined our knowledge representations, we can describe how they are used in Guru's ANLP tasks. In Guru, and in an ITS generally, the problem of natural language understanding is to map a student's utterance into a representation aligned with the domain and student models. In this way the ITS can determine both if the student's

input is correct or incorrect and what should be discussed next as a result. In Guru we are using both LSA and conceptual graphs to address these issues. Each has its strengths and weaknesses, which we describe in turn.

As mentioned previously, LSA has been used to assess student contributions in tutorial dialogue (Graesser, Chipman, Haynes, & Olney, 2005; VanLehn et al., 2007; Olney, 2009b). These approaches use a domain model based on a combination of a curriculum script, which must be authored by hand, and LSA. The curriculum script contains all of the domain-dependent dialogue to be spoken by the tutor, as well as expected answers, i.e. correct answers, that a student should say in response to a question. The goal of the tutor is to get the student to produce a multi-part explanation in response to a problem posed by the tutor. Student answers are compared to expected answers using LSA, which returns a score roughly between 0 and 1 indicating the degree of similarity (0 not similar; 1 identical) between the student's answer and the expected answer. The problem posed by the tutor is considered answered when the LSA match between student answers and the expected answers crosses a given threshold, e.g. 0.7.

LSA is a valuable technique as an approximate match mechanism. However, LSA is a bag of words technique, so it cannot handle word order. One way of understanding this is to realize that the sum of vectors does not change when vectors are added in a different order. For a high-precision natural language understanding system, insensitivity to word order is a problem: even a novice human tutor knows that "John likes Mary" and "Mary likes John" have two different meanings. However the cosine between these sentences is 1 (complete similarity).

We have addressed the problem of word order in LSA in our recent work (Olney, 2006, 2007b, 2009a), which has resulted in a new algorithm for creating LSA spaces, i.e. singular value decomposition, within the tradition of algorithms proposed by Cullum and Willoughby (2002). The new algorithm allows for much larger input matrices to be

processed than traditional algorithms used for this process (Berry, 1992). As a result, an LSA-like space can be constructed using a mixture of words and multi-word units (n-grams) which would exceed the memory capacity of a conventional computer. Because the multi-word units are inherently ordered, their vectors may be added in the standard LSA fashion but the underlying order is preserved. This allows us to assess student utterances with higher precision than is possible with traditional LSA, though we also note that there are situations in which word order has a negligible effect (Olney, 2009a; Landauer, 2007).

In addition, we and colleagues have developed an extension to LSA that makes use of an orthonormal basis of LSA vectors (Graesser et al., 2007; Hu, Cai, Wiemer-Hastings, Graesser, & McNamara, 2007; Olney & Cai, 2005a, 2005b; Olney, 2007a). Each vector in the basis is linearly independent of the others. Linear independence means that the orthonormal basis representation preserves its constituent components, the individual word vectors. Although this approach does not explicitly address the word order problem, it does allow LSA to make finer discriminations. As a result, we have used the orthonormal basis technique to segment dialogue, detect entailments, summarize, and understand tutorial dialogue in ways that consistently outperform traditional LSA.

Concept maps offer another way of achieving high precision in natural language understanding. Recall from our previous discussion that a concept map domain model consists of a large set of triples where the start node is a key term in our domain, the edge is from a restricted and pedagogically relevant set of relations, and the end node is an arbitrary proposition. Rather than mix all these components in a single bag of words, concept maps allow us to keep them separate, with pedagogically interesting implications.

With a concept map, when one element of a triple is incorrect in a student's utterance, we can identify what that component is and respond accordingly. First, if only the student's start node is incorrect (and the other elements correct), we can recognize

that the student likely has not adequately discriminated the actual start node of the triple and the start node in their utterance. For example, if the student's utterance contains the triple `white blood cell` *has-consequence* `delivers oxygen`, then we can identify that this student knows something about red blood cells that is being incorrectly generalized to white blood cells. This is quite different from the second case of incorrect edge relation, where the student fails to understand the relation between two items, e.g. `red blood cell` *lacks* `delivers oxygen`. Finally, the student can know that a key term has a proper edge relation, but confuses the proposition related, e.g. `red blood cell` *has-property* `found in plants`, which is not a property of red blood cells.

