Interaction Between Lexical Semantics and Discourse Planning During Foreign Language Tutoring

Bonnie Dorr and Jim Hendler
Department of Computer Science
University of Maryland
A.V. Williams Building
College Park, Maryland 20742

1 Introduction

The general goal of the tutor under development at the University of Maryland is to allow a wide range of lessons to be developed for use in language training.[1] To support this goal, the natural language processing system must be able to encode lexical, syntactic, and semantic knowledge that is relevant for understanding utterances, whether they are placed in an overt lesson or dialog context. Baseline prerequisites to supporting this aim are: (1) adequate syntactic and lexical representations that have crosslinguistic validity, (2) coverage of the range of constructions that are likely to occur in lessons, and (3) a semantic analyzer and discourse planning component capable of interpreting complex utterances. Our approach to fulfilling this third prerequisite will be the focus of this paper.

Probably the most demanding specific goal of this research is to support language training in a manner as similar to immersion learning as possible. The tutor project centers on the development of a tool which can incorporate the ability for the student to interact with a simulated native language speaker in a relatively unconstrained dialog. In one example of this mode of interaction the system might take the role of a person responding to questions, which the student is asking to the best of his or her ability. Figure 1 illustrates what might be a small segment of the dialog with the simulated Spanish speaker, "Diego Ramirez."

The ability to converse in such an unrestricted manner puts a significant burden on the natural language system. The system must be able to correctly analyze the questions (or statements) the student has produced and it must be able to generate a correct answer. This can require either a simple knowledge-based retrieval (as in the first three sentences of figure 1), or it can require the production of a more complex answer, depending on the question (as in the remaining sentences).

When the student's input cannot be analyzed (due to syntactic or semantic error), the system must produce an appropriate response that can keep the dialog going. For example, in the fourth sentence above, the student, who mistakes the inappropriate cognate "capturada" for the correct term "detenido" asks "Dónde fue la capturada?" (which translate roughly as "where was the captured woman?"). If

[1] The first languages targeted for this system are Arabic and Spanish.

HUMAN>	CUÁL ES SU APELLIDO?
	(What is your name?)
TUTOR>	RAMIREZ.
HUMAN>	SU NOMBRE?
	(Your first name?)
TUTOR>	DIEGO.
HUMAN>	CUÁNTOS AÑOS TIENE USTED?
	(How old are you?)
TUTOR>	YO TENGO 24 AÑOS.
	(I am 24 years old.)
HUMAN>	DÓNDE FUE LA CAPTURADA?
	(Incorrect sentence — Where was the captured woman?)
TUTOR>	NO COMPRENDO.
	(I don't understand.)
HUMAN>	DÓNDE FUE DETENIDO?
	(Where were you captured?)
TUTOR>	DURANTE LA REDADA YO ESTABA DETRÁS LA CASA Y SALÍ CORRIENDO.
	(During the raid, I was in back of the house and ran away.)
HUMAN>	Y ENTONCES?
	(And then?)
TUTOR>	LA POLICIA ME VIO Y FUI ARRESTADO.
	(The police saw me and I was arrested.)

Figure 1: Sample of Small Segment of Dialog with Simulated Spanish Speaker

we assume there is no captured woman the Spanish speaker knows about, he might answer "No comprendo." — "I don't understand you." The student then finds the right word, asks the appropriate question, and the discourse continues.

Efforts to support this type of dialog interaction require the addition of a *planning component* with sophisticated semantic representations that can gauge the semantic appropriateness of student input and structure responses that move the dialog along in an appropriate fashion. These components are currently being added to the already mature syntactic analyzer. Also under implementation is a *generation component* that turns the output of the NLP interpretation into a response phrased appropriately in the foreign language.

One of the types of knowledge that must be captured in the tutor is linguistic knowledge at the level of the lexicon, which covers a wide range of information types such as verbal subcategorization for events (e.g., that a transitive verb such as "hit" occurs with an object noun phrase), featural information (e.g., that the direct object of a verb such as "frighten" is animate), thematic information (e.g., that "John" is the agent in "John hit the ball"), and lexical-semantic information (e.g., spatial verbs such as "throw" are conceptually distinct from verbs of possession such as "give"). By modularizing the lexicon, we treat each information type separately, thus allowing us to vary the degree of dependence on each level so that we can address the question of how much knowledge is necessary for the success of the tutoring system.

