A Cognitive-Based Approach to Learning Integrated Language Components

Charles Hannon and Jonathan Clark

Department of Computer Science,
Texas Christian University,
Fort Worth, TX

Abstract. The learning component of a cognitive-based language model (LEAP) designed to easily integrate into agent systems is presented. Building on the Interlaced Micro-Patterns (IMP) theory and the Alchemy/Goal Mind environment, the LEAP research improves agent-to-human and agent-to-agent communication by incorporating aspects of human language development within the framework of general cognition. Using a corpus of child through youth fiction, we provide evidence that micro-patterns can be used to simultaneously learn a lexicon, syntax, thematic roles and concepts.

1 Introduction

To study language use and learning within a reading task, a robust distributed cognitive model called LEAP (Language Extraction from Arbitrary Prose) and a new working theory of cognition called IMP (Interlaced Micro-Patterns) have been developed as part of the current Alchemy/Goal Mind modeling effort. LEAP examines how language development (at the lexical, syntactic, semantic and conceptual level) occurs within the context of general cognitive development used by all sensory modalities. LEAP differs from most other cognitive modeling efforts of language in that the Alchemy/Goal Mind environment:

- uses process concurrency to increase the amount of processing power available (distributed processing) and simulate the massively parallel processing (MPP) aspects of the brain
- allows models to be built using any number of robust symbolic and non-symbolic inference engines communicating via a generalized messaging structure that simulates neural pathways
- draws a model's explanatory depth from the way components are interrelated and function, not on the underlying theory of their construction.

This paper will lay out the theoretical underpinning of LEAP and discuss how the current results validate both the IMP theory of cognition and the linguistic theory on which it is based.

Hannon C. and Clark J. (2006).
A Cognitive-Based Approach to Learning Integrated Language Components.
In *Proceedings of the 3rd International Workshop on Natural Language Understanding and Cognitive Science*, pages 27-36
DOI: 10.5220/0002503600270036

2 Related Work

This section will address the modeling efforts directly related to LEAP, other general modeling and language capture efforts and an extremely brief overview of the linguistic theory being used in LEAP.

2.1 Directly Related Existing Models

LEAP directly builds upon the TALLUS (Teacher Assisted Language Learning and Use System) model [5], but also draws insight from the STRESS (Stroop Test Responce Evaluation Sub-System) [6] and FEAR [9] models. TALLUS was a cognitive model to study language use and learning in a visual context. It consisted of three agents, a teacher and two students. The utterance level of the model was based on Government and Binding theory, but much of the model dealt with discourse and conceptual processing. The TALLUS model's inability to learn anything past a rudimentary surface-level language was a driving factor in the creation of both the IMP theory and LEAP model.

STRESS used components of the TALLUS model to build a model of the Stroop effect. This model again focused on concept reasoning but did provide some insight into the reading task that has been used in LEAP. FEAR explored the use of emotion to control attention and arousal in an agent system. Much of what was learned from the control structure of this model has been reused in LEAP.

2.2 Related Language Modeling and Capture Efforts

A number of on-going research efforts are addressing the cognitive modeling of language at some level. Many of these models address language within the context of other sensor modalities and are aimed at directly supporting an agent-based application. LEAP attempts to; 1) be explanatory, 2) be closely tied to well known cognitive mechanisms such as priming, spreading activation and memory consolidation, and 3) directly support use of its components within multiagent applications. This makes it similar to models built with SOAR [9], ACT-R [1] and ACT-R/PM [3]; but not restricted to monolithic processes controlled by some underlying cognitive mechanism such as ACT-R's symbolic productions and subsymbolic activations.

In relation to major language capture systems, LEAP's current understanding of language is very small. For example, WordNet contains 144,309 unique words organized into synonym sets representing underlying lexical concepts [4]. The Cyc knowledge base contains almost 2 million assertions (rule and fact), 118,000 concepts and a lexicon of 17,000 English root words [12]. But size does not directly translate to making them useful candidates for knowledge components within an adaptive multiagent application. LEAP is attempting to capture, for use in an agent system, the way humans learn by the slow consolidation of knowledge into a complex and multifaceted representation of their surrounding world.