Depending on the type of error the student makes, we can differentially respond. In the case of an incorrect start node, we can contrast the student's start node with the correct one, in order to teach them finer discrimination. In the second case, we can highlight the relationship that the student has incorrect, explain what that relationship means, and help the student understand the proper relationship. In the final case, it's likely that the student has very little background knowledge on the topic, so we might engage in extended direct instruction. These are just examples of possible strategies, but the point is that with a concept map representation, we can make finer discriminations of what the student's error is and respond more appropriately than we can with a bag of words approach, even an n-gram approach with partial order.

We are currently exploring fusions of LSA with concept maps. There are two reasons to consider this type of hybrid approach. The first is computational efficiency. The LTH parser requires significant memory resources to operate and is difficult to run on a computer with 4 gigabytes or less of main memory. Therefore one hybrid approach would be to use LSA to try to identify the elements of a triple in a student's explanation. This approach would also allow the student to use synonyms to express the same ideas without penalty. The problem with this kind of hybrid is that it sacrifices some of the

increased precision that can be gained from concept maps as described above, since LSA is more capable of checking that the elements of a triple are present in a student utterance than for matching words to specific elements. A second kind of hybrid approach combines the LTH parser with LSA. In this approach the student's utterance is first parsed with the LTH parser. Then each element of the triple, as extracted from the student's utterance, is compared via LSA with the expected triple. This allows the precision of the concept map to be augmented with LSA's flexibility: students can use related words rather than exact words to satisfy the triple. We are currently exploring both approaches.

Natural Language Gneration

In Guru, natural language generation maps a knowledge representation to a tutor's utterance. This is a two step process in which we need to decide not only what category of response to make, e.g. a hint, but also the specific text constituting the hint, as in Table 2. The first step, selection of the move category, is based on our three models of dialogue structure. The second step, generating the text of the move, is based on our domain model and student model.

For the first step, we described the three levels of dialogue model extracted from the expert human tutoring corpus in a previous section, so we only briefly describe how these are incorporated into Guru. In a nutshell, dialogue management performs the function of deciding what a tutor should say next. Of course what is said next is not completely arbitrary; it is dependent on what the tutor and student have been talking about. Typically, this is formulated in terms of planning (Freedman, 1996; Khuwaja, Evens, Rovick, & Michael, 1994; Zinn, Moore, & Core, 2002). Framed as a planning problem, the decision of what to say next depends on finding the sequence of dialogue moves the tutor can make that will best accomplish the goal of student understanding. Our previous dialogue management research (Graesser et al., 2001; Olney, 2009b) leads us to believe

that Prolog, a declarative language that has been widely used for artificial intelligence research and cutting-edge dialogue systems (Bratko, 1986; Larsson & Traum, 2000; Zinn et al., 2002) is an excellent framework for incorporating structural dialogue models into an intelligent tutoring system.

In the second step of our natural language generation process, we must transform a given dialogue move category into speakable text. The two different kinds of knowledge representation we have described, LSA and concept maps, are not equally suited to natural language generation. We first consider LSA, wherein the problem of generating text from a knowledge representation is essentially mapping an LSA vector to some ordered set of words. Typically, an LSA vector representation is one-way, so one cannot reverse engineer a sentence representation to find its constituent words. One way to understand this problem is to consider an arbitrary number, e.g. 42, and the problem of determining what numbers were added together to reach this number. Clearly there are many possible numbers that may be added to reach 42; likewise there are many possible vectors that may be added to reach a particular target vector. Thus generating ordered text from LSA vectors does not seem like a promising approach at this point.

