A Corpus-Based Learning Technique for Building A Self-Extensible Parser

Rey-Long Liu and Von-Wun Soo
Department of Computer Science
National Tsing-Hua University
HsinChu, Taiwan, R.O.C.
e-mail: soo@cs.nthu.edu.tw

Abstract

Human intervention and/or training corpora tagged with various kinds of information were often assumed in many natural language acquisition models. This assumption is a major source of inconsistencies, errors, and inefficiency in learning. In this paper, we explore the extent to which a parser may extend itself without relying on extra input from the outside world. A learning technique called SEP is proposed and attached to the parser. The input to SEP is raw sentences, while the output is the knowledge that is missing in the parser. Since parsers and raw sentences are commonly available and no human intervention is needed in learning, SEP could make fully automatic large-scale acquisition more feasible.

Keywords: fully automatic natural language acquisition, self-extensible parser, corpus-based learning

1 Introduction

It is commonly believed in many psycholinguistics studies [Pinker, 1984; Wexler & Culicover, 1980] that extra input (in addition to raw sentences) is necessary for human language learners. Most existing computational natural language acquisition models also assumed various kinds of the extra input (e.g. semantic associations [Siskind, 1990; Webster & Marcus, 1989; Zernik, 1987] and syntactic structures [Berwick, 1985; Liu & Soo, 1992a] of input sentences) and human intervention (e.g. information interactively given by the trainer [Lang & Hirschman, 1988; Velardi et al., 1991]) during learning. The preparation of the extra input and human intervention may often cause inconsistencies, errors, and inefficiency in learning. It is often a bottle neck in scaling up natural language processing systems.

Therefore, simple syntactic heuristics had been used to collect the extra input [Brent, 1993; Sekine, 1992]. However, the information that may be collected by the simple heuristics is limited. More sophisticated processors such as taggers and parsers had also been tested in collecting the extra input [Webster & Marcus, 1989; Zernik, 1989]. However, since learning was based on the input that may be successfully and unambiguously analyzed by the processors, upgrading the performance of the processors became a new bottle neck of upgrading the performance of learning. Furthermore, since the heuristics and processors are separated from the learning component, the learning component cannot know what knowledge is actually missing in the heuristics and processors. The learning component might learn the knowledge that the processors already have.

In this paper, we investigate the robust ways of acquiring parsing knowledge *without* requiring extra input and human intervention. The input to the system is raw sentences, while the output is the knowledge that is missing before learning. The parser takes an active role in collecting extra information for learning. Thus a parser could be self-extensible in the sense that it could automatically acquire what it actually lacks without relying on extra information from the outside world. The parsing capability is improved through learning, and in turn, the learning capability is improved due to the improved parsing capability.

To achieve that, learning should be triggered when the parser fails to analyze input sentences. In that case, however, there might be a large number of hypotheses to fix the failures. For example, suppose a parser finds an unknown word when parsing a sentence. In that sentence, the unknown word may have many possible syntactic and semantic behaviors (e.g. parts-of-speech and argument structures) which may lead to many different sentence structures. Therefore, the major challenge of the study is the generation and verification of the hypotheses for missing knowledge.

We thus propose a learning technique called SEP that is attached to the parser. SEP is triggered when the parser fails in parsing raw sentences. In the next two sections we describe SEP's two modules: the hypothesis generation module and the hypothesis verification module. For each training sentence, the two modules are triggered sequentially. Experimental results are then presented in section 4.

2 Generation of hypotheses

SEP generates hypotheses based on the partial results of failed parsing, universal linguistic constraints, and the parser's existing knowledge.

2.1 Collecting partial results of parsing

As the parser's knowledge is incomplete for parsing an input sentence, the bottom-up chart parsing strategy is suitable in collecting partial results of parsing. The failures of parsing are due to the lack of some knowledge pieces for grouping the partial results into the target constituent (e.g. a major sentence Smaj).

As an example, consider the sentence "taking exercises is good for your health". A parser with a complete knowledge base for parsing the sentence may construct a constituent Smaj after deriving all relevant constituents such as noun phrases (NPs) and verb phrases (VPs). The grouping of constituents should be based on both syntactic and semantic constraints. As the parser does not have adequate knowledge for parsing the sentence, failures will occur. For example, parsing will fail if "take" is an unknown word for the parser. There might be a large number of hypotheses to fix the failure. For example, "take" may be a noun or a verb. The learner may even hypothesize that Smaj may be constructed by matching the sequence "taking NP VP". That is, without the help of additional information, a huge number of ridiculous hypotheses might be generated.