The most intricate component of lexical knowledge is the lexical-semantic information, which we encode in the form of Lexical Conceptual Structure (LCS). The basic idea of LCS is that the human language can be modeled using a uniform internal representation of conceptual information with a few parameters that can be toggled one way or another to obtain various languages. The versatility of the LCS allows us to reuse labor-intensive features of the lexicon across languages. We perform natural language analysis by mapping the syntactic representation of the input sentences into this internal representation; the result is used later as the basis for semantic interpretation. This technique produces more accurate analyses than naive syntactic-tree matching approaches, which tend to be much more limited in coverage.

More sophisticated semantic representations that can take information from a knowledge base for the purpose of responding to queries is crucial to run a dialog. In particular, two main types of information must be encoded: information about discourse processing and information about the domain of discourse. Although how to design a formal model of discourse processes is still largely an open question, a number of approaches have shown to be useful when the domain of discourse is limited and controlled.

In the next section, we describe our approach to producing such discourses, and the necessary technology to support them.

2 Approach

To support dialogs such as the one above, the natural language system includes morphological, syntactic, and semantic processing that transforms the student's sentence into a language-independent representation (augmented by the discourse planner as supported by the knowledge representation system); the semantic component then composes a representation that the syntactic/morphological generator transforms into the system's output response. Figure 2 illustrates the overall design of the system.

The syntactic analyzer makes use of Government Binding principles (developed by Chomsky (1981, 1986) and his followers) that are parameterized for the particular language in question. Thus, the analyzer is able to function in both Arabic and Spanish, given the appropriate lexicon. These lexicons are extensible by non-linguists using an editing tool. The parse tree produced by the syntactic analyzer is passed to the lexical-semantic analyzer, and a Lexical Conceptual Structure (LCS) is generated. The LCS's are based on the work of Jackendoff (1983, 1990). They provide a language-free representation of the meaning of the parsed sentence. Details of this representation are reported in Dorr (1992, 1993a, 1993b).

The LCS is then passed to the discourse planner. This planner manages the dialog by choosing the appropriate answers to the student's question in a knowledge base of information about the foreign language speaker and his or her circumstances. The knowledge base is implemented using the PARKA frame language, a knowledge representation system developed at the University of Maryland (see Spector et al. (1990)). This language provides tools for the creation, editing, and querying of