2.3 Related Language Theory

Most symbolic AI work has been based on Generalized Phrase Structure Grammar (GPSG), Head-Driven Phrase Structure Grammar (HPSG) or Lexical-Functional Grammar (LFG) and relies on one or two pipelines of processing from the incoming morphemes to concepts. It assumes that language understanding is built up from syntactic and semantic structures that must be directly mapped to all constituents of the received utterance before it can be processed. Connectionist approaches have been more pragmatic in what needs to be part of language processing but have not proven to be very extensible.

Using micro-patterns, LEAP is free to examine a wider range of possible linguistic approaches and currently draws some of its ideas from theories as far a field as Relational Grammars and Principles and Parameter Theory. This is possible because Alchemy/Goal Mind provides a distributed AI environment that removes many of the sequential limits imposed of traditional NLP systems.

3 Interlaced Micro-Patterns (IMP) Theory

The Interlaced Micro-Patterns (IMP) cognitive theory extends the traditional pattern matching mechanism by proposing that if a set of simple patterns are interlaced (i.e., allowed to overlap), the mechanism can be used to learn, retrieve and recall elements of far greater complexity, and thus, could be the driving mechanism of such tasks as language use and learning. The support for IMP as a working theory comes from both a set of thought problems and the results of cognitive modeling work.

The TALLUS model failure resulted in the first thought problem. Why do children find it much easier to learn a natural language than the proposed grammar rules that are suggested to define such a language? Hierarchical syntactic approaches to natural language (NL) align well with the way NL grammars are taught in traditional educational settings, but not with how language development naturally occurs. The teaching of prescriptive grammars seldom controls the complete use of either spoken or written language ‘rules’.

So, is there a way to capture the computational strength of generative grammar without it being driven by a hierarchical set of rules? One possible method to do this is to use interlaced micro-patterns. While all possible well-formed utterances conform to some syntactic, semantic and conceptual pattern, the storage of every possible utterance pattern would clearly be too computationally complex to be feasible. However, if all possible sentence patterns were made up of smaller patterns that relied on overlapping elements to ensure correctness, a set of smaller patterns could not only generate correct utterances, but also block the creation of malformed utterances.

3.1 Relationship of IMP to Symbolic AI

We can define a system’s composite Knowledge Representation and Reasoning (KRR) as a set of layered component KRRs with each component’s KRR being any

desired type. This composite KRR can be stored in Long Term Memory (LTM) and access points within each layer can be activated into Short Term Memory (STM) by a pattern input from an external source (either another layer within the agent or an interface to the external world). In addition to the actual access points activated, other parts of the layer's Knowledge Representation (KR) can be activated by a temporal-based spreading activation mechanism when needed and deactivated by removal from the STM when the knowledge is 'timed-out'. Changes to the KRR occur by updating the KR stored in LTM as a result of changes that occur in STM during activation.

A simple formalization of the effect of using IMP to control a layered KRR can be given if we simplify the KR of an agent to a uniform set of semantic networks. Each of these semantic networks can be viewed as a directed multi-graph,

$$R_n = \text{pair}\,(N_n, A_n),\ A_n = \{(\nu_{ni}, \nu_{nj}) \mid \nu_{ni}, \nu_{nj} \in N_n\} \tag{1}$$

where, n is the level of representation, N_n is a set of nodes, and A_n is a bag of named relationships between these nodes. A sub-representation of this network can be defined as,

$$\begin{gathered} R'_n = \text{pair}\,(N'_n, A'_n),\ N'_n \subseteq N_n, \text{ and} \\ A'_n \subseteq A_n \wedge ((\nu_{ni}, \nu_{nj}) \in A'_n \rightarrow \nu_{ni}, \nu_{nj} \in N'_n). \end{gathered} \tag{2}$$

All possible sub-representations at a level n is, of course, the power set of R_n; however, this set has little meaning in the IMP theory since only the activated subrepresentations are of interest. Given all possible activated sub-representations at a level n, defined as,

$$\Phi_n = \{R'_n \mid R'_n \subset R_n \wedge \text{active}(R'_n) \rightarrow \text{True}\}, \tag{3}$$

connections between representation levels can also be viewed as a directed multi-graph,

$$\begin{gathered} K_{i,j} = \text{pair}\,(\Phi_{i,j}, \Gamma_{i,j}),\ \Phi_{i,j} = R'_i \cup R'_j, \text{ and} \\ \Gamma_{i,j} = \{(R'_i, R'_j) \mid R'_i, R'_j \in \Phi_{i,j}\}, \end{gathered} \tag{4}$$

where, i and j are levels of representation being connected and $\Gamma_{i,j}$ is a set of named relationships between these levels.