2.2 Generating hypotheses based on universal constraints and the parser's knowledge

There are universal linguistic constraints that may restrict the forms of missing knowledge. The X-Bar theory, for example, postulates that any maximal projection (constituent) should be composed of at most three components: a specifier, an argument structure of the lexical head (i.e. X-bar), and a modifier [Chomsky, 1981]. The set of possible subcategorization frames of lexical heads had also been set-up in many studies [Gazdar, 1985]. Based on these studies, SEP allows a constituent to have at most three components. This constraint (called *Three-Components Constraint*) is incorporated into SEP's hypothesis generation process which is composed of two phases: the top-down phase and the bottom-up phase. The bottom-up phase is triggered after the top-down phase is completed.

2.2.1 The top-down phase

In the top-down phase, SEP uses the parser's existing knowledge to perform top-down prediction of missing knowledge. This phase is for the case in which the parser has knowledge for constructing a constituent (e.g. Smaj may be constructed by an NP followed by a VP) excepts the knowledge for constructing all the individual components (e.g. the NP and the VP) of the constituent.

The input sentence:

  Taking exercises is good for your health.

Constituents already constructed:

  NP(2-2), VP(3-4), NP(6-7), PP(5-7), VP(3-7)

The top-down phase:

  Pass 1: **Smaj(1-7)** :- **NP(1-2)**, VP(3-7).

  Pass 2: **NP(1-2)** :- **VP(1-2)**.

  Pass 3: **VP(1-2)** :- ??

The bottom-up phase:

  Step 1: Completed constituents: NP(2-2)

  Step 2: Hypothesis: **VP(1-2)** :- verb(1-1), NP(2-2)

  Step 3: Checking the Three-Components Constraint

  Step 4: If valid, return the hypothesis

Fig.1. An example trace of SEP's hypothesis generation

As an example, consider "taking exercises is good for your health". Suppose "take" is an unknown word. Thus parsing fails, and SEP is triggered. The reasoning process is illustrated in Fig.1. In Fig.1, the numbers denote the starting positions and ending positions of constituents, and the constituents that cannot be constructed in parsing are marked in the boldface form. In the top-down phase, SEP first searches for the parser's knowledge pieces for constructing the top level goal constituent Smaj. Suppose one of the knowledge pieces says that Smaj may be constructed by an NP followed by a VP (Pass 1 in Fig.1). Since the VP may be instantiated by "is good for your health", SEP expects there should be an NP from position 1 to position 2. Thus SEP retrieves all knowledge pieces for constructing NPs. Suppose that one of them says that a predicate NP may be constructed from a VP (Pass 2 in Fig.1). Thus, SEP attempts to retrieve VP rules. However, since "take" is unknown, no knowledge may be retrieved for constructing the VP (Pass 3 in Fig.1). The top-down phase thus stops and the bottom-up phase is triggered.

2.2.2 The bottom-up phase

In the bottom-up phase, SEP uses the partial results of parsing to perform bottom-up prediction of missing knowledge. This phase is for the case in which the parser has knowledge for constructing all the individual components (e.g. an NP and a VP) of a constituent (e.g. Smaj) excepts the knowledge for grouping these components (e.g. Smaj may be constructed by the NP followed by the VP).

For the above example, SEP has hypothesized that there is a VP from position 1 to position 2. In the bottom-up phase, SEP first observes the partial results of parsing from position 1 to position 2. Since only the NP "exercises" is constructed in this range (Step 1 in Fig.1), the NP is the only possible argument in the VP. Thus the hypothesis "VP(1-2) :- verb(1-1), NP(2-2)" is generated (Step 2 in Fig.1). Then the Three-Components Constraint is checked (Step 3 in Fig.1). Since the hypothesis satisfies the constraint, it may be returned as a hypothetic knowledge piece (Step 4 in Fig.1). If the hypothesis is confirmed (see section 3), SEP acquires both a category and an argument structure of "take".