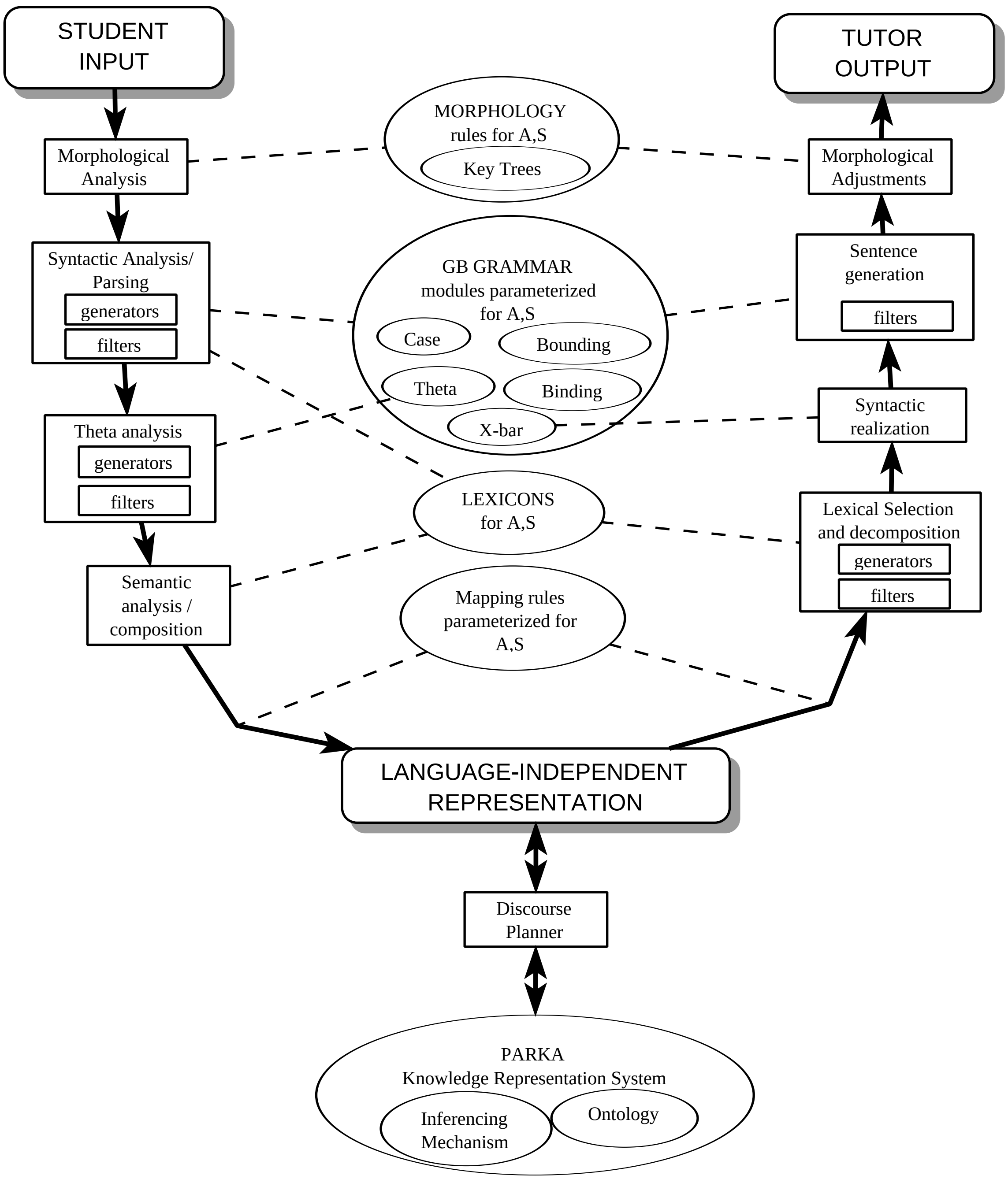

Figure 2: Diagram of Natural Language Processor for Tutor Project

```
(cause (X (HUMAN +))
   (go-poss (REFERENT (PERSON 2) (NUMBER SG) (HUMAN +))
      (to-poss (REFERENT (PERSON 2) (NUMBER SG) (HUMAN +))
         (at-poss (REFERENT (PERSON 2) (NUMBER SG) (HUMAN +))
               (X (HUMAN +)))))
      (location WH-LOCATION)
      (manner FORCEFULLY))
```

Figure 3: LCS Corresponding to DÓNDE FUE DETENIDO?

knowledge bases.[2]

The discourse planner uses PARKA queries in the knowledge base to produce one or more LCS's corresponding to an appropriate answer to the student's question. The LCS is then passed back to the lexical-semantic component, which turns it into a representation that is then passed to the syntactic generator. In addition, the discourse planner passes the generator information which may concern tense, aspect, or particular variable binding information necessary for lexical selection. The result is an annotated parse tree that is then transformed by the generator into the system's output response using lexical knowledge.

As an example, consider how the system would understand and produce the answer to one of the questions in the dialog shown in the previous section. The student types "DÓNDE FUE DETENIDO?" to the tutor, which passes this string to the syntactic parser. The syntactic parser produces a parse tree which recognizes this sentence as a "WH" clause (i.e., a question) with the appropriate syntactic relations among the words noted.

The parse tree is then sent to the lexical-semantic analyzer, which is responsible for producing a semantic structure which unambiguously captures the meaning of the sentence. This structure is created by combining the meanings of the separate words based on the rules of the language in question (encoded via appropriate parameterization) using the features prescribed in the lexicon for guidance. In this case, the various parts of the parse tree are used to produce the LCS in figure 3. (The actual LCS is more complex; we omit some details not relevant to the current discussion.) This roughly corresponds to the question "Where were you when you were caused to be forcefully transferred to someone else's possession?"

This LCS is then passed to the Discourse Planner which attempts to understand it in the context of the current dialog. Based on information about the situation, the system can infer that this is a question about location, that it concerns Diego Ramirez (the persona the system is trying to imitate), and that the question concerns his capture.[3]

Based on this analysis, the discourse planner is invoked. A number of planning

[2] One of the most important aspects of Parka is that it runs extremely efficiently on the massively-parallel Connection Machine. The work in this project uses only the serial implementation. For details about the parallel work (see Evett et al. (1993)).

[3] This is deduced via the use of a discrimination network which matches components of the LCS against target meanings in an efficient manner.

```
OPERATOR-SCHEMA Where-interrogative
  PRECONDITION: FTYPE(Inst-Frame) ≡ WH
                TYPE(Inst-Frame) ≡ Locative
  BIND: Agent ← *Subject
        Mode ← Process-clause[*Internal-Clause]
  CONDITIONS:
    Prev-Estab(Agent,?Location,Mode)
        → Reply(Already-desc)
    Unknown(Agent,?Location,Mode)
        → Reply(Noncommittal,*where*)
    OTHERWISE
        Match-Structure([?Frame,episode = Mode,subj:= = Agent])
```

Figure 4: Operator for Locative questions

operators may be applicable, each one chosen based on pattern matching against appropriate parts of the LCS. For example, questions about *where* something has occurred are handled by the operator shown in figure 4. Roughly translated, this operator states that to answer questions about where something occurred, we need to check if we already have answered this question (if so, generate a reply to this effect). If not, but we don't know where this has occurred, then we need a noncommittal response. Finally, if we do know where this has occurred, then we query for the LCS associated with the episode in question.