Concept maps, on the other hand, have been previously used to generate text. Our concept map representation is very close to previous work in psychology that uses a fixed set of edge relations (Gordon et al., 1993; Graesser & Franklin, 1990). A particular advantage of limiting relations to a set of categories is that the categories can then be set into correspondence with certain question types, e.g. definitional, causal consequent, and procedural, for both the purposes of answering questions (Graesser & Franklin, 1990) as well as generating them (Gordon et al., 1993). For example, `red blood cell` *has-consequence* `delivers oxygen` can be used to generate the questions "What causes oxygen to be delivered," "What does a red blood cell do," or "What can you say about a red blood cell and oxygen" depending on whether we want to query the start node, the

end node, or the edge relation between them respectively.

A related approach is currently used in Betty's Brain, an ITS in the "learning by teaching" paradigm (Biswas, Schwartz, Leelawong, & Vye, 2005; Leelawong & Biswas, 2008). Students teach an agent, Betty, whose brain is reified as a causal concept map with additional hierarchical (i.e. is-a) and descriptive relations (i.e. has-property). Once created, the concept map can be queried by the student, or even allow Betty to "take" a quiz, using a qualitative reasoning algorithm. In the same way that we describe the generation of questions from concept maps, Betty can describe her reasoning by reading off the relationships in the map. However, a major difference is that we are generating complex questions rather than simply reading off relations. Thus our approach requires greater sophistication with respect to the syntax and morphology of constituent elements.

In order to specifically target start node, edge relation, or end node in our questions, we are using SimpleNLG (Gatt & Reiter, 2009), a Java library for natural language generation tasks. SimpleNLG is representation agnostic, so it is fairly straightforward to use it to generate question forms based on our concept maps. This is done using a different strategy for each of the dialogue moves listed in Table 2. Additionally, we make use of the context free prompts provided by Chi, Siler, Jeong, Yamauchi, and Hausmann (2001), which we slightly modify to create templates for question generation. For example, we can convert Chi et al.'s "Any thoughts about that sentence?" to "Any thoughts about START?", where START is the start node of a triple, e.g. `red blood cell`. This is a useful template for a pump dialogue move. In a similar way, Chi et al.'s context free prompts offer useful templates for the hints, prompts, and pumps in Table 2. However, not all of the moves in Table 2 can be so easily generated. We are currently investigating other strategies for moves like forced choice that rely on traversal of the concept map representation. For example, both red and white blood cells are a type of cell, but only one delivers oxygen. So the forced choice "What delivers oxygen, a red blood cell or a

white blood cell" can be generated by starting at `red blood cell`, traversing the *is-a* link to `cell`, finding `white blood cell` as another type of `cell`, and then confirming that it does not also have the property `delivers oxygen`. Given our current results, the use of concept maps for generating pedagogical questions appears very promising.

Conclusion

This chapter described our previous and ongoing work in the dialogue design of the Guru ITS, based on our analyses of expert human tutorial dialogues. Our approach is corpus-based, driven by our extensive collection of expert human tutoring dialogues. Through data mining techniques, we have extracted models of dialogue structure at multiple grain sizes. These dialogue models are the foundation for our approach to building an intelligent tutoring system because they outline what happens in an expert tutoring conversation. However, because our models only represent a kind of "dialogue syntax" they are an incomplete model of expert human tutoring.

The ANLP techniques for knowledge representation, natural language understanding, and natural language generation that we have presented can be used to fill in specific gaps in our structural models and to create a functioning system. Both LSA and concept maps are useful knowledge representations with strengths and weaknesses for natural language understanding: LSA is more forgiving of student input, while concept maps can finely discriminate both what is correct and incorrect in a student's answer. For natural language generation, on the other hand, concept maps are proving to be much more useful than LSA. Not only can concept maps be used to generate questions aimed at different components of a triple, but they can also be used to generate questions like forced choice that combine multiple triples.

Although we have given an overview of many of Guru's key ANLP components, development of an ITS is a complex task, involving many different kinds of dependencies

that constrain the system. The three major sources of constraints are the expert tutoring corpus, the ANLP components themselves, and the end users. Thus there are many other aspects to the Guru ITS that are beyond the scope of this chapter, such as curriculum development, usability studies, and development methodologies that are equally important in the creation of an intelligent tutoring system. These and other factors are extremely important if the ultimate goal of enhanced learning outcomes is to be realized.