The number of representation levels (R_n) and number of level connections ($K_{i,j}$) can vary based on application. A traditional agent-based method for using the overall representation structure would be a set of m stacks of representation levels 1 to k with the top-level (level 1) of each stack being an interface representation and the kth level of each stack being either a common conceptual structure or a set of connected conceptual structures.

Given a set of available general inference rules at each level (ρ_n) and between two levels ($\rho_{i,j}$), the extent of general inference at each level (ι_n) and across levels ($\iota_{i,j}$) can be naively described as,

$$\iota_n \approx |\,\rho_n\,| \text{ and } \iota_{i,j} \approx |\,\rho_{i,j}\,|, \tag{5}$$

assuming no serious difference exists in the number of pre and post conditions of each rule. The total extent of representation at each level also can be naively described as,

$$\gamma_n \approx | N_n | \bullet \max \{\nu_{ni}, \nu_{nj}\} d(\nu_{ni}, \nu_{nj}), \tag{6}$$

which given the amount of accessible (or activated) knowledge at each level being $\beta_n = \cup\Phi_n$, leads to an activated representation extent of,

$$\alpha_n \approx | \beta_n | \bullet \max \{\nu_{ni}, \nu_{nj}\} d(\nu_{ni}, \nu_{nj}) \mid \nu_{ni}, \nu_{nj} \in \beta_n \text{ and} \tag{7}$$
$$\alpha_{i,j} \approx | \Phi_{i,j} | \bullet \max \{R'_i, R'_j\} d(R'_i, R'_j).$$

The activation potential at any level can be described as,

$$\eta_n \approx \Sigma_{\{i = 1 \text{ to } k\}} \alpha_{i,j} \bullet \iota_{i,j}, \tag{8}$$

and its inference potential as,

$$\kappa_n \approx \alpha_n \bullet \iota_n . \tag{9}$$

Assuming that pattern matching activation mechanism are only allowed to work between levels, the extent of cross-layer general inference ($\iota_{i,j}$) can be viewed as approaching the value one for all levels. Thus, the activation potential of all levels approximately equals their part of the cross-layer activated representation extent, $\alpha_{i,j}$, which is simply their own activated representation extent α_n. A pattern matching interface between layers reduces the overall inference potential in each layer to a function of the number of activated access points and its own inference extent. To the outside world, any results of a layer's inference engine look like the sum of an Artificial Neural Network's (ANN) forward or backward activation potentials.

4 The LEAP Model Implementation

A Goal Mind model is driven by both explanatory theory and pragmatic computational issues. We first use cognitive science data and theories to map out a general component structure of the cognitive agent, and then, response methods to intra-agent stimuli. This structure is then mapped into Goal Mind processing components called Etherons which are built from the environment's. production system and semantic network libraries and its standard PostgreSQL 'C' language interface. The ability to reuse components between models and subdivide functionality between component clones are the two major controls on cognitive correctness of our research.

As shown in figure 1, the LEAP model is divided into five major subsystems, each of which can be made up of as many Goal Mind processing nodes as required. The current model uses twenty-four Goal Mind components. Message flow in LEAP is done using Goal Mind stimuli allowing components to communicate with each other without understanding the model's complete topology. Control of the model is implemented using Shallow Knowledge Integrated Production System (SKIPS) engines described in [8].

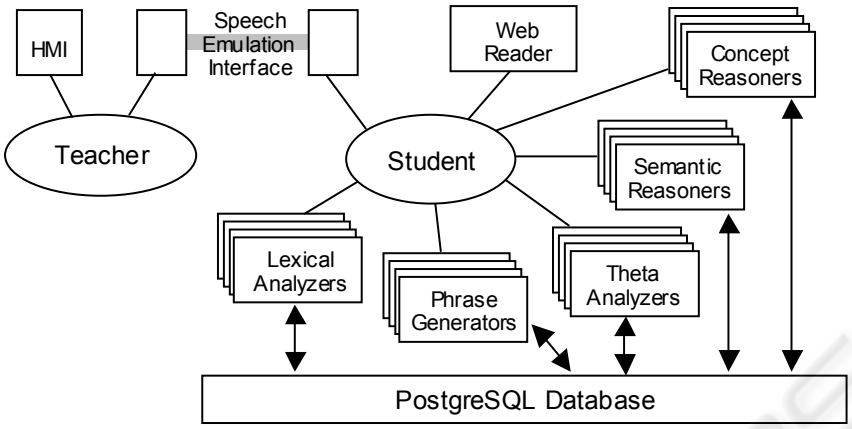

Fig. 1. A simplified view of the LEAP model. The model currently uses eight lexical analyzers and one each of the other four component types, but since it is built on Alchemy/Goal Mind the number of each can be easily changed as needed. All components generate and use micro-patterns and store LTM information in the database.