This query retrieves a set of frames corresponding to episodes in the knowledge base. Associated with each frame is a corresponding LCS which can be used in producing the answer. Thus, for the example we are following, the answer would be found to be the LCS in figure 5 (where T3 is the time of the raid in the current episode). These representations roughly correspond to "DIEGO was at a location which was behind the HOUSE. He ran from there to an unspecified location."

This LCS information is passed back to the lexical-semantic component which uses this information in the lexicon both to assign words to the variable entries ("Diego" and "House") as well as determining the syntactic structure that best conveys the idea in the appropriate language. This is used to produce a syntactic

```
(be-loc (PERSON *DIEGO*)
   (at-loc (PERSON *DIEGO*)
      (behind (PERSON *DIEGO*) (LOCATION *HOUSE*)))
   (time (T1 overlaps T3)))

(go-loc (PERSON *DIEGO*)
   (away-from-loc (behind (PERSON *DIEGO*) (LOCATION *HOUSE*)))
      (manner RUNNINGLY)
   (time (T2 > T1)))
```

Figure 5: LCS for producing answer

structure which is transformed by the generator into the output response "Yo estaba detrás de la casa." A similar process is used to interpret the latter LCS and produce "Yo salí corriendo." Following this, time information is used to recognize that the running occurred at the same time as the raid, and thus "Durante la redada" is added to the set of sentences to be expressed. Finally a generational rule is used to combine the two into a more grammatical compound sentence and the final answer is produced:

TUTOR> DURANTE LA REDADA YO ESTABA DETRÁS LA CASA
Y SALÍ CORRIENDO.
(During the raid, I was in back of the house and ran away)

The discourse planner is also invoked when the unification algorithm fails to find a query that matches the LCS produced. For example, following the sentence above, the student might type:

HUMAN> Y ENTONCES?
(And then)

In this case, no predicted query matches, and a default rule in the discourse planner is used. To handle such cases, the planner tracks the dialog, keeping account of the context of the discourse. Since the last answer produced focused on running away, this sentence is requesting information about the next temporal event (signaled by "entonces"), and the knowledge base is checked to see what the next event was. Again, an appropriate response such as:

TUTOR> LA POLICIA ME VIO Y FUI ARRESTADO
(The police saw me and I was arrested)

would be produced.

3 Conclusion

In this paper we have described the integration of an LCS-based semantics with a knowledge-based discourse planner for use in imitating natural language dialog for language training. The ability to converse in such an unrestricted manner puts a significant burden on the natural language system. The system must be able to correctly analyze the questions (or statements) the student has produced and it must be able to generate a correct answer. To provide these abilities, our system uses a combination of a syntactic parser, semantic analyzer, and discourse planner.

References

1. Chomsky, Noam A. (1981) *Lectures on Government and Binding*, Foris Publications, Dordrecht, Holland.
2. Chomsky, Noam A. (1986) *Knowledge of Language: Its Nature, Origin and Use*, MIT Press, Cambridge, MA.
3. Dorr, Bonnie J. (1992) "Lexical Semantics for Interlingual Machine Translation," *Machine Translation*, 7:3.

4. Dorr, Bonnie J. (1993b) "Interlingual Machine Translation: A Parameterized Approach," *Artificial Intelligence*, 63:1&2.

5. Dorr, Bonnie J. (1993) *Machine Translation: A View from the Lexicon*, MIT Press, Cambridge, MA, forthcoming.

6. Evett, Matt, James Hendler, and Lee Spector (1993) "Parallel Knowledge Representation on the Connection Machine" *International Journal of Parallel and Distributed Computing,* in press.

7. Jackendoff, Ray S. (1983) *Semantics and Cognition*, MIT Press, Cambridge, MA.

8. Jackendoff, Ray S. (1990) *Semantics Structures*, MIT Press, Cambridge, MA.

9. Spector, Lee, James Hendler, and Matt Evett (1990) *PARKA: Parallel Knowledge Representation on the Connection Machine* UMIACS TR 90-22, CS Dept. TR 2409.