References

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2006). The cognitive tutor authoring tools (CTAT): Preliminary evaluation of efficiency gains. In *Intelligent tutoring systems* (p. 61-70).

Aronson, J. (2002). *Improving academic achievement: Impact of physchological factors on education.* San Diego, CA: Academic Press.

Beck, J., Stern, M., & Haugsjaa, E. (1996). Applications of AI in education. *Crossroads*, *3*(1), 11–15.

Berry, M. W. (1992). Large scale singular value computations. *International Journal of Supercomputer Applications*, *6*(1), 13-49.

Biswas, G., Schwartz, D., Leelawong, K., & Vye, N. (2005, March). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, *19*, 363–392.

Bloom, B. S. (1984, June). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, *13*(6), 4–16.

Bratko, I. (1986). *Prolog programming for artificial intelligence.* Reading, Massachusetts: Addison Wesley Publishing Company.

Cade, W. L., Copeland, J. L., Person, N. K., & D'Mello, S. K. (2008). Dialogue modes in expert tutoring. In *Its '08: Proceedings of the 9th international conference on intelligent tutoring systems* (pp. 470–479). Berlin, Heidelberg: Springer-Verlag.

Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from tutoring. *Cognitive Science*, *25*, 471-533.

Cohen, P. A., Kulik, J. A., & Kulik, C. L. C. (1982). Educational outcomes of tutoring: a meta analysis of findings. *American Educational Research Journal*, *19*, 237-248.

Corbett, A. (2002, November). Cognitive tutor algebra I: Adaptive student modeling in widespread classroom use. In *Technology and assessment: Thinking ahead.*

proceedings from a workshop (p. 50-62). Washington, D.C.: National Academy Press.

Cromley, J. G., & Azevedo, R. (2005). What do reading tutors do? a naturalistic study of more and less experienced tutors in reading. *Discourse Processes*, *40*(2), 83–113.

Cullum, J. K., & Willoughby, R. A. (2002). *Lanczos algorithms for large symmetric eigenvalue computations, volume 1: Theory.* Philadelphia: Society for Industrial and Applied Mathematics.

Deerwester, S. C., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*(6), 391-407. Available from `citeseer.ist.psu.edu/deerwester90indexing.html`

Derry, S. J., & Lajoie, S. P. (1993). *Computers as cognitive tools.* Hillsdale, NJ: Erlbaum.

Derry, S. J., & Potts, M. K. (1998). How tutors model students: A study of personal constructs in adaptive tutoring. *American Educational Research Journal*, *35*(1), 65–99.

D'Mello, S., Olney, A. M., & Person, N. (in press). Mining collaborative patterns in tutorial dialogues. *Journal of Educational Data Mining*.

Dumais, S. (1991). Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, *23*(2), 229-236.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211-218.

Evens, M., Spitkovsky, J., Boyle, P., Michael, J., & Rovick, A. (1993). Synthesizing tutorial dialogues. In *Proceedings of the 15th annual conference of the cognitive science society.* Lawrence Erlbaum Associates.

Fisher, K., Wandersee, J., & Moody, D. (2000). *Mapping biology knowledge.* Kluwer Academic Pub.

Foltz, P., Gilliam, S., & Kendall, S. (2000). Supporting Content-Based Feedback in On-Line Writing Evaluation with LSA. *Interactive Learning Environments*, *8*(2), 111–127.

Fox, B. (1993). *The human tutoring dialogue project.* Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Freedman, R. (1996). *Interaction of Discourse Planning, Instructional Planning, and Dialogue Management in an Interactive Tutoring Systemli.* Unpublished doctoral dissertation, Northwestern University.

Gatt, A., & Reiter, E. (2009). Simplenlg: a realisation engine for practical applications. In *Enlg '09: Proceedings of the 12th european workshop on natural language generation* (pp. 90–93). Morristown, NJ, USA: Association for Computational Linguistics.

Glass, M., Kim, J., Evens, M., Michael, J., & Rovick, A. (1999). Novice vs. expert tutors: A comparison of style. In *Midwest artificial intelligence and cognitive science conference.*

Gordon, S., Schmierer, K., & Gill, R. (1993). Conceptual graph analysis: Knowledge acquisition for instructional system design. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *35*(3), 459–481.