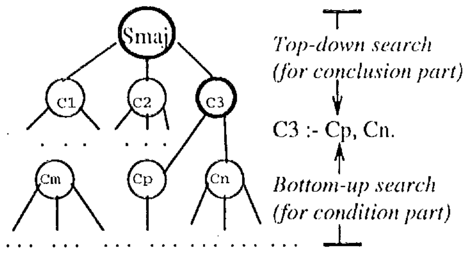

Fig.2. The top-down phase and the bottom-up phase

In summary, the top-down phase and the bottom-up phase of SEP are complementary to each other. In the top-down phase, SEP hypothesizes the conclusion parts of the missing rules, while in the bottom-up phase, SEP hypothesizes the condition parts of the missing rules. A schematic view of the two phases is illustrated in Fig.2. The top-down phase starts from the constituents not constructed (marked with boldface circles) to the constituents already constructed. The bottom-up phase starts from the constituents already constructed to the constituents not constructed. The two phases meet at the possible failure points of parsing. A failure point indicates that there is a missing rule in the parser.

It should be noted that, in generating hypotheses, SEP might need to consider several reasoning trees such as the one in Fig.2. This is because there might be several rules for constructing a constituent (e.g. Smaj). Each rule indicates a path of reasoning, and thus leads to a new reasoning tree.

If the parser has only one missing rule for parsing a sentence, the top-down phase and the bottom-up phase will be able to meet at the corresponding failure point. This is because in that case SEP will have enough rules (including grammar or lexical rules) to perform top-down traversal and enough constituents (already constructed in parsing) to perform bottom-up traversal. On the other hand, if the parser has more than one missing rules for parsing a sentence, the top-down phase will stops at the points which the bottom-up phase cannot reach. For example, if both Cp and Cn in Fig.2 cannot be constructed in parsing, the two phrases cannot meet. In that case, the ambiguity space may be too large to resolve. That is, we may have:

The Complete-When-One-Missing Theorem: Supoose the parser lacks only one rule R (either a lexical rule or a grammar rule) to completely parse a sentence. Then R will be included in the set of hypotheses generated by SEP. That is, if the input sentence is "not difficult" for the parser to learn knowledge from it, the missing rule will be generated by SEP.

Therefore, in each step of finding an existing rule to perform top-down traversal, SEP selects a rule only when all but one of the components in the condition part of the rule are constructed in parsing. If no such rules may be found, the top-down phase stops and the bottom-up phase is triggered. If the input sentence is not too difficult for the learner, the bottom-up phase and the top-down phase can meet at a failure point, and thus hypotheses of missing knowledge may be generated. If the parser has only one missing rule for parsing the input sentence, the missing rule will be in the hypothesis set generated. This from-simple-to-difficult learning sequence is assumed in many learning models (and human learners as well). Since raw sentences are commonly available, SEP may easily get suitable training sentences.

3 Verification of hypotheses

SEP's hypothesis generation module might generate several hypotheses whose validities should be verified before assimilated into the parser's knowledge base. The algorithm of the hypothesis verification module is outlined in Fig.3.

```
Algorithm: SEP's hypothesis verification
Input: Sets of hypotheses generated
Output: A hypothesis of the target missing rule
Begin:
  For each hypothesis set H generated for a sentence
    If there is only one hypothesis in H, return the hypothesis; (Step 1)
    Otherwise, increment the frequency of each hypothesis h in H by one; (Step 2)
  Return the hypothesis with the highest frequency; (Step 3)
End.
```

Fig.3. The algorithm of SEP's hypothesis verification

SEP's hypothesis verification module makes decision based on the hypothesis sets generated for training sentences (one hypothesis set per training sentence). If there is only one hypothesis in any one of the hypothesis sets, SEP returns the hypothesis as the target missing knowledge (Step 1). Since only one hypothesis in each hypothesis set may be confirmed (i.e. only one rule is missing), other hypotheses in the hypothesis sets may be excluded.

If more than one hypotheses are generated for a training sentence, the frequency of each of the hypotheses is updated (Step 2). After considering all the hypothesis sets, SEP returns the hypothesis with the highest frequency of occurrence (Step 3). It is obvious that, a hypothesis with a higher frequency of being generated is more likely to be the target missing knowledge.

As a hypothetic knowledge piece is confirmed, it should be annotated with critical syntactic and semantic features [Liu & Soo, 1992a, 1992b]. A knowledge piece without suitable feature annotation will be too general and thus useless. For example, suppose the learner acquires a knowledge piece for constructing a predicate NP from a VP (e.g. "taking exercises"). It must annotate the NP with the feature "NUM=singular"; otherwise the ungrammatical sentence "taking exercises *are* good for your health" will be accepted as well. The annotation is based on universal linguistic principles such as the universal feature instantiation principles in generalized phrase structure grammar (GPSG [Gazdar, 1985]). For example, the feature "NUM=singular" is annotated by observing the fact that the verb "is" needs a singular external argument.