One way that LEAP controls the amount of language information being processed at any given time is to divide knowledge along the traditional theoretical boundaries of Long Term Memory (LTM) and Short Term Memory (STM) although this is implemented in ways similar to theories (like the one used in ACT-R) that basically reject such a theoretical division. In Goal Mind, the LTM store is a distributed PostgreSQL database. Using standard database normalization (whenever possible) LTM information is stored in compressed form in the database and accessed in small activated chunks. For example, parts of the large LTM semantic network used by LEAP can be uncompressed into a dynamic memory structure within each of the semantic and concept reasoners using spreading activation triggered by input stimuli. This activated STM memory is then 'returned' to LTM using a concept called temporal garbage collection. By using such a STM/LTM knowledge scheme, LEAP can support semantic feedback learning over the level of language knowledge needed for something like Information Retrieval or Extraction without experiencing computational explosion.

The LEAP model tries to process language input from either a teacher agent or a web reader input. This processing is highly parallelized, but can be viewed at an abstract level as being made up of three levels; surface structure, deep structure and conceptual. At the surface level, words of an incoming utterance (either from the web reader or speech emulation interface) are tagged using a set of lexical analyzers and a special purpose Stimuli Routing Network (SRN) used to filter some closed categories. These tags (or PoS) stimuli are used by the phrase generators to create (xP) stimuli containing simple syntactic relationships between words. While the xP stimuli represent a form of tree parsing, the results are far different than a traditional parse. For example, in the noun phrase 'the brown dog of the happy girl from the grand city of Kent', only the information about the direct relationships between nouns and prepositions is extracted from the embedded prepositional phrases. Further, this information

is only used to determine the semantic distance between the verbs and nouns so that the right theta grid can be accessed.

The deep structure processing of an utterance in LEAP does not rely on a single surface level input, nor does it necessarily require (or wait for) a complete surface level 'parse' of the utterance. The semantic reasoners use PoS stimuli to activate the deep structure forms (stored as nodes in a semantic network) related to the surface forms of the utterance. At the same time, theta analyzers use PoS and xP stimuli to propose theta relations (tP stimuli) between words in the utterance. These tP stimuli are tested by matching them to the theta grids of the activated deep structure forms and then following the relational arcs in the activated part of network to see if all of the activated forms can be realized by known relationships. Failure to realize a grid relationship between deep structure forms can cause either a rejection of the related semantic role in the deep structure or the learning of a new semantic role based on the semantic context of the utterance. What happens in each case is determined by the conceptual level of the model.

Like TALLUS, LEAP divides conceptual processing across a number of reasoners. These reasoners handle both higher level language processing like symbol grounding and discourse and provide the mechanisms for connecting the language processing to other cognitive tasks. Since most of our current work with LEAP has been aimed at trying to fix language learning problems with TALLUS, LEAP's set of conceptual reasoners is currently limited to those directly related to the language learning task.

5 Results

LEAP testing was done on a 16-node Beowulf cluster using a web-based corpus of children and young adult stories. This corpus can be found at red.cs.tcu.edu:14321. It contains 12,830 distinct sentences, 12,185 distinct words and total of 301,312 total words. The mean length of sentence for the complete corpus is 23 words. Since the corpus is made up entirely of classic fiction, the grammar is often quite complex.

5.1 Syntactic Learning

The testing method for PoS and xP learning consisted of setting up a baseline database of 20 words and 8 patterns per open category. The model was then allowed to read the complete corpus three times at four different learning rates of 10, 20, 50 and 100. Results of these runs include the incorporation of a number of new words and patterns for the five open categories being studied (i.e., noun, verb, gerund, adjective and adverb). Words learned during a test run were compared against WordNet to determine the accuracy of learning. No semantic feedback was used during these test.