Graesser, A. C., Chipman, P., Haynes, B., & Olney, A. (2005, November). AutoTutor: An intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, *48*(4), 612- 618.

Graesser, A. C., & Franklin, S. P. (1990). Quest: A cognitive model of question answering. *Discourse Processes*, *13*, 279-303.

Graesser, A. C., Louwerse, M. M., McNamara, D., Olney, A., Cai, Z., & Mitchell, H. (2007). Inference generation and cohesion in the construction of situation models: Some connections with computational linguistics. In F. Schmalhofer & C. Perfetti (Eds.), *Higher level language processes in the brain: Inferences and comprehension*

processes (p. 289-310). Mahwah, NJ: Erlbaum.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H., Ventura, M., Olney, A., et al. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, *36*, 180-193.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. *American Educational Research Journal*, *31*(1), 104–137.

Graesser, A. C., Person, N. K., Harter, D., et al. (2001). Teaching tactics and dialog in AutoTutor. *International Journal of Artificial Intelligence in Education*, *12*(3), 257–279.

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, *9*, 1-28.

Graesser, A. C., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Tutoring Research Group, T., & Person, N. (2000). Using Latent Semantic Analysis to Evaluate the Contributions of Students in AutoTutor. *Interactive Learning Environments*, *8*(2), 129–147.

Hu, X., Cai, Z., Wiemer-Hastings, P., Graesser, A. C., & McNamara, D. S. (2007). Strengths, limitations, and extensions of LSA. In T. K. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of latent semantic analysis.* Mahwah, NJ: Lawrence Erlbaum Associates.

Johansson, R., & Nugues, P. (2008). Dependency-based syntactic-semantic analysis with PropBank and NomBank. In *Conll '08: Proceedings of the twelfth conference on computational natural language learning* (pp. 183–187). Morristown, NJ, USA: Association for Computational Linguistics.

Jordan, P., & Siler, S. (2002). Student initiative and questioning strategies in computer-mediated human tutoring dialogues. In *Proceedings of its 2002 workshop on empirical methods for tutorial dialogue systems.* Available from

http://www.pitt.edu/ pjordan/papers/its02-workshop.pdf

Khuwaja, R., Evens, M., Rovick, A., & Michael, J. (1994). Architecture of CIRCSIM-Tutor (v. 3): a smart cardiovascularphysiology tutor. In *Proceedings 1994 ieee seventh symposium on computer-based medical systems* (pp. 158–163).

Lajoie, S., Faremo, S., & Wiseman, J. (2001). Tutoring strategies for effective instruction in internal medicine. *International Journal of Artificial Intelligence and Education*, *12*, 293-309.

Landauer, T. K. (2007). LSA as a theory of meaning. In T. Landauer, D. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of latent semantic analysis* (p. 379-400). Mahweh, New Jersey: Lawrence Earlbaum.

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211-240.

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2007). *Handbook of latent semantic analysis*. Mahwah, New Jersey: Lawrence Erlbaum.

Larsson, S., & Traum, D. (2000). Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering*, 323–340. (Special Issue on Best Practice in Spoken Language Dialogue Systems Engineering)

Leelawong, K., & Biswas, G. (2008). Designing learning by teaching agents: The Betty's Brain system. *Int. J. Artif. Intell. Ed.*, *18*(3), 181–208.

Lepper, M. R., Aspinwall, L. G., Mumme, D. L., & Chabay, R. W. (1990). Self-perception and social-perception processes in tutoring: Subtle social control strategies of expert tutors. In J. M. Olson & M. P. Zanna (Eds.), *Self-inference processes: The ontario symposium* (p. 217-237). Hillsdale, NJ: Erlbaum.

Lepper, M. R., & Woolverton, M. (2002). The wisdom of practice: Lessons learned from the study of highly effective tutors. In J. Aronson (Ed.), (p. 135-158). San Diego,

CA: Academic Press.