4 Experiment

In the experiment, SEP's hypothesis generation and verification modules were evaluated. We used a parser whose knowledge base included 2513 lexicon entries, 22 grammar rules, and 20 morphological rules.

4.1 Evaluation of the hypothesis generation module

To compare the performance of SEP with a learner that is provided with extra input, a set of sentences that had been tested in previous experiments [Liu & Soo, 1993a, 1992a, 1992b] was entered to SEP. The difference was that, only raw sentences were entered to SEP. There were 165 sentences (about 1700 words) in the corpus. These sentences were majorly extracted from two articles of an English textbook.

Among the 165 sentences, 80 sentences were successfully parsed by the parser, and hence SEP was not triggered for them. SEP was triggered for acquiring the missing knowledge for parsing the other 85 sentences. There were totally 202 hypotheses generated. Thus, on average SEP produced 2.38 (202/85) hypotheses per input sentence that cannot be parsed by the parser. Furthermore, among the 85 sentences that triggered learning, SEP successfully generated hypotheses from 55 sentences. That is, the 55 sentences were not too difficult for the parser. From this point of view, SEP produced 3.67 (202/55) hypotheses per missing rule. Therefore, SEP needs to collect more evidences in order to determine a target missing rule among 3.67 hypothetic knowledge pieces.

(1) AP(3-11) :- NP(3-5), S(6-11).
(2) NP(3-11) :- NP(3-5), S(6-11).
(3) VP(2-11) :- is(2-2), NP(3-5), S(6-11).
(4) NP(1-6) :- S(1-5), NP(6-6).
(5) S(1-11) :- S(1-5), S(6-11).
(6) Smaj(1-11) :- S(1-5), S(6-11).

Fig.4. An example of the hypotheses generated by SEP

As an example, consider the sentence "Lead is a soft metal that serves many purposes in home" in the corpus. The parser had a missing rule for constructing NPs with relative clauses. Six hypotheses were generated by SEP They are illustrated in Fig.4 (for the illustrative purpose, syntactic and semantic features are omitted). Hypothesis (1) and (2) were generated based on the existing argument structures of "is"; SEP thought that if an AP (Adjective Phrase) or an NP may be constructed from position 3 to position 11, parsing may become successful. Hypothesis (3) was generated for learning a new argument structure of "is". Hypothesis (4) was generated when SEP hypothesized "serve" as the main verb of the sentence. Hypothesis (5) and (6) were generated since SEP thought the parser might need to know a new sentence structure. Among the six hypotheses, the target missing rule is hypothesis (2), which is quite likely to have the highest frequency of being generated in learning on a large corpus of sentences.

4.2 Evaluation of the hypothesis verification module

In the experiment, we evaluated SEP's performance in hypothesis verification. Training sentences were extracted from the DJ corpus (Wall Street Journal articles). The size of the corpus was about 32 mega bytes. Since SEP only assumed raw sentences as input, other kinds of information (e.g. the part-of-speech information) in the DJ corpus were not considered in the experiment.

For verifying the hypotheses generated for parsing the above sentence "Lead is a soft metal that serves many purposes in home" (Fig.4), 1000 sentences containing "that" were extracted from 19200 sentences (71294 words) of the DJ corpus. The 1000 sentences were fed into the parser. As described above, since the knowledge for the NPs with relative clauses was missing, parsing failed and SEP was triggered for each sentence. In many cases, SEP could not generate any hypotheses, since there were many unknown words for the parser. As hypotheses could be generated, SEP updated the frequencies of the hypotheses (Step 2 in Fig.3). As learning proceeded to the sentence:

... SEC, *an agency that covets its independence,* ...

SEP generated only one hypothesis "NP :- NP, S". This was because, "an agency ..." was unambiguously segmented by commas and expected to be a noun phrase. Since only one hypothesis was generated, the hypothesis was returned as the target missing knowledge (Step 1 in Fig.3). That is, SEP concluded that hypothesis (2) in Fig.4 was needed (and missing in the parser) for parsing both sentences. Therefore, although the sentences that are too difficult for the parser are skipped, SEP may still find suitable sentences to learn since raw sentences are commonly available.