Since the learning of word and patterns are so closely tied (i.e., new pattern generate new words and new words generate new patterns) we use the same learning rate for both words and patterns in each experiment. The percentage of correct nouns learned varies little for a learning rate of 20 or above. However, correct verb and adjective learning improves by increasing the threshold at which new words and

patterns were learned. The current pattern method is not sufficient to learn adverbs. Setting the learning rate too high reduces the rate of learning, a well known machine learning phenomenon. However, there are dangers to setting the learning rate too low if the corpus is not extremely large. At low learning rates, the initial pass through the corpus provides limited supporting evidence for misidentified patterns, but rereading stories tends to reinforce these incorrect assumptions. This again follows from what we know of language development in children where reinforcement of incorrect grammar by example leads to the incorporation of incorrect grammar formations.

The fifth open category learned during the current data selection was that of gerunds. Unlike the other categories, the pattern for these words included the '–ing' morpheme. Gerunds were identified with 100% accuracy at all learning levels, the 79 of them being found at a learning level of 10.

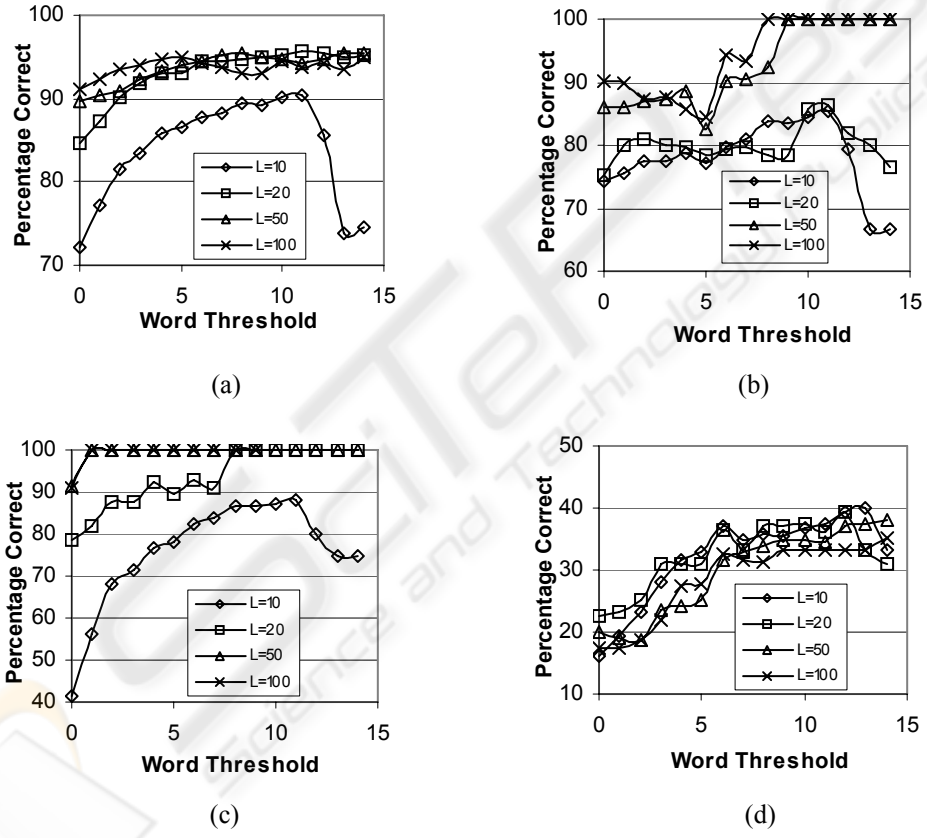

Fig. 2. Percentage of correct detections after three passes through the corpus for (a) nouns, (b) verbs, (c) adjective and (d) adverbs. A pattern learning rate of 10, 20, 50 and 100 is graphed against a threshold of word learning rates (0 to 14).

5.2 Semantic and Conceptual Learning

Once PoS and xP stimuli have been generated, the simple tP generator creates possible theta roles between words in the utterance. If these relationships match existing relations in the semantic information stored within LEAP, semantic feedback information is used to reinforce the PoS information and a conceptual representation of the utterance is activated. If it does not match, then new relationships are proposed in much the same way as new PoS patterns are learned. Learning is currently fairly slow due to the need to restrict this learning to a single unknown relationship utterance and many sentences in the corpus are semantically complex. The theta role learning in LEAP is still being refined but the current system is showing about a 75% accuracy in learning new noun-verb relationships when a learning rate greater than 20 is used.