Lepper, M. R., Woolverton, M., Mumme, D., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer based tutors. In S. Lajoie & S. Derry (Eds.), *Computers as cognitive tools.* Hillsdale, NJ: Erlbaum.

Martin, D. I., & Berry, M. W. (2007). Mathematical foundations behind latent semantic analysis. In T. K. Landauer, D. S. McNamara, S. Dennis, & W. Kintsch (Eds.), *Handbook of latent semantic analysis* (pp. 35–55). Mahwah, New Jersey: Lawrence Earlbaum.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., et al. (2004, May 2 - May 7). The NomBank project: An interim report. In A. Meyers (Ed.), *Hlt-naacl 2004 workshop: Frontiers in corpus annotation* (pp. 24–31). Boston, Massachusetts, USA: Association for Computational Linguistics.

Murray, T. (1998). Authoring Knowledge-Based tutors: Tools for content, instructional strategy, student model, and interface design. *Journal of the Learning Sciences*, *7*(1), 5.

Novak, J. D., & Canas, A. J. (2006, January). *The theory underlying concept maps and how to construct them* (Tech. Rep.). Institute for Human and Machine Cognition.

Ohlsson, S. (1992). Constraint-based student modelling. *International Journal of Artificial Intelligence in Education*, *3*(4), 429–447.

Olde, B. A., Franceschetti, D., Karnavat, A., & Graesser, A. C. (2002). The right stuff: Do you need to sanitize your corpus when using Latent Semantic Analysis? In *Proceedings of the 24th annual meeting of the cognitive science society* (pp. 708–713). Mahwah, NJ: Erlbaum.

Olney, A. M. (2006). *Unsupervised induction of latent semantic grammars with application to parsing.* Unpublished doctoral dissertation, University of Memphis.

Olney, A. M. (2007a). Dialogue generation for robotic portraits. In *Proceedings of the international joint conference on artificial intelligence 5th workshop on knowledge and reasoning in practical dialogue systems* (p. 15-21). Hyderabad, India.

Olney, A. M. (2007b). Latent semantic grammar induction: Context, projectivity, and prior distributions. In *Proceedings of the second workshop on textgraphs: Graph-based algorithms for natural language processing* (pp. 45–52). Rochester, NY, USA: Association for Computational Linguistics. Available from `http://www.aclweb.org/anthology/W/W07/W07-0207`

Olney, A. M. (2009a). Generalizing latent semantic analysis. In *Ieee international conference on semantic computing* (p. 40-46). Los Alamitos, CA, USA: IEEE Computer Society.

Olney, A. M. (2009b, November 5-7). Gnututor: An open source intelligent tutoring. system based on AutoTutor. In *Proceedings of the 2009 aaai fall symposium on cognitive and metacognitive educational systems* (p. 70-75). Washington, DC: AAAI Press. (FS-09-02)

Olney, A. M. (in press). Extraction of concept maps from textbooks for domain modeling. In *Proceedings of the international conference on intelligent tutoring systems.*

Olney, A. M., & Cai, Z. (2005a). An orthonormal basis for entailment. In *Proceedings of the eighteenth international florida artificial intelligence research society conference* (p. 554-559). Menlo Park, CA: AAAI Press.

Olney, A. M., & Cai, Z. (2005b). An orthonormal basis for topic segmentation in tutorial dialogue. In *Proceedings of the human language technology conference and conference on empirical methods in natural language processing* (p. 971-978). Philadelphia: Association for Computational Linguistics.

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An annotated corpus of semantic roles. *Comput. Linguist.*, *31*(1), 71–106.

Person, N. K., Graesser, A. C., Magliano, J. P., & Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: the role of student questions and answers. *Learning and individual differences*, *6*(2), 205-229.

Shah, F., Evens, M., Michael, J., & Rovick, A. (2002). Classifying Student Initiatives and Tutor Responses in Human Keyboard-to-Keyboard Tutoring Sessions. *Discourse Processes*, *33*(1), 23–52.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, *31*, 3-62.