The current version of SEP may be extended in the following two ways:

- Acquisition of movement constructions: Movement constructions cannot be learned using SEP. For example, the movement construction in the sentence "Here comes the dog" cannot be learned, since it cannot be detected if only raw sentences are entered to the learner [Liu & Soo, 1992b].
- The use of more universal linguistic constraints: More suitable universal linguistic constraints may be used to both reduce the number and promote the quality of the hypotheses generated. For example, hypothesis (1) in Fig.4 may be filtered out by consulting the fact that the head component of a constituent should be included in the condition parts of the rules for constructing the constituent. Since neither NP nor S may be the head of AP, the hypothesis may be discarded. As another example, according to X-Bar theory, each constituent is composed of a specifier, an argument structure, and a modifier. The possible syntactic categories (e.g. NP, PP, VP, and AP) of the components of each kind of constituents have been identified in previous linguistics studies (e.g. a determiner may be the specifier of an NP). It is obvious that, if SEP generates hypotheses by considering the universal constraints, the quality of the generated hypotheses may be promoted.

5 Related work

Previous natural language acquisition models could be characterized as interactive acquisition [Lang & Hirschman, 1988; Liu & Soo, 1993a, 1993b; Velardi et al., 1991], corpus-based acquisition [Brent, 1993; Jacobs & Zernik, 1988; Zernik, 1989], dictionary-based acquisition [Montemagni & Vanderwende, 1992; Sanfilippo & Pozanski, 1992], statistics-based acquisition [Smadja, 1991; Sekine et al, 1992], and connectionist-based acquisition [Faisal & Kwasny, 1990; McClelland & Kawamoto, 1986].

Our motivation in the study is to provide the parser with the capability of extending itself without using extra input and intervention from the outside world. From this point of view, interactive acquisition and connectionist-based acquisition will have difficulties in resolving the problems of inconsistencies, errors, and inefficiency of learning, since they required the information encoded by the trainer.

SEP could be characterized as corpus-based acquisition. The point here is that SEP only assumes raw sentences as the input. From the point of view, corpus-based acquisition and statistics-based acquisition that require pre-processed data will have difficulties in getting adequate extra input, since raw sentences are much more commonly available than pre-processed data.

SEP collects observations from the parser which may grow through learning. From this point of view, collecting information using simple but non-extensible heuristics [Brent, 1993] might miss many opportunities of learning (due to the inadequacy of the collected information), although raw sentences are assumed as the major input in that study as well.

In addition to grammar and lexical information from the parser, other types of useful information may include contextual, conceptual, and associational semantic information [Siskind, 1990; Jacons & Zernik, 1988; Webster & Marcus, 1989; Zernik, 1987], although they are much more difficult to collect in practice (especially when the parser is incomplete and the input sentences are noisy in many real-world applications). This paper explores the feasibility of allowing a parser (either preliminary or sophisticated) to extend itself with available raw sentences and information from itself. Noise-tolerant learning is implemented by allowing the learner to acquire knowledge based on a large number (rather than one) of observations. The practical learning method may make efficient, fully-automatic, and large-scale acquisition more approachable.

Parsers (or taggers) had been used in many previous models as well. They were pre-processors of learning [Zernik & Jacobs, 1990; Pustejovsky et al., 1993; Montemagni & Vanderwende, 1992; Sanfilippo & Pozanski, 1992; Zernik, 1989] or post-processors of learning [Smadja, 1991]. In those models, the processors were assumed to be complete and thus separated from the learning components. Learning was based on the "success" of syntactic processing. To extend the capability of the processors, learning should be triggered when the processors fail to analyze input sentences. This is the reason why SEP attempts to learn knowledge when the parser fails in parsing raw sentences. For those models assuming syntactic information as their necessary input, SEP may be attached to them to make their processors more self-extensible.

We are currently integrating SEP with a framework of syntactic and semantic knowledge acquisition [Liu & Soo, 1992a; Liu & Soo, 1993b].

6 Conclusion

Building necessary knowledge bases is a major bottleneck of designing a practical parser. Previous studies on the problem had proposed many learning methods to reduce the difficulty. However, during learning most of them still need extra information or human intervention, which are the major sources of inconsistencies, errors, and inefficiency in learning. This paper is thus dedicated to the *fully* automatic natural language learning in which only raw sentences are entered to the system. We study how a parser may extend itself by observing its own experiences of parsing. The proposed learning technique SEP is triggered when the parser fails in parsing raw sentences. It is shown that hypotheses for missing knowledge may be generated based on the parser's existing knowledge, universal linguistic constraints, and partial results of the failed parsing. Those hypothetic knowledge pieces that are likely to facilitate successful parsing may then be extracted as the new parsing knowledge. Thus the parser may acquire the knowledge that it actually lacks in parsing. As more parsers become available, SEP may be attached to them in order to enhance their knowledge fully automatically.