Conceptual learning is one aspect of the TALLUS model that worked fairly well so we have not, to date, addressed this aspect in LEAP in any great detail. Early tests are showing that concept learning works as well, if not better in LEAP.

6 Future Work

Much more work needs to be done at both the semantic and conceptual levels of LEAP to get it to the level of discourse and visual context processing of the TALLUS model but we expect the improved learning of LEAP at these levels to make up the significant portion of the model's final contribution. The current method of generating theta roles needs to be more interdependent to be completely correct, but should again be fixed with minor improvements to the model.

The current corpus is too small to completely test LEAP. We are looking for additional corpora to test against. However, we are in the process of integrating the Sphinx speech recognition tool into the system so that it can be driven off direct human interaction. Such a live data approach is actually a better test of the IMP theory since it emulates the actual learning environment of a small child.

We have also started work on a parallel model to LEAP called REAP (Resource Extraction for Arbitrary Prose) that allows reference information from on-line resources like WordNet to be directly accessed by a LEAP-like language processor during the learning task. The cognitive justification for this model is the use of reference material (like dictionaries) by young readers during the reading task. The practical application of this model is to feed a shared database enough surface level knowledge to allow use to jump-start the deep structure learning of LEAP across more meaningful corpora.

7 Conclusion

The research being presented here is merely a stepping stone to a more complete language learning system, but it is already demonstrating that it is a pretty large stone. The data collection effort to date has focused on syntax, and clearly, the semantic aspects of LEAP are going to be its most significant final contribution. But the syn-

tactic data in itself seems be telling us something very important about the way language is processed by a human at the surface level. It could be argued that the patterns being learned in the syntax level experiment are merely shallow reflections of generative rules underlying the text, but if this is the case, why can a simple pattern do such a good job of finding a verb? Again one could argue that the verb may be closer to the center of the verb phrase, and thus, components of the verb phrase may be more likely to be contained in the pattern, but before you accept this argument you might want to carefully examine some of the grammar of the corpus we were using (red.cs.tcu.edu:14321). It is not clear to us that such a simple explanation is supported by the data. More likely, regardless of the complexity of the utterance, pattern rules dictate the range of words that can fit in the next word slot in the text. Remember that language learning in humans starts as a very slow process with the infant being immersed in a sea of language for nearly a year before they attempt to add to the flood. Could it be that they are not in this time learning a few rules, but whole lot of simple patterns? The data presented here at least make this question worth asking.

References

1. Anderson, J. R. and Lebiere, C. Atomic Components of Thought. Hillsdale, NJ: Lawrence Erlbaum Associates, Pub., 1998.
2. Anderson, J. R. Cognitive Psychology and its Implications. New York: W. H. Freeman and Company, 1995.
3. Byrne, M., "ACT-R/PM and Menu Selection: Applying a Cognitive Architecture to HCI", International Journal of Human Computer Studies, 1999.
4. Fellbaum, C (Editor), WordNet: An Electronic Lexical Database, Cambridge, MA, MIT Press, 1998.
5. Hannon, C. and D. J. Cook. "Developing a Tool for Unified Cognitive Modeling using a Model of Learning and Understanding in Young Children." The International Journal of Artificial Intelligence Tools, 10 (2001): 39-63.
6. Hannon C. and D. J. Cook. "Exploring the use of Cognitive Models in AI Applications using the Stroop Effect." In *Proceedings of the Fourteenth International Florida AI Research Society Conference*, May 2001.
7. Hannon, C., A Geographically Distributed Processing Environment for Intelligent Systems. In Proceedings of PDPS-2002. 355-360, 2002.
8. Hannon, C., Emotion-based Control Mechanisms for Agent Systems. In *Proceedings of International Conference on Information Systems and Engineering*, 2003.
9. Newell, A. Unified Theories of Cognition. London: Harvard University Press, 1990.
10. Martindale, C., Cognitive Psychology: A neural-Network Approach, Belmont, CA, Brooks/Cole, 1991.
11. Smith, G. W., Computers and Human Language, New York: Oxford Press, 1991.
12. Witbrock, Michael, D. Baxter, J. Curtis, et al. An Interactive Dialogue System for Knowledge Acquisition in Cyc. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.