Zinn, C., Moore, J., & Core, M. (2002). A 3-tier planning architecture for managing tutorial dialogue. In *Its '02: Proceedings of the 6th international conference on intelligent tutoring systems* (pp. 574–584). London: Springer.

Author Note

Andrew Olney may be contacted at email:aolney@memphis.edu

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305A080594 and by the National Science Foundation, through Grant BCS-0826825, to the University of Memphis. The opinions expressed are those of the authors and do not represent views of the Institute or the U.S. Department of Education or the National Science Foundation.

Table 1

Tutor Motivational Moves

Move Category	Example
Attribution Acknowledgment	that's easy
Conversational Ok	alrighty
General Motivational Statement	cause you're such a good student I just enjoy you and are ...
Humor	so you're going to have kids and you go "oh I hope he looks like ...
Negative Feedback	no no no no
Negative Feedback Elaborated	actually no you're gonna have some potential energy there too ...
Neutral Feedback	not quite
Neutral Feedback Elaborated	mm you're thinking of vertical vertical angles and stuff like that
Positive Feedback	very good alright
Positive Feedback Elaborated	very good because everything is on top
Repetition	negative 2
Solidarity Statement	let's do it

Table 2

Tutor Pedagogical Moves

Move Category	Example
Counter Example	not multiply we'll add in the area of the bases right
Comprehension Gauging Question	you see what I'm saying
Direct Instruction/Explanation	so that's your lateral area
Example	so as a male you will undergo meiosis and your gametes will ...
Forced Choice	so if we're trying to simplify are we going bigger or smaller
Hint	but now we're not gonna add this many dots right because now ...
New Problem	let's look at this example here it's called expansion of the third ...
Other	does he give you a time limit
Paraphrase	you take out an r squared and you'd have 4 minus pi
Provide Correct Answer	first outer inner last
Preview	we're going to talk about how atoms ions whatever come ...
Prompt	can we simplify the radical of 9 is simply
Pump	and then what do we do
Simplified Problem	what inside the cell would have an electrical charge
Summary	so that's all there is to it so you got a circular chromosome so...

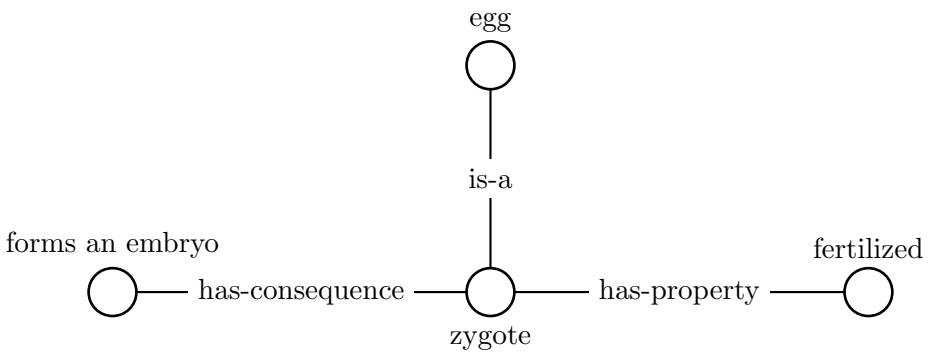
Table 3

A simplified parse

Id	Form	Lemma	POS	Head	Dependency Relation	Predicate	Arg 1	Arg 2
1	abdomen	abdomen	NN	2	SBJ	_	_	_
2	is	be	VBZ	0	ROOT	_	_	_
3	a	_	DT	5	NMOD	_	_	_
4	posterior	posterior	JJ	5	NMOD	_	_	_
5	part	part	NN	2	PRD	part.01	_	_
6	of	_	IN	5	NMOD	_	A1	_
7	an	_	DT	8	NMOD	_	_	_
8	arthropod	arthropod	NN	10	NMOD	_	_	A1
9	s	_	POS	8	SUFFIX	_	_	_
10	body	body	NN	6	PMOD	body.03	_	_
11	.	_	.	2	P	_	_	_

Figure Captions

Figure 1. A concept map radiating from `zygote`


egg
is-a
forms an embryo
has-consequence
zygote
has-property
fertilized