Acknowledgement This research is supported in part by NSC (National Science Council of R.O.C.) under the grant NSC83-0408-E-007-008.

Reference

Berwick R. C. (1985). *The Acquisition of Syntactic Knowledge,* The MIT Press, Cambridge, Massachusetts, London, England.

Brent M. R. (1993). *From Grammar to Lexicon: Unsupervised Learning of Lexical Syntax,* Computational Linguistics, Vol. 9, No. 2, pp. 243-262.

Chomsky N. (1981). *Lectures on Government and Binding,* Foris Publications - Dordrecht.

Faisal K. A. and Kwasny S. C. (1990). *Design of a Hybrid Deterministic Parser,* Proc. of COLING.

Gazdar G., Klein E., Pullum G. K., and Sag I. A. (1985). *Generalized Phrase Structure Grammar,* Harvard University Press, Cambridge, MA.

Jacobs P. and Zernik U. (1988). *Acquiring Lexical Knowledge from Text: A Case Study,* Proc. of AAAI.

Lang F.-M. and Hirschman L. (1988). *Improved Portability and Parsing through Interactive Acquisition of Semantic Information,* Proc. of the second conference on Applied NLP, pp. 49-57.

Liu R.-L. and Soo V.-W. (1993a). *Parsing-Driven Generalization for Natural Language Acquisition,* International Journal of Pattern Recognition and Artificial Intelligence, Vol. 7, No. 3.

Liu R.-L and Soo V.-W (1993b). *An Empirical Study on Thematic Knowledge Acquisition Based on Syntactic Clues and Heuristics,* Proc. of the ACL-93.

Liu R.-L. and Soo V.-W. (1992a). *Augmenting and Efficiently Utilizing Domain Theory in Explanation-Based Natural Language Acquisition,* Proc. of the 9th International Machine Learning Conference.

Liu R.-L and Soo V.-W. (1992b). *Acquisition of Unbounded Dependency Using Explanation-Based Learning,* Proc. of ROCLING V.

McClelland J. L. and Kawamoto A. H. (1986). *Mechanisms of Sentence Processing: Assigning Roles to Constituents of Sentences,* in Parallel Distributed Processing, Vol. 2, pp. 272-325.

Montemagni S. and Vanderwende L. (1992). *Structural Patterns vs. String Patterns for Extracting Semantic Information from Dictionary,* Proc. of COLING-92, pp. 546-552.

Pinker S. (1984). *Language Learnability and Language Development,* The Harvard University Press, Cambridge, Massachusetts, London, England.

Pustejovsky J., Berger S, and Anick P. (1993). *Lexical Semantic Techniques for Corpus Analysis,* Computational Linguistics, Vol. 9, No. 2, pp. 331-358.

Sanfilippo A. and Pozanski V. (1992). *The Acquisition of Lexical Knowledge from Combined Machine-Readable Dictionary Sources,* Proc. of the Third Conference on Applied NLP, pp. 80-87.

Sekine S., Carroll J. J., Ananiadou S., and Tsujii J. (1992). *Automatic Learning for Semantic Collocation,* Proc. of the Third Conference on Applied NLP.

Siskind J. M. (1990). *Acquiring Core Meanings of Words, Represented as Jackendoff-style Conceptual Structures, from Correlated Streams of Linguistic and Non-linguistic Input,* Proc. of the ACL-90.

Smadja F. A. (1991). *From N-Grams to Collocations: An Evaluation of EXTRACT,* Proc. of the 29th annual meeting of the ACL, pp. 279-284.

Velardi P., Pazienza M. T., and Fasolo M. (1991). *How to Encode Semantic Knowledge: A Method for Meaning Representation and Computer-Aided Acquisition,* Computational Linguistics, Vol. 17, No. 2.

Webster M. and Marcus M. (1989). *Automatic Acquisition of the Lexical Semantics of Verbs from Sentence Frames,* Proc. of the ACL-89, pp. 177-184.

Wexler K. and Culicover P. W. (1980). *Formal Principles of Language Acquisition,* The MIT Press, Cambridge, Massachusetts, London, England.

Zernik U. (1987). *Learning Idioms – With and Without Explanation,* Proc. of IJCAI, pp. 133-136.

Zernik U. (1989). *Lexicon Acquisition: Learning from Corpus by Capitalizing on Lexical Categories,* Proc. of IJCAI, pp. 1556-1562.

Zernik U. and Jacobs P. (1990). *Tagging for Learning: Collecting Thematic Relation from Corpus,* Proc. of COLING, pp. 34